NAIST-IS-DD0561040

Doctoral Dissertation

Constructing a Temporal Relation Identification System of Chinese based on Dependency Structure Analysis

Yucheng CHENG

March 17, 2008

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to the Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Engineering.

Thesis Committee:
Professor Yuji Matsumoto (Supervisor)
Professor Shin Ishii (Co-supervisor)
Associate Professor Kentaro Inui (Co-supervisor)
Assistant Professor Masayuki Asahara (Co-supervisor)

依存構造に基づく中国語事象表現の時間関係同定システムの構築に関する研究[1]

鄭育昌

内容梗概

時間情報処理は時間表現や事象表現を同定し、表現間の時間関係を解明する研究である。言語学、心理学においては言語の時間表現の論理モデルに関する研究は数多く提案されている。しかし、時間情報の自動処理を行うため必要な時間情報タグ付きコーパスの構築に関する研究が少ない。特に、英語ではタグ付き基準 TimeML (Pustejovsky, 2006 [59]) が提案されているのに対して、中国語の時間情報タグ付きコーパスに関する研究は殆どない。本研究では、中国語の事象間の時間関係をタグづけする基準を策定し、その関係を同定する手法を提案する。

本研究の調査により、中国語と英語の新聞記事中に出現する事象表現と時間表現の数は不均衡であり、時間情報の理解には事象表現同士の関係推定が重要なタスクであることが判明した。それればかりではなく、中国語においては、テンスとアスペクトを陽に示す表現が欠如し、時間情報は語彙と文脈に依存するという特徴がある。従って、中国語の時間関係処理に対して時間表現を中心とする手法は不十分であり、事象間の時間関係同定が重要なタスクである。従来の時間表現を中心とする手法と異なって、本研究は事象間の時間関係同定を時間情報処理の独立したタスクとして扱い、中国語の事象間の時間関係を同定する手法を提案し、タグ付きコーパスを構築し機械学習器で事象間の時間関係の自動同定システムを提案する。

全種類の事象を同定することが困難であるため、本研究は事象を動詞に限定する。本研究で提案する中国語の時間関係コーパスのタグ付け基準は TimeML の基準に基づいて

---

[1]奈良先端科学技術大学院大学情報科学研究科情報処理学専攻学位論文, NAIST-IS-DD0561040, 2008 年 3 月 17 日.

いる。提案するタグ付け基準は依存構造解析結果を用いて、タグ付け作業量を減らすことを目標にしている。依存構造を用いて三つの時間関係（隣接関係、依存構造上の親子関係及び依存構造上の兄弟関係）を定義する。さらに長距離の関係を得るために推論規則を使用して時間関係を拡張する。同一記事中ある n 個の事象に対して、すべての組合せの時間関係を考慮するため n(n-1)/2 回 の判定が必要であるが、提案手法による作業量は 3n 回の判定でよく、これにより人手による動詞事象対の時間関係のタグ付け作業量を軽減できる。

本研究では構文解析済みのコーパスーPenn Chinese Treebank を基本データとして、動詞間時間関係のタグ付きコーパスを構築し、本タグ付きコーパスを用いて時間関係の自動同定システムを実装する。依存構造の情報を獲得するため、本研究では機械学習に基づく中国語の依存構造解析器を開発し、既存の中国語依存構造解析器を上回った単語の依存関係の正解率（88 %）を達した。

時間関係の同定には次の手順による: 依存構造を解析し、事象の時間関係を解析し、最後は推論規則を使用して長距離関係を拡張する。提案手法で構築したタグ付きコーパスを用いて、機械学習器として Support Vector Machine を用いることにより、時間関係の自動同定システムを作成した。システムの評価を測るため、本研究では小規模コーパスを用いて被覆率を検証する。評価実験において提案した各種類の事象の時間関係の解析正解率は 68%~71% に達し、記事中のすべての可能な事象間の時間関係の 53% が提案手法により実装したシステムで同定できることがわかった。

キーワード：時間関係同定、事象関係、自然言語処理、依存構造、構文解析アルゴリズム、中国語構文解析

Constructing a Temporal Relation Identification System of Chinese based on Dependency Structure Analysis[2]

Yucheng CHENG

Abstract

"Temporal information (Time)" has been a subject of study in many disciplines particularly in philosophy, physics, and is an important dimension of natural language processing. The temporal information includes temporal expressions, event expressions and temporal relations. There are many researches dealing with the temporal expressions and event expressions. However, researches on temporal relation identification and the construction of temporal relation annotated corpus are still limited. There is a well-known temporal information annotated guideline for English, TimeML (Pustejovsky, 2006 [59]). However, there is no such a research that focuses on this in Chinese. Our research is the first work of the temporal relation identification between verbs in Chinese texts. In this research, we propose a temporal information annotation guideline for Chinese and a machine learning-based temporal relation identification method.

Following the observation of our investigation, the distribution of events and temporal expressions is un-balance. The temporal information processing includes two independent tasks: anchoring the temporal expressions on a timeline and ordering the events to temporal order. Our research focuses on ordering the events, which is to identify the temporal relations between events. Because identifying the nominal event is difficult, we limit the events to the verbs in articles. The proposed annotation guideline is based on the TimeML language. We newly introduce dependency structure information to limit target temporal relations. The proposed method reduces the manual

---

[2] Doctoral Dissertation, Department of Information Process, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561040, March 17, 2008.

efforts in constructing the annotated corpus. To annotate temporal relations of all combinations of events requires $n(n-1)/2$ manual judges. Our proposed method requires at most $3n$ manual judges. While the dependency structure based attributes reduce manual annotation costs, the limited relations preserve the majority of the temporal relations.

We use a syntactic parsed corpus - Penn Chinese treebank as the original data for annotating a basic annotated corpus. For using the dependency structure in temporal relation identification, we first construct a dependency analyzer for Chinese and combine it into the temporal relation annotating system. The process of temporal relation identification includes following steps: to analyze the dependency structure, to analyze the temporal relation attributes of events and to extend the relation using the inference rule. We define events as those expressed by verbs and define the temporal relation types of event pairs which include the adjacent event pairs, the head- modifier event pairs and the sibling event pairs. These relations include most meaningful information, and we extend these relations using the inference rules to acquire long distance relations.

We train a machine learner with our temporal relation annotated corpus to construct the temporal relation identifying system. Support Vector Machine is used as the machine learner in this system. We survey the coverage of our system with a small corpus. The accuracy of the dependency analyzer is 88% for word dependency analysis and this is better than existed Chinese dependency analyzer. In our experiments, the accuracies of the automatic annotating the temporal relation attributes are 68%~71%. The result shows that our proposed system covers about 53% of temporal relations of all possible event pairs.

Keywords: Temporal Entities, Event Entities, Temporal Reasoning, Event Semantics, Dependency Structure

Contents:

LIST OF FIGURES:

List of Tables:

Chapter 1 Introduction

1.1 Motivations

1.1.1 Temporal information

"Temporal information (Time)" has been a subject of study in many disciplines particularly in philosophy, physics, and art. Temporal information is an important dimension of any information space. The descriptions at an English dictionary (Alonso et al., 2007 [94]) shows the following definition for "time": a) a non-spatial continuum in which events occur in apparently irreversible succession from the past through the present to the future; b) An interval separating two points on this continuum; a duration. In natural language processing, newspaper texts, narratives and other such texts describe events which occur in time and specify the temporal location and order of these events. Text comprehension, even at the most general level, involves the capability to identify the events described in a text and locate these in time. This capability is crucial to a wide range of NLP applications, from document summarization and question answering to machine translation.

Temporal information is explicit or implicit in documents. The temporal information includes three elements:

- Temporal expressions, which describe time points, periods or sets of time points / periods on a timeline. Temporal expressions can be fully specified temporal expressions, underspecified and relatives (Schilder and Habel, 2001 [93]). The fully specified temporal expressions directly describe entries in some timeline, such as an exact date or year (ex. 3/17/2008, 2008…) and they can be mapped directly to chronons in a timeline. The underspecified Temporal Expressions depend on the underlying time ontology and capabilities of the named entity extraction approach, even apparently imprecise temporal information, such as names of holidays or events can be anchored in a timeline (ex. Monday, Christmas…). The relative temporal expressions represent temporal entities that can only be anchored in a timeline in reference to another fully specified or underspecified temporal expression, already anchored temporal expression. For example, the expression "today" alone cannot be anchored in any timeline. However, it can be anchored if the document is known to have a creation date. This date then can be used as a reference for that expression, which then can be mapped to a chronon. There are many instances of relative temporal expressions, such as the names of weekdays (e.g., "on Thursday") or months (e.g., "in July"). A well-known format–ISO8601 standard is usually used to represent temporal expressions. This format represents a time point as following notification- "YYYY-MM-DDThh:mm:ss (years–month-dateThour:minute:second)". For example, a temporal expression "November 5, 2007, 8:15:30 am" can be rewritten to "2007-11-05T08:15:30". An ideal processing of the temporal expression reorganization is to translate all kinds of temporal expressions to ISO-8601 standards. However, many temporal expressions include non-numerical part and have ambiguity in its interpretation. To translate all temporal expressions to ISO standards is still a studying task.

- Event or situation expressions that occur instantaneously or that last for a period of time in the actual or hypothetical world. The events are tensed / untensed verbs, stative adjectives and event nominals. Events are a focus entity in our research. The representation of events depends on languages. More detail and definition of events in Chinese articles are described in section 1.2 and 5.3.
- Temporal relations, which describe the ordering relation between an event expression and a temporal expression, or between two event expressions. Temporal relations include three kinds of actions- anchoring, ordering and embedding. Anchoring means to anchor a temporal expression or an event on a timeline with a correct, clear position. Ordering means to sort temporal expressions and events on a timeline with a temporal order. Embedding means an event or a temporal expression

that is subordinated into another event or temporal expression. The orderings and embeddings are regarded as "related relations" and the anchoring is "absolute relation". In our research, dealing with the related relations between events is the main task. The proposed method in Chapter 3 describes more details.

The automatic identification of all temporal referring expressions, events and temporal relations within a text is the ultimate aim of research in this area. As an alternative to document ranking techniques like those based on popularity, time can be valuable for placing search results in a timeline for document exploration purposes (Alonso et al., 2007 [94]). Current information retrieval systems and applications, however, do not take advantage of all the time information available within documents to provide better search results and thus to improve the user experience. In next section, we illustrate an example for describing the usefulness of the temporal information processing.

1.1.2 The applications of temporal information

Temporal information plays an essential role in a variety of application areas in many NLP applications such as question answering, text summarization, machine translation, information extraction and discourse analysis. Question-answering systems need to provide an answer to a when-question, whereas information-extraction systems are often required to fill template slots for information regarding time (e.g., when the event took place). For these systems, temporal information is one of the final targets to be extracted. More recently, time-based automatic summarization has been studied as a relatively unexplored area (Allan et al. 2001 [2]), where temporal information is used to select key sentences from multiple news stories covering the same topic. Here, temporal information is a useful vehicle toward improved performance, rather than an ultimate target. Topic detection and tracking (TDT) is another area where a processing of temporal information is in high demand (Kim and Myaeong, 2004 [41]). The task is to track a particular topic/event or to detect an occurrence of a new topic/event from a series of news stories or broadcasted messages.

A concept of the aim of temporal information analysis is illustrated in Figure 1-2. Assuming that we have news articles with the same topic "the breakout of the golden price", the most important information that a reader want to know is "what's happen?" and “when?”. “What’s happen?” can be regarded as “events” and “when?” can be regarded as “the occurrence time of an event”. If we focus on the specific topic, we want to know the temporal order of these events to acquire more useful information of the golden price. The main events in the topics of articles and the related temporal expression (the publication time, the reference temporal expression and occurrence time) are shown in the table of Figure 1-2. “Publication time” is the time that the article is published. “Reference time” is temporal expressions that are the clues for anchoring the events on a timeline. Generally, the position in a timeline of the publication time is usually after the position of the reference time and the occurrence time. The temporal order of publication times is different from the order of the reference time and the occurrence time. The reference time is a powerful clue for recognizing the order of events. However, the meaning of the reference time is usually ambiguous and it lacks particular information. In many situations, the event does not have a reference time that can anchor the event on a timeline (see section 1.2 and 2.2.2). Therefore, we want to acquire a temporal ordered event sequence but we usually need to order them without the particular reference time. This concept is the motivation of our research.

In this research, we propose a temporal information annotation guideline for Chinese and a machine learning-based temporal relation identification method. Figure 1-1 illustractes the construction of our temporal relation analysis system and Figure 1-5 illustrates an overview of this dissertation. We focus on how to order events without temporal expression and we propose an automatic identification system for dealing with the temporal relation between events in Chinese news articles. In next section, we introduce more observations of the temporal information in Chinese and explain our novel ideal for temporal information processing.

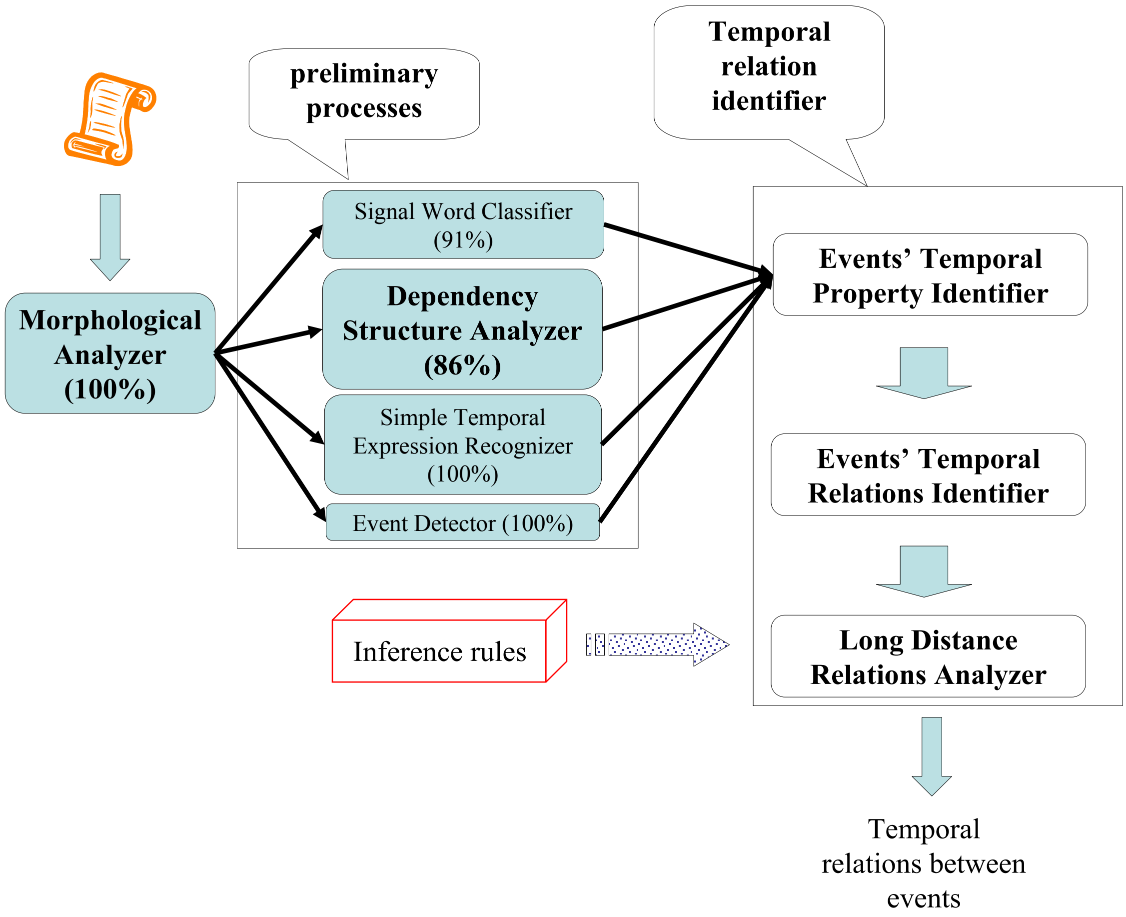


Figure 1-1: The processing flow of our temporal information analyzer

Example: News articles of the breakout of golden price

Publication time	Reference time	Occurrence time	Topic Events
2007/11/23	今天(today)	2007/11/22	金價創27年新高 (the golden price achieved new high price in recent 27 years)
2007/11/23	XX	XX	央行釋出儲備黃金(The central bank release the stockpiles gold)
2007/11/23	XX	XX	匯率大跌(The exchange rate greatly falls)

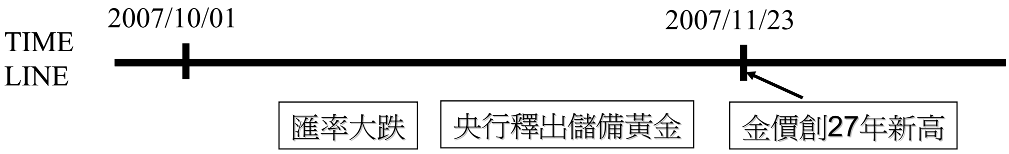


Figure 1-2: The news articles of the breakout of the golden price

1.2 Temporal Information processing for Chinese

1.2.1 The characteristic of the temporal information in Chinese

The target language of our research is Chinese. Most important characteristic of Chinese that needs to be noticed is that verbs in Chinese don't change the morphemes according to tense. For example:

- 我*去*東京 (I (will) ***go*** to Tokyo).
- 我昨天*去*東京(I ***went*** to Tokyo yesterday).

Even though the verb event “去 (go)” has different tense in these examples, the morphemes in them are the same. Therefore, to order events on a timeline, we need to consider the context and other useful clues of temporal relation in Chinese instead of the tense of verbs. In addition to the three kinds of the temporal information, we conclude several clues for the temporal information processing in Chinese. That is, verb classes with temporal features, temporal adverbs and aspect auxiliary words. The verb classes with temporal features (陳平, 1987 [79]; 鄧守信, 1986 [80]) are the semantic information that can be used to describe the temporal properties of a verb. We apply this concept into the event features in our annotating criteria (see the section 5.2.4).

Extremely, Chinese is a language without tense (Lin, 2005 [53]). But we have some clues for recognizing the meaning of tense and aspect in Chinese (張秀, 1957 [81]; 馬慶株, 2000 [82]). They are the temporal adverbs (ex. 在 (being), 已經 (have), 將要 (will)) and aspect auxiliary words (了 (been), 過 (was)). These adverbs and auxiliary words can combine with verbs to describe the tense and the aspect of verbs. However, abbreviating these adverbs and auxiliary words is possible in Chinese. The temporal information processing for Chinese needs more contextual and lexical information. For example:

- 當/你/去/便利/商店/時，幫/我/買/汽水 (when you go to the convenience store, please buy a soft drink for me.)
- 當/我/去/便利/商店/時，看到/他/在/買/汽水 (when I went to the convenience store, I saw he was buying a soft drink.)

Even though the temporal relations (“當/你/去/便利/商店/時 (when you go to the convenience store)” and “當/我/去/便利/商店/時 (when I went to the convenience store)”) in these examples describe different reference time, they have the same syntactic structure in Chinese. To distinguish difference occurrence time in the main clauses (“幫/我/買/汽水 (please buy a soft drink for me)” and “看到/他/在/買/汽水” (I saw he was buying a soft drink)”), the context and lexicon of the main clauses is the only clue. In Chinese, the verb “幫” is able to mean “help” and “do something for me”. The verb “看到” includes two particular meaning “看 (see)” and “到 (been)”. The verb “看到” is usually regarded as a past event[3]. To create rules from these instances to recognize the temporal information is difficult. We want to create a machine learning model that is trained from a temporal relation annotated corpus to analyze the temporal information in Chinese.

1.2.2 Distribution of verbs and temporal expressions in Penn Chinese treebank

In this section, we describe a model that we proposed to deal with the temporal information. First, we investigate the distribution of temporal expressions and events in Penn Chinese Treebank. In the treebank, it has a special tag "*-TMP" to describe a temporal expression. The phrases that describe a time point, a time period and a time direction are tagged with this special tag. Table 1-1 shows the distribution of the temporal expression phrases and examples in the treebank. Most of the temporal expressions are noun phrases, prepositional phrases and localizer phrases. We experimented with the identification of these temporal expressions by a sequence tagging machine learner—Conditional Random Field (CRF)[4] and achieved 92% in precision and 86% in recall of the TMP phrases identification task. However, the temporal phrases in Penn Chinese treebank do not include attributes that can describe the meaning of the temporal phrases.

Table 1-1 also shows the verbs in the treebank. In our research, we focus on verbs and regard them as events. Even though not all verbs are events, we observe that the distribution of temporal phrases and verbs is un-balance. We distinguish the temporal relations "between a temporal expression and an event" from the ones "between two events" to deal with the un-balance distribution.

We assume that temporal relations include anchoring relation from an event to a temporal expression and ordering relation between two events. Intuitively, ordering two events requires a temporal expression that can anchor events on the timeline. However,

---

[3] The suffix "到 (been)" means the completed aspect and past tense in many instances. It is able to be considered that the suffix should be separated from the action prefix "看 (see)". However, this method involves the definition of the concept "word" in Chinese. No final conclusion of this concept is accepted. Therefore, we follow the concept of a word in Penn Chinese treebank and consider the string "看到" is a word.

[4] We used a tool: http://chasen.org/~taku/software/CRF++/, features using in this identifying experiment include the word, the pos and the head word of the focus token.

some events cannot be anchored on the timeline without ordering the events independently. For example, in Figure 1-3, there are one temporal expression “昨天早上六點” (6 A.M. yesterday) and four verb phrases (“起床” (wake up), “吃早餐” (eat breakfast), “搭公車” (by bus), and “上學” (go to school)) in the example sentence. For ordering these events (verbal phrases) on the timeline, we can analyze the temporal relation between an event and a temporal expression. In this example, there is only one temporal expression “昨天早上六點” (6 A.M. yesterday) can be analyzed and it is the anchor time of the event “起床” (wake up). Figure 1-3 (1) describes the temporal relations of the adjacent event pairs[5]. We can recognize some temporal relations by considering not the temporal expression but the event pairs. These temporal relations are: the event “起床” (wake up) occurs before the event “吃早餐” (eat breakfast); the event “吃早餐” (eat breakfast) occurs before the event “搭公車” (by bus); and the events “搭公車” (by bus) and “上學” (go to school) occur at the same time. Figure 1-3 (2) describes the temporal relation between the anchor time “昨天早上六點” (6 A.M. yesterday) and the event “起床” (wake up). This temporal relation can anchor the event on the timeline. Combining the temporal relations in Figure 1-3 (1) and (2), the reader can recognize what happened and when they happened. Therefore, we can divide the process of recognizing the temporal information in the sentence into two steps: (1) recognizing the temporal relation between two events; (2) anchoring the events on the timeline. Extremely, we regard that a reader recognize the situation by only considering the temporal relation between events, even the reader does not know the anchor time of the events. Therefore, we think that to annotate the temporal relation between event pairs is an independent task in temporal information processing.

A general temporal information processing model of our ideal is illustrated in Figure 1-4. For analyzing an input article, first step is to identify the events and the temporal expressions from an article. Second step is to order the events[6] as their temporal order and to anchor the temporal expressions on a timeline ***independently***. Final step is connecting the temporal relations and their observe events. Therefore, we acquire an ordinal event

---

[5] These temporal relations of the adjacent event pairs do not include all recognizable temporal relations.

[6] Here, the ordering of events includes the action “ordering” and “embedding” in section 1.1.1. We do not distinguish them because our temporal ordering processing includes both of these actions.

sequence on a timeline. In this research, we focus on ordering the events. That is, annotate the temporal relation between two events.

TMP tagged phrases	count	example
NP-TMP	4045	"一九九五年九月" (September 1995)
PP-TMP	1816	"自成立以來" (after it established)
LCP-TMP	1733	"近年來" (in recent years)
ALL TMP tag phrases	8254	
Verbs	74089	

Table 1-1: The distribution of temporal phrases in Penn Chinese treebank

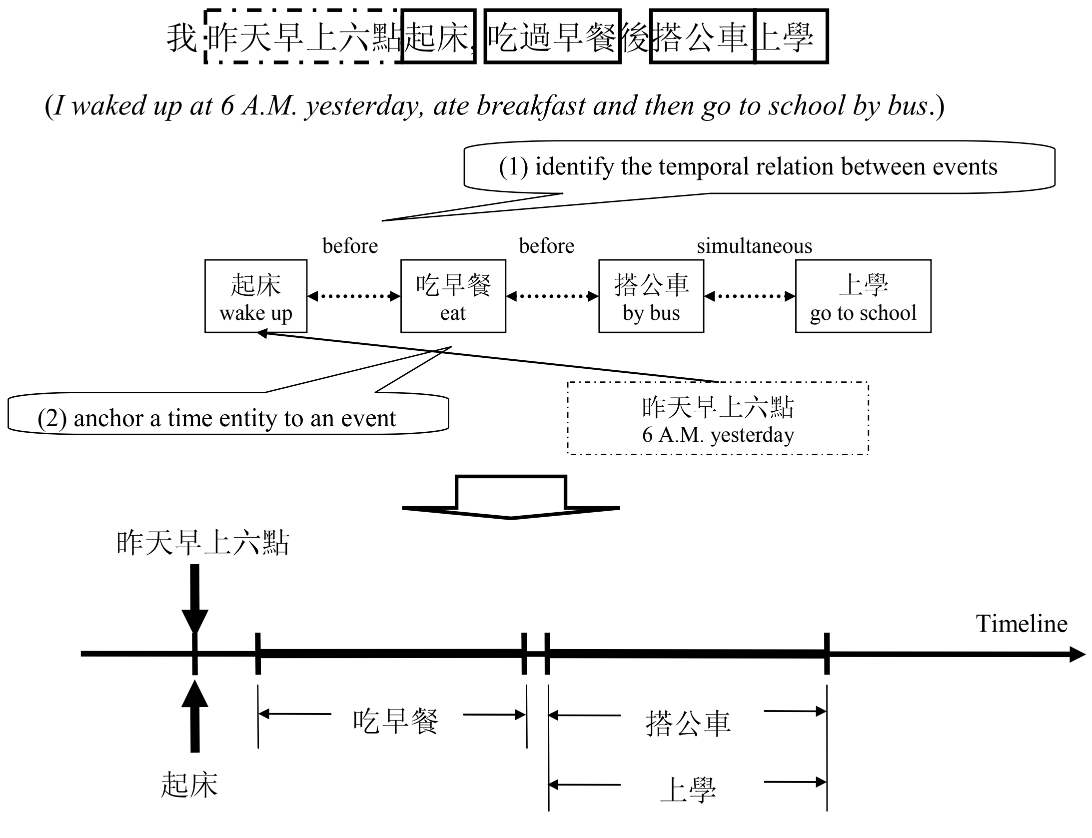


Figure 1-3: **The temporal relations between events and a temporal expression**

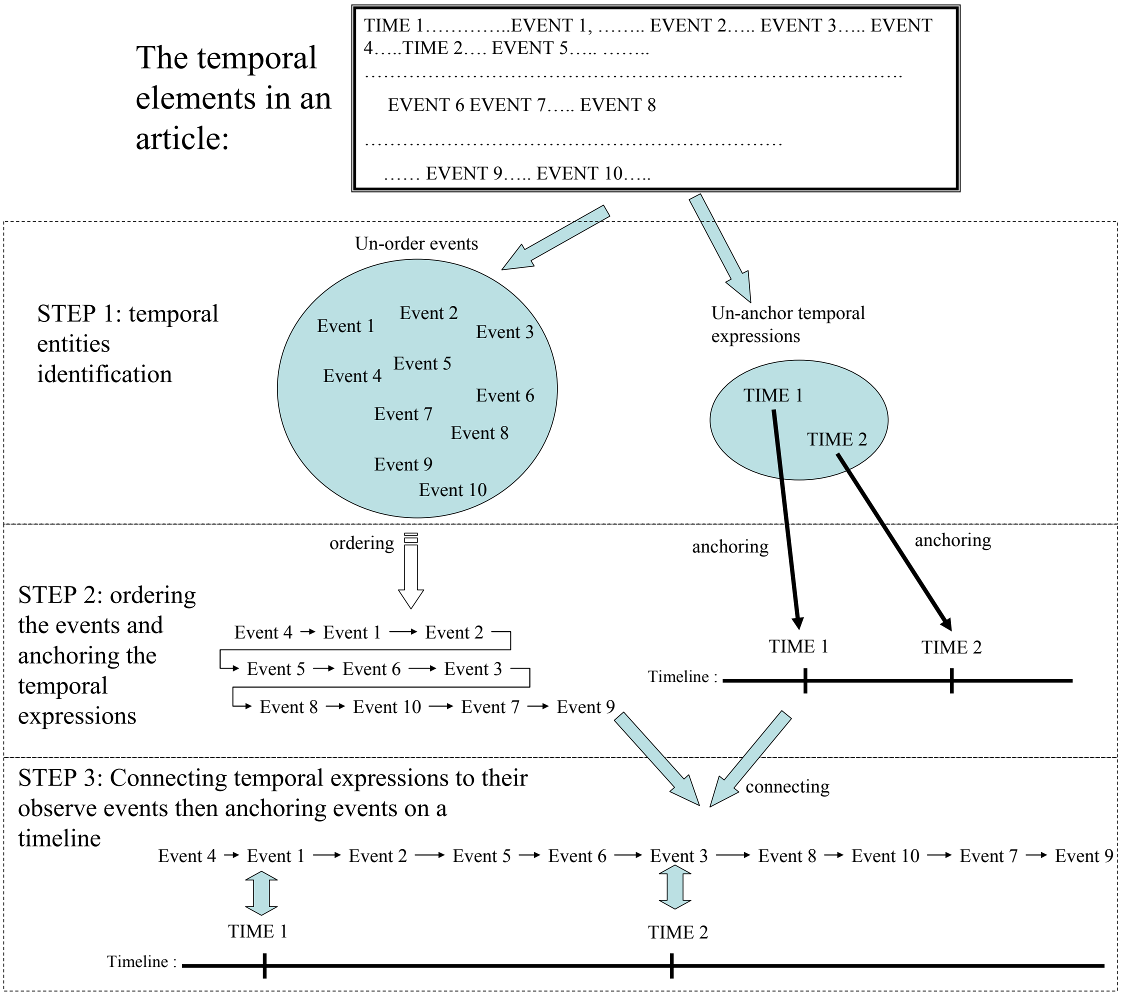


Figure 1-4: A temporal information processing model

1.3 The aims of our research

1.3.1 A dependency structure based temporal relation annotation

The goal of our research is to efficiently construct a temporal relation tagged corpus of Chinese. In English, TimeBank (Pustejovsky, et al., 2006 [67]), a temporal information tagged corpus, is available for introducing machine learning approaches to automatically extract temporal relation. In Chinese, there are some related researches of temporal expression extraction (see section 2.1). However, there is no publicly available resource for temporal information processing in Chinese. Following the description in section 1.2, analyzing the temporal relation between events is an important independently task. We are making such resources; event and temporal relation tagged corpora for creating a machine learning based temporal relation analyzer. Annotating all temporal relations of event pairs is time-consuming. We propose a dependency structure based method to annotate temporal relations manually on a limited set of event pairs and extend the relations using inference rules. This method reduces manual effort. The dependency structure helps to detect subordinate and coordinate structures in sentences (see Chapter 3). We also describe a guideline for corpus annotation. Our annotation guideline is based on TimeML (Saurí, et al., 2005 [70]) (see section 2.2.1). We use a syntactic tagged Chinese treebank (Penn Chinese treebank) (Palmer, et al., 2005 [65]) to create this temporal relation annotated corpus. In section 5.5, we survey the distribution of the temporal relations in our tagged corpus. In section 6.1.2, we evaluate the coverage of the limited event pairs in our criteria.

1.3.2 Constructing a machine learning based temporal relation analyzer

After creating a temporal relation annotated corpus, we construct the temporal relation analyzer. We developed a machine learning based dependency analyzer for Chinese (see Chapter 4) and we have a Chinese morphological analyzer (GOH, 2006 [29]). Our temporal relation analyzer uses the output of the dependency structure analyzer. Our temporal relation analyzer is trained on the temporal relation annotated corpus with support vector machines (see section 2.4). The inference rules are used to extend the temporal relations (see section 3.2.3). The feature selection and the recall of the analyzer are described in Chapter 6.

1.3.3 The contribution of our research

Following the description in section 1.1.2 and 1.2.1, analyzing the temporal relation between events is an independent task in temporal information processing. In Chinese, the temporal relation analysis needs contextual and lexical information. We can create a machine learning model for this task. However, analyzing all combinations of events in an article is inefficiency and the manual effort in annotating work is huge. Our proposed method reduces manual effort in annotating work and the temporal relation analyzer identifies most important relations in the article. This temporal relation analysis system can be applied on many natural languages processing application.

For example, to translate the Chinese sentences in section 1.2.1 to English with correct verb tense, our proposed system can identify the temporal relation between events "去/便利/商店/時 (go to the convenience store)" and "幫/我/買/汽水 (to buy a soft drink for me)" / "看到/他/在/買/汽水 (I saw he was buying a soft drink)". And then the tense of the verb" 買 (to buy)" can be decided in different context.

In an information retrieval system, the relevancy of a query has a temporal aspect from a user's perspective (Alonso et al., 2007 [94]). The more data sources an information retrieval system acquires, the more important temporal aspect can be in the retrieval process. Instead of assuming that the user wants relevant search results implicitly sorted by date, it would be interesting to investigate a system that is aware of time for relevancy and shows search results in a temporal context. Following the description in section 1.1.2 and 1.2.2, many events do not have their monopolized implict temporal expression. To require the answer of the query, identifying the temporal relation between events without the temporal expression recognization to acquire the causal relation is an efficient method. Our proposed system can satisify these motivations.

1.3.4 The track of our research

Figure 1-5 illustrates the track of our research and the overview of this dissertation. A basic motivation of our research is the insufficiency of recognizing temporal information. To deal with this problem, we focus on identifying the temporal relation between two events. We discuss these descriptions in Chapter 1. Following these descriptions and the invesatigation in other related research, we find that the related works have two problems - their temporal identification processing is inefficient and they identify the temporal relations in local context. Chapter 2 describes these investigations.

For dealing with the problems in related works, we propose two methods-"Adopt dependency structure for temporal relation identification" and "Deduce the long distance temporal relations by inference rules". Additionally, we focus on the temporal relations between events. These proposed methods ("Identify the temporal relation between two events", "Adopt dependency structure for temporal relation identification" and "Deduce the long distance temporal relations by inference rules") are the axiom of our efficient identification processing. Chapter 3 describes these proposed methods and introduces the dependency structure.

We construct a temporal relation identification system of Chinese based on our efficient identification processing. The constructing work includes two parts-"Constructing an annotated corpus" and "Constructing the system". Our temporal relation identification system is machine learning based system; therefore we need to construct an annotated corpus. In Chapter 5, we first introduce our proposed annotation guideline for Chinese temporal relation annotated corpus. And then we report the progesses of the corpus annotating work.

Chapter 6 describes the construction of the temporal relation identification system. It includes two parts- "Constructing a dependency analyzer for Chinese" and "Machine learning models for temporal relation identification". We explain the construction of the dependency analyzer for Chinese in Chapter 4. We investigate two algorithms and improvement the analyzer for analyzing the word dependency of Chinese sentence. Other preliminary processes and the machine learning models for temporal relation identification with the estimation of the system are also described in Chapter 6. The summary of our research and the future direction are described in Chapter 7.

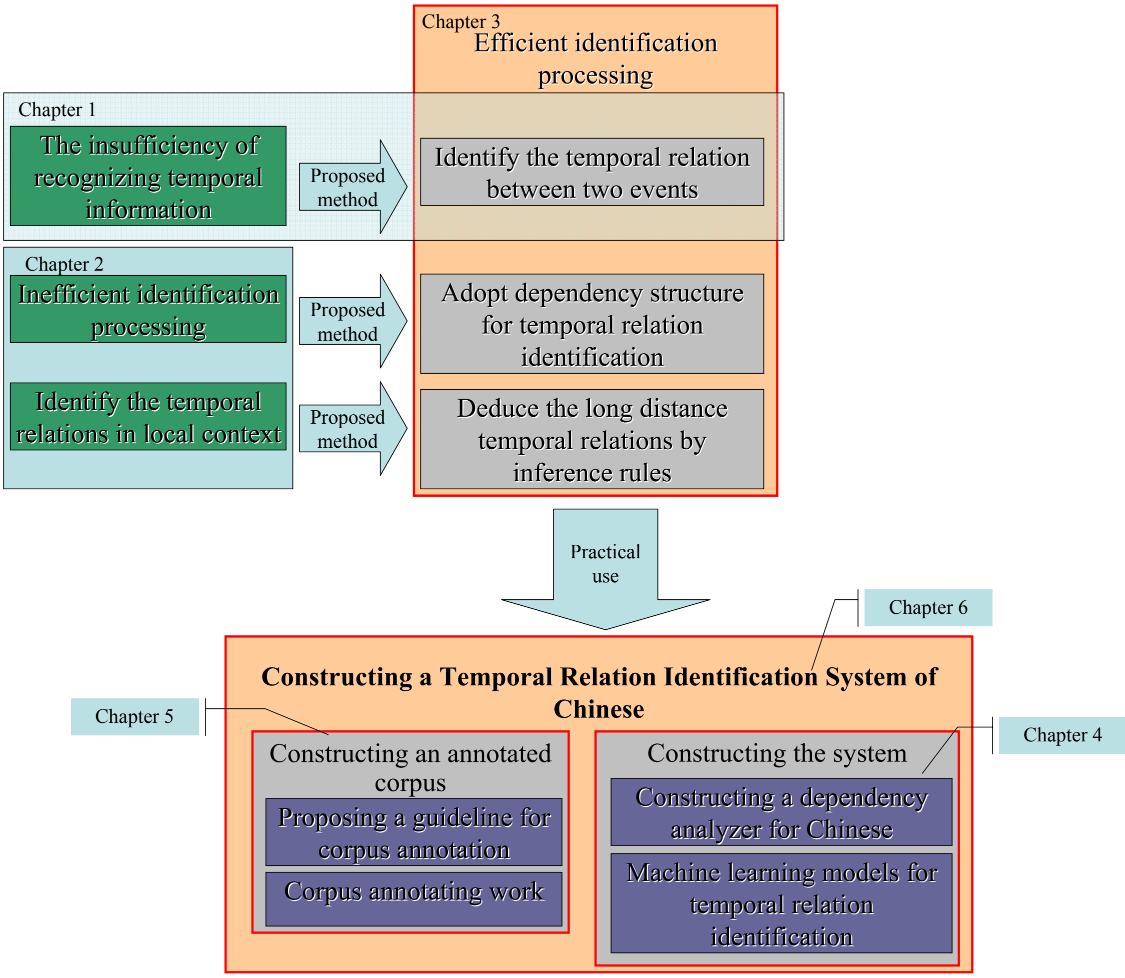


Figure 1-5: An overview of the track of our research

Chapter 2 Preliminary Investigations

In this chapter, we describe several preliminary investigations before we will describe our proposed method in next chapter. First, we review several related researches, especially the researches that deal with temporal relation analysis based on machine learning. Second, we introduce the well-known English temporal information annotating guideline - TimeML and we investigate the distribution in the data of Timebank. Finally, we describe a temporal relation identification shared task- TempEval: Temporal Relation Identification shared task. These investigations are used for designing the annotation criteria. We also introduce the machine learner – support vector machine (SVM) in last section, which is used in all experiments in this thesis.

2.1 Related research of temporal information process

Our research is the first work of the temporal relation identification between verbs in Chinese texts. We cannot compare our research to other similar researches directly. However, the temporal information processing is a studied topic in NLP. Many related researches give us the ideal to deal with the temporal relations in Chinese.

There are several directions for dealing with the temporal information, temporal expressions and event expressions. Extracting temporal expressions is a subtask of NER (IREX committee, 1999 [37]) and widely studied in many languages. Normalizing temporal expressions is investigated in evaluation workshops (Chinchor, 1997 [15]). In recent researches, **Ahn** (Ahn et al., 2007 [83]) proposed a system for identifying and interpreting temporal expressions in timex annotation. In the system, the large set of complex hand-crafted rules standard in systems for this task is replaced by a series of machine learned classifiers and a much smaller set of context-independent semantic composition rules. Instead, they decouple recognition from normalization and factor out context-dependent semantic and pragmatic processing from context independent semantic composition. Their system uses machine learned classifiers to make context-dependent disambiguation decisions, which use a small set of simple, context-independent rules for semantic composition. **Han** (Han et al., 2006 [84]) described a system capable of anchoring such expressions in English: system TEA features a constraint-based calendar model and a compact representational language--TCNL to capture the intensional meaning of temporal expressions. They report favorable results from experiments conducted on several email datasets.

Event semantics is also investigated in linguistics and AI fields (Bach, 1986 [4]). **Noro's** (Noro et al., 2006 [85]) study aimed at identifying when an event written in text occurs. They classified a sentence for an event into four time-slots; morning, daytime, evening, and night. They focused on expressions associated with time-slot (time-associated words). Because there sre numerous time-associated expressions, they used a semi-supervised learning method, the Naïve Bayes classifier backed up with the Expectation Maximization algorithm, in order to iteratively extract time-associated words. They used Support Vector Machines to filter out noisy instances that indicated no specific time period. In order to avoid the class imbalance problem, they used a 2-step classifier, which first filters out time-unknown sentences and then classifies the remaining sentences into one of 4 classes. Their proposed method outperformed the simple 1-step method.

Tsang (Tsang et al., 2002 [86]) investigated the use of multilingual data in the automatic classification of English verbs, and show that there is a useful transfer of information across languages. They experimented with three lexical semantic classes of English verbs. They collected statistical features over a sample of English verbs from each of the classes, as well as over Chinese translations of those verbs. They used the English and Chinese data, alone and in combination, as training data for a machine learning algorithm whose output is an automatic verb classifier. They found that Chinese data is indeed useful in helping to classify the English verbs and a multilingual combination of data outperforms the English data alone.

Zhu (Zhu et al., 2000 [87]) proposed an algorithm is rendered to classify Chinese verbs into different situation types. Because a verb's situation depends on the meaning of the verb, the essence of our algorithm takes advantage of collocations to avoid semantics. The result shows the algorithm is successful. The classification algorithm itself is independent of resources, so it can be applied to other resources (dictionaries) if these resources include sufficient collocation information.

Researches on temporal relation extraction are still limited. Temporal relation extraction includes the following issues: identifying events, anchoring an event on the timeline, ordering events, and reasoning of contextually underspecified temporal expressions. Mani's researches (2000 [88], 2006 [55]) and Li's researches (Li et al., 2004 [48]; Wu et al., 2005 [50]) proposed a system for analyzing all temporal information. Their methods give us several ideals to construct our system.

Mani and Wilson (2000 [88]) proposed an annotation scheme for temporal expressions, and describe a method for resolving temporal expressions in print and broadcast news. The algorithm, which is relatively knowledge-poor, uses a mix of hand-crafted and machine-learned rules and obtains reasonable results against hand-annotated data. Some initial steps towards tagging event chronologies are also described. Then **Mani** (Mani et al., 2003 [89]) proposed a domain-independent approach to temporally anchoring and ordering events in news. The approach is motivated by a pilot experiment with 8 subjects providing news event-ordering judgments which revealed that the narrative convention applied only 47% of the time in ordering the events in successive past-tense clauses. Their approach involves mixed-initiative corpus annotation, with automatic tagging to identify clause structure, tense, aspect, and temporal adverbials, as well as tagging of reference times and anchoring of events with respect to reference times. They report on machine learning results from event-time anchoring judgments. The approach achieves 84.6% accuracy in temporally anchoring events and 75.4% accuracy in partially ordering them.

Mani (Mani et al., 2006 [55]) also proposed a machine learning based method for temporal relation analysis. They investigated a machine learning approach for temporally ordering and anchoring events in natural language texts. They used temporal reasoning as an over-sampling method to dramatically expand the amount of training data, resulting in predictive accuracy on link labeling as high as 93% using a Maximum Entropy classifier on human annotated data. This method compared favorably against a series of increasingly sophisticated baselines involving expansion of rules derived from human intuitions. Their research uncovered one finding: semantic reasoning (in this case, logical axioms for temporal closure), can be extremely valuable in addressing data sparseness. Without it, performance on this task of learning temporal relations is poor; with it, it is excellent. They showed that temporal reasoning can be used as an over-sampling method to expand the amount of training data for TLINK labeling. Their results confirm the lessons learned from the corpus-based revolution, namely that rules based on intuition alone are prone to incompleteness and are hard to tune without access to the distributions found in empirical data. Clearly, lexical rules have a role to play in semantic and pragmatic reasoning from language. Such rules, when mined by robust, large corpus based methods, as in the Google-derived VerbOcean (Chklovsky and Pantel, 2004 [16]), are clearly relevant.

Li (2004 [49]) proposed a computational model based on machine learning and heterogeneous collaborative bootstrapping for analyzing temporal relations in a Chinese multiple-clause sentence. The model makes use of the fact that events are represented in different temporal structures. It takes into account the effects of linguistic features such as tense/aspect, temporal connectives, and discourse structures. The model combines linguistic knowledge and machine learning approaches. Two learning approaches, namely probabilistic decision tree (PDT) and naive Bayesian classifier (NBC) and 13 linguistic features are employed. Due to the limited labeled cases, they used a collaborative bootstrapping technique to improve learning performance. A set of experiments has been conducted to investigate how linguistic features could affect temporal relation resolution.

Wu and Li (Wu et al., 2005 [50]; Li et al., 2004 [48]) presented a temporal parser for extracting and normalizing temporal expressions from Chinese texts. They also propose a temporal framework, which includes basic temporal objects and relations, the measurement and classification of temporal expressions. To cope with kinds of temporal expressions, constraint rules are employed to retrieve genuine expressions and resolve ambiguities. Their temporal parser CTEMP is fully implemented, which is based on the chart parsing and constraint checking scheme. They evaluated the temporal parser on a manually annotated corpus and achieved promising results of F-measures of 85.6% on extent and 76.8% on value.

There has other researches that deal with the temporal information based on TimeML standard. **Boguraev and Ando** (2005 [90]) indicated that reasoning with time needs more than just a list of temporal expressions. They used TimeML for bridging the gap between temporal analysis of documents and reasoning with the information derived from them. They addressed the problem that the small size of the only currently available annotated corpus makes it even harder with a hybrid TimeML annotator, which uses cascaded finite-state grammars (for temporal expression analysis, shallow syntactic parsing, and feature generation) together with a machine learning component capable of effectively using large amounts of un-annotated data.

An important application of temporal information processing is that using these technics to deal with the tense / aspect translation problems in Chinese-English machine translation. **Ye** (Ye et al., 2006 [91]) focused on the task of determining the tense to use when translating a Chinese verb into English; current systems do not perform as well as human translators. Their proposed method is that to identify features that human translator use, but which are not currently automatically extractable. They tested a particular hypothesis about what additional information human translators might be using, and as a pilot to determine where to focus effort on developing automatic extraction methods for features that are somewhat beyond the reach of current feature extraction. They showed that incorporating several latent features into the tense classifier boosts the tense classifier’s performance, and a tense classifier using only the latent features outperforms one using only the surface features. They confirm the utility of the latent features in automatic tense classification, explaining the gap between automatic classification systems and the human brain.

Hacioglu (Hacioglu et al., 2005 [92]) described systems for automatic labeling of time expressions occurring in English and Chinese text as specified in the ACE Temporal

Expression Recognition and Normalization (TERN) task. They cast the chunking of text into time expressions as a tagging problem using a bracketed representation at token level, which takes into account embedded constructs. They used a left-to-right, token-by-token, discriminative, deterministic classification scheme to determine the tags for each token. A number of features are created from a predefined context centered at each token and augmented with decisions from a rule-based time expression tagger and/or a statistical time expression tagger trained on different type of text data, assuming they provide complementary information. They trained one-versus-all multi-class classifiers using support vector machines.

Finally, our research includes a guideline of the temporal relation annotation. We refer to several related researches that deal with the temporal information annotation to establish our guideline. Recent works on the annotation of event and temporal relations have resulted in both a de-facto standard for expressing these relations (TimeML) and a hand-built gold standard of annotated texts (TimeBank). These have already been used as the basis for automatic Time and Event annotation tasks in a number of research projects in recent years. We describe TimeML and Timebank in section 2.2 and describe a shared task that uses Timebank to deals with the temporal relation identification.

2.2 The guideline and corpus for temporal information processing—TimeML and Timebank

In this section, we describe an important related research—TimeML— that defines a temporal information annotation guideline. We investigate the distribution of the tags in the temporal relation tagged corpus—Timebank, which is tagged by TimeML standard. We first introduce TimeML standard then describe our investigation in Timebank. Second, we investigate the distribution of temporal information in the timebank.

2.2.1 The corpus annotating guideline

TimeML (Sauri et al., 2006 [70]) is a corpus guideline of temporal information for English news articles. Table 2-1 lists the attributes of each tag. They include temporal entities (EVENT, MAKEINSTANCE, SIGNAL and TIMEX3) and temporal links (TLINK, SLINK and ALINK). The tags are described in XML format.

"EVENT", "MAKEINSTANCE", "TIMEX3" and "SIGNAL" tags in TimeML mark up the temporal entities such as event expressions, temporal expressions and clue expressions to identify temporal relations. The definition of each temporal entity is:

- EVENT: Situations that "happen" or "occur", includes tensed / un-tensed verbs, nominalizations, adjectives, predicative clauses or prepositional phrases.
- TIMEX3: Temporal expressions, includes date, time and duration.
- SIGNAL: Textual elements that make explicit the relation holding between two entities.
- MAKEINSTANCE: To create the actual realizations of an event.

The attribute "class" in the tag "EVENT" is a classification of different situations in event expression. It can be regarded as the semantic role of an event. This classification corresponds to the event types and sub-ordinate attribute in our annotation criteria (see section 5.2, 5.3 and 5.4). The tag "TIMEX3" includes all temporal expressions in articles. The definition of this tag referred to another temporal expression annotation—TIDES TIMEX2 annotation (Ferro et al., 2002 [27]). The attributes of this tag include the time point and time interval of a temporal expression. The attribute "functionInDocument" describes the relation between the temporal expression and the document related time (not an event). For example, "document publication time" is a document related time. The tag "SIGNAL" could exist between timex and event, timex and timex, or event and event. "SIGNAL" is generally temporal prepositions, temporal conjunctions, prepositions signaling modality and several special characters. A special viewpoint of TimeML is that they distinguish "EVENT" and "MAKEINSTANCE" as two different temporal entities. The tag "EVENT" only represents event expressions in the article. An actual realization of an event will be created by the tag "MAKEINSTANCE". The motivation is examples like "*John taught on Monday and Tuesday*", where one verb represents two events—"*taught on*

Monday" and "t*aught on Tuesday*". Several attributes of this tag describe more information of the actual realization, such as the tense, aspect, POS[7], and polarity.

Link tags annotate the temporal relations between entities. The definitions of identifying each temporal relation links are:

- TLINK: Temporal relation links, represent the temporal relationship holding between two temporal entities (a TIMEX3 and a TIMEX3, an event instance and an event instance, or a TIMEX3 and an event instance)
- SLINK: Subordinate links, represent contexts introducing relations between two events
- ALINK: Aspectual links, represent the relationship between an aspectual event and its argument event

The tag "TLINK" represents the temporal relationship between two tagged entities. The definitions of temporal relation types in the tag "TLINK" (corresponds to the attribute "relType") is based on Allen's (1983 [1]) temporal relations. We will compare the temporal relation types of Allen's research, TimeML, and our criteria in section 5.4.1 and Figure 5-6. The tags "SLINK" and "ALINK" annotate the relations between a main event and its subordinate event. While the tag "ALINK" describes an aspectual relation, the tag "SLINK" describes a subordinate relation without explicit aspectual meaning. To annotate "SLINK" and "ALINK", the TimeML guideline suggests the annotator to consider two viewpoints—lexical-based and structural-based (includes purpose clauses and conditional constructions). This suggestion indicates that using syntactic viewpoint to analyze the relation between events is important.

[7] The POS in the tag "MAKEINSTANCE" is different from the normal definition in morphological analysis. It only distinguish the roughly category of the event word.

Tags	Attributes and values
Temporal entities	
EVENT	eventID, class={REPORTING | PERCEPTION | ASPECTUAL | I_ACTION | I_STATE |STATE | OCCURRENCE}
TIMEX3	TimeID, type= {DATE | TIME | DURATION | SET}, beginPoint, endPoint, quant, freq, functionInDocument= {CREATION_TIME | EXPIRATION_TIME | MODIFICATION_TIME | PUBLICATION_TIME | RELEASE_TIME| RECEPTION_TIME | NONE}, temporalFunction= {true | false}, value= {duration | dateTime | time | date | gYearMonth | gYear | gMonthDay | gDay | gMonth}, valueFromFunction, mod= {BEFORE | AFTER | ON_OR_BEFORE | ON_OR_AFTER | LESS_THAN | MORE_THAN | EQUAL_OR_LESS |EQUAL_OR_MORE | START | MID | END | APPROX}, anchorTimeID,
SIGNAL	signalID
MAKEINSTANCE	eventinstanceID, eventID, tense= {PAST | PRESENT | FUTURE | NONE | INFINITIVE |PRESPART | PASTPART}, aspect={PROGRESSIVE | PERFECTIVE | PERFECTIVE_PROGRESSIVE | NONE}, pos = {ADJECTIVE | NOUN | VERB | PREPOSITION | OTHER}, polarity= {NEG | POS}, modality, signalID, cardinality
Link	
TLINK	LinkID, eventInstanceID, timeID, signalID, relatedToEventInstance, relatedToTime, relType = {BEFORE | AFTER | INCLUDES | IS_INCLUDED | DURING | DURING_INV | SIMULTANEOUS | IAFTER | IBEFORE | IDENTITY | BEGINS | ENDS | BEGUN_BY | ENDED_BY}
SLINK	LinkID, eventInstanceID, subordinatedEventInstance, signalID, relType= {MODAL | EVIDENTIAL | NEG_EVIDENTIAL | FACTIVE | COUNTER_FACTIVE | CONDITIONAL}
ALINK	LinkID, eventInstanceID, signalID, relatedToEventInstance, relType={INITIATES | CULMINATES | TERMINATES | CONTINUES | REINITIATES}

Table 2-1: Tags and their attributes in TimeML annotation

2.2.2 The data analysis of TimeBank

TimeBank (Pustejovsky, et al., 2006 [67]) is a temporal information tagged corpus of English that includes full temporal information (temporal expressions, events and temporal relations). The corpus is annotated by the TimeML guideline. In this section, we investigate the distribution of tags (events, temporal expressions and all kind of links) in TimeBank. We find that the distribution of events and temporal expressions is uneven. Therefore, our proposed temporal relation annotation criteria do not focus on the relations between an event and a temporal expression, but between two events.

Considering the distribution of the temporal entities in TimeBank (see the upper part of Table 2-2), the number of Events (7940) are more than the number of temporal expressions (TIMEX3, 1414). Similar to the discussion in section 1.2.2, many events share a temporal expression or should be analyzed as the temporal relation of event pairs with no corresponding temporal expression when ordering the events on a timeline. Only a part of events in an article have their own temporal expression (phrases). The other events do not have direct temporal expression to anchor the events on the timeline. If we consider the verbs as the events in Treebank, most of temporal relations are not between a temporal expression and an event, but between two events. It is necessary for recognizing the temporal information that to analyze the temporal relation between events that are not grounded onto any time expression.

TimeBank 1.2[8] contains 183 articles with over 61,000 non-punctuation tokens. The distribution of temporal relation link tags is shown in the lower part of Table 2-2. Following the definition of TLINK, it includes the temporal relations between events and temporal expressions. We distinguish the TLINK tag into normal TLINKs and event TLINKs. TimeBank includes 9615 links (TLINK with all kinds of temporal entities, SLINK, and ALINK), of which, 5763 links are the relations between the adjacent entity pairs (an adjacent pair means the focus event and its linearly preceding event). The tag "TLINK" includes the temporal relations between document creation time and other temporal entities in an article. These TLINKs are long distance links and cannot be identified by the adjacent relation and the dependency structure. An adjacent relation may be able to be a head-modifier relation simultaneously. Therefore the numbers of the column "adjacent relations" and the column "head-modifier relations" in the table are not exclusionary. The column "all relations" is not the sum of "adjacent relations", "head-modifier relations" and "ancestors- descendant relations". Furthermore, we only consider the dependency structure of "sentences". If the relation links in Timebank cross different sentences, the dependency structures cannot recognize these links. The remnants of SLINKS that are not head-modifier relations are the links crossing different sentences.

---

8 http://www.ldc.upenn.edu/

The distribution of all links shows that if we are able to recognize adjacent relations correctly (at least 60% (5763/9615) of temporal relations are recognized), we expect to acquire more temporal relations with an additional process, such as adaptation of inference rules that we will describe in section 3.2.3. We refer to the links of adjacent relations as "adjacent links". To recognize the adjacent links of events, we annotate adjacent event pairs. Additionally, we can find that approximately forty percent (2296/5763) of the links in the adjacent links are SLINKs and ALINKs. If we consider to the relation between events, the subordinate links in all adjacent links are approximately fifty percent (2296/4053). Again, the tags "SLINK" and "ALINK" mean subordinate relation between events (not from an event to a temporal expression). Subordinate relation is the relation between a focus event and a main event that the focus event depends on. Therefore we can count these links in the same category. SLINKs and ALINKs do not include TLINKs. If we extract SLINKs and ALINKs first, to extract other TLINKs from the remaining temporal entities would become simple. This observation gives us the idea that recognition of subordinate relations is an important task for annotating adjacent relations.

Since the majority of adjacent links are subordinate relations, we cannot analyze the temporal relations between contiguous pairs of matrix verb events without analyzing the structure of the subordinate relations, namely dependency structure. For calculating the distribution of links in TimeBank using dependency structure, we parse the sentences in Timebank into the dependency structure and estimate the number of head-modifier (governor-dependent) relations that are SLINK or ALINK. We use the POS-tagger "TnT" (Brants, 2000 [6]) to tag the sentences and use the MST parser[9] (McDonald, et al., 2005 [59]) to parse sentences to the dependency structures. The column "Head-modifier relations" in Table 2-2 shows the number of each type of links that is a head-modifier relation. The column "ancestor- descendant relations" describes the event pairs that are ancestors- descendant relations [10] in dependency structures. Seventy-three percent (2331/3197) of S/ALINKs (SLINK + ALINK) in TimeBank are of head-modifier relations. The percentage of ancestors- descendant relations in S/ALINKs is similar to the

[9] We train the MST parser using Penn Treebank (Marcus et al. 1993 [58]).

[10] The ancestor- descendant relation is that a focus event and its related event is not head-modifier pair, but is in the same path from a leaf to the root of dependency structure.

head-modifier relations because subordinate relations are all head-modifier relations in dependency structures. This shows that dependency structure can be used to extract most S/ALINKs in English articles.

The related research also shows that syntactic (dependency) information is useful for temporal information extraction (Li, et al., 2004 [49]; Mani, 2006 [55]). We use dependency structure for annotating temporal relation. The reason is that dependency structures are simpler and more comprehensible than phrase structures. The dependency grammar is composed of asymmetric head-modifier relations between words. We focus on the relation of event pairs. Dependency structure can describe the semantic relation between events clearly. The subordinate relations can be identified by the dependency structure. Therefore dependency structure analysis is very useful for annotating the temporal relation.

<table>
<tr><td colspan="6">Distribution of temporal entities tags</td></tr>
<tr><td colspan="2">Tags</td><td>EVENT</td><td>MAKEINSTANCE</td><td>TIMEX3</td><td>SIGNAL</td></tr>
<tr><td colspan="2">Number</td><td>7935</td><td>7940</td><td>1414</td><td>688</td></tr>
<tr><td colspan="6">Distribution of temporal links in adjacent and dependency structure viewpoints</td></tr>
<tr><td></td><td>Entities</td><td>all links</td><td>adjacent relations</td><td>head-modifier relations</td><td>adjacent and head-modifier relations</td></tr>
<tr><td rowspan="2">TLINK</td><td>Timex3 and event</td><td>6418</td><td>3467</td><td>1372</td><td>4458</td></tr>
<tr><td>Event and event</td><td>3314</td><td>1757</td><td>1186</td><td>2826</td></tr>
<tr><td>SLINK</td><td>Event and event</td><td>2932</td><td>2129</td><td>2174</td><td>2833</td></tr>
<tr><td>ALINK</td><td>Event and event</td><td>265</td><td>167</td><td>157</td><td>251</td></tr>
</table>

Table 2-2: Distribution of tags in TimeBank

2.3 TempEval: Temporal Relation Identification shared task

In this section, we describe a shared task that deals with temporal relation in English articles (Verhagen et al., 2007 [73])[11]. We first introduce the shared task. This shared task specifies three separate tasks that involve identifying event-time and event-event temporal relations. Second, we describe our system for this shared task and the results of the system. We attempt to use a sequence labeling model with features from dependency parsed tree for temporal relation identification. Finally, we refer to other participants' methods.

2.3.1 Shared Task Description

This shared task specifies three separate temporal relation identification tasks. Given a set of test texts (DataSet1) for which, sentence boundaries are annotated, all temporal expressions are annotated in accordance with TIMEX3, the document creation time (DCT) is specially annotated, and a list of root forms of event identifying terms (the Event Target List or ETL) is supplied, complete the following tasks

- **Task A:** For each event, whose root form occurs in the ETL, link this event to time expressions in the same sentence as appropriate using a restricted set of temporal relations
- **Task B:** For each event whose root form occurs in the ETL, link this event to the DCT as appropriate using a restricted set of temporal relations
- **Task C:** For each contiguous pair of matrix verbs link the events signalled by these verbs as appropriate using a restricted set of temporal relations. For task C a separate set of test texts (DataSet2) is supplied which is annotated as is DataSet1,

[11] This shared task is hold on the workshop "SemEval-2007: 4th International Workshop on Semantic Evaluations" in 45th Annual Meeting of the Association for Computational Linguistics

and in addition is annotated to identify the main verb in the matrix clause ("matrix verb") of each sentence.

The temporal relations of this shared task contains: BEFORE, AFTER, and OVERLAP (defined to encompass all cases where event intervals have non-empty overlap). In addition, organizers allow three relaxed relations: BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE (for completely underspecified relations). For tasks A and B, in cases where there are multiple time expressions in the sentence, the event should be linked to all appropriate TIMEXs. For the ETL we propose to use those terms whose variants in all inflected forms occur as events in TimeBank 20 times or more, which yields a list of around 63 root forms whose variants are included. Task C is the most ambitious in the three tasks proposed one which we view as exploratory in nature. Given the challenges it presents we would not expect all participants to attempt it.

The data set of this shared task is TimeBank (183 documents, approx. 2500 sentences) which has TimeML annotations. The test corpus will consist of a number of articles not currently included within TimeBank, which will be annotated in accordance with the schemes outlined above. For tasks A and B, it is intended that this should include at least 5 occurrences for each item in the ETL. For task C, we propose to annotate around 20-25 news articles (including of the order of 200-250 sentences) drawn from sources similar to those used for TimeBank.

2.3.2 Our proposal system for the shared task

In our proposed system for the shared task (Cheng et al., 2007 [11]), we attempt to use a sequence labeling model with features from dependency parsed tree for temporal relation identification. In the sequence labeling model, the relations of contextual pairs can be used as features for relation identification of the current pair. Head-modifier relations between pairs of words within one sentence can be also used as the features. These features are effective for the temporal relation identification tasks.

Our proposed system for the shared task has two charactaristics: sequence labeling model and use of dependency parsed tree. Firstly, we treated each problem a sequence labeling problem, such that event/time pairs were ordered by the position of the events and times in the document. This idea is for task B and C. In task B, the adjacent relation between an

EVENT and the DCT-TIMEX3 tends to interact. In task C, when EVENT-a, EVENT-b, and EVENT-c are linearly ordered, the relation between EVENT-a and EVENT-b tends to affect the one between EVENT-b and EVENT-c.

Secondly, in this shared task, we introduced dependency features where each word was annotated with a label indicating its tree position to the event and the time, e.g. “descendant” of the event and “ancestor” of the time. The dependency features are introduced for our machine learning-based relation identifier. In task A, we need to label several different event-time pairs within the same sentence. We can use information from TIMEX3, which is a descendent of the target EVENT in the dependency tree.

Our approach to identify temporal relation is based on a sequence labeling model. The target pairs are linearly ordered in the texts. Sequence labeling model can be defined as a method to estimate an optimal label sequence. The sequence labeling approach is natural for task B and C. In task B, if a document is about affairs in the past, the relations between events and a document creation time tend to be “BEFORE”. All relations in task B depend on each other. In task C, if a relation between the preceding event and the current one is “AFTER”, the current one is in the past. The information helps to determine the relation between the current and succeeding one. Whereas we have reasonable explanation to introduce sequence labeling for task B and C, we cannot for task A. However, in our preliminary experiments with trial data, sequence labeling models outperformed point-wise models for task A. Thus, we introduce sequence labeling model for task A

Now, we present sequence labeling model for each task in detail by Figure 2-1. The left parts of figures are the graphical models of the sequence labeling. The right parts are the tagged corpus: <S> and </S> are sentence boundaries; a EVENT-*nn* denotes an EVENT; a TIME-*nn* denotes a TIMEX3; a TIME-DCT denotes a TIMEX3 with document creation time; an italicized boldface EVENT-*nn* denotes a matrix verb event of the sentence.

For task A in Figure 2-1, x is a sequence of pairs between an EVENT and a TIMEX3 in the same sentence. y is a sequence of corresponding relations. Event-time pairs are ordered first by sentence position, then by event position and finally by time position. For task B in Figure 2-1, x is a sequence of pairs between an EVENT and a DCT-TIMEX3. y is a sequence of corresponding relations. All pairs in the same text are linearly ordered and connected. For task C in Figure 2-1, x is a sequence of pairs between two matrix verb

EVENTs in the neighboring sentences. *y* is a sequence of corresponding relations. All pairs in the same text are linearly ordered and connected, even if the two relations are not in the adjacent sentences.

The dependency structures in Figure 2-2 show the dependency parsing result of the following sentence –"*The warrants may be exercised until 90 days after their issue date*". We also parsed the TimeEval data using MSTParser (McDonald et al., 2005 [59]), which is trained with all Penn Treebank (Marcus et al., 1993 [53]) without dependency label. We introduce tree position labels between a target node and another node on the dependency parsed tree: ANC (ancestor), DES (descendant), SIB (sibling), and TARGET (target word). The left-upper part in Figure 2-2 shows the labels, in which the box with double lines is the target node. The tree position between the target EVENT and a word in the target TIMEX3 is used as a feature for our machine learning-based relation identifier. We also use the words in the sentence including the target entities as features. Each word is annotated with (1) its tree position to the EVENT, (2) its tree position to the TIMEX3, and (3) the combination of the labels from (1) and (2). The labels of tree positions are shown in Figure 2-2. The right-upper picture illustrates (1) EVENT-based labels of the tree position with the target EVENT "*exercised*". The left-lower picture illustrates (2) TIMEX3-based ones with the target TIMEX3 "*90 days*". The right-lower picture illustrates (3) JOINT ones which are combinations of the relation label with the EVENT and with the TIMEX3. We perform feature selection on the words in the current sentence according to the tree position labels. Note that, when MSTparser outputs more than one tree for a sentence, we introduce a meta-root node to bundle the ones in a tree.

The attributes *value* in TIMEX3 is encoded as the relation with DCT-TIMEX3: {BEFORE, OVERLAP, AFTER, VAGUE}. In task A, only words in the current sentence with JOINT relation labels "TARGET/*nn*" or "ANC/*nn*" or "*nn*/DES"[12] are used. In task C, attributes in the TIMEX3 are annotated with the flag whether the TIMEX3 entity is the highest (namely the nearest to the root node) in the dependency parsed tree. Some adverbs and conjunctions in the succeeding sentence help to determine the adjacent two relations. Thus, we introduce all words in the succeeding sentence for Task A and B.

---

[12] "*nn*" stands for wild cards.

The upper part of Table 2-3 is our results on the shared task. In the evaluation of temporal relations in this shared task, it defines a weight factor for giving partial credit for disjunctions, but not so much that non-commitment edges out precise assignments. The weights of each situation are shown in the left-lower part of Table 2-3. For example, assigning VAGUE as the relation type for every temporal relation results in a precision of 0.33. The evaluation without the weights is the "strict", and the evaluation using the weights is the "relaxed" lattices.

Our system is average rank in task A and B, it is the worst mark in task C. The features from dependency parsed trees are effective for task A and B. However, these are not for task C. Therefore, we will focus on what goes wrong instead of what goes right in our preliminary experiments in trial data. We tried point-wise methods with other machine learners such as maximum entropy and multi-class support vector machines. However, sequence labeling method with HMM SVM (Altun et al., 2003 [3]) outperformed other point-wise methods in the trial data. We have dependency parsed trees of the sentences. Naturally, it would be effective to introduce point-wise tree-based classifiers such as Tree Kernels in SVM (Collins and Duffy, 2002 [22]; Vishwanathan and Smola, 2002 [75]) and boosting for classification of trees (Kudo and Matsumoto, 2004 [45]). We tried a boosting learner[13] which enables us to perform subtree feature selection for the tasks. However, the boosting learner selected only one-node subtrees as useful features. Thus, we perform simple vector-based feature engineering on HMM SVM. We believe that it is necessary for solving task C to incorporate knowledge of verb-verb relation. We also tried to use features in verb ontology such as VERBOCEAN (Chklovsky and Pantel, 2004 [16]) which is used in (Mani et al., 2006 [55]). It did not improved performance in our preliminary experiments with trial data.

This shared task can be considered as a proving ground of our temporal relation analyzing system. However, it should be noted that the focus of the shared task is to identify the assigned relations. That is, the test data expresses clearly which element pairs have temporal relations. Participants only identify the relation values of the assigned un-identify relation links. Our temporal relation analyzing system annotates the relation attributes of all events, therefore our system also needs to identify whether an event pair

---

[13] http://chasen.org/~taku/software/bact/

has understandable temporal relation or not. We cannot adopt the proposed system for this shared task instead of another novel viewpoint for temporal relation analysis. We describe our proposed method in next chapter. However, the experiments of the shared task give us ideas for constructing our system that use of dependency location feature is effective in our temporal relation analyzer.

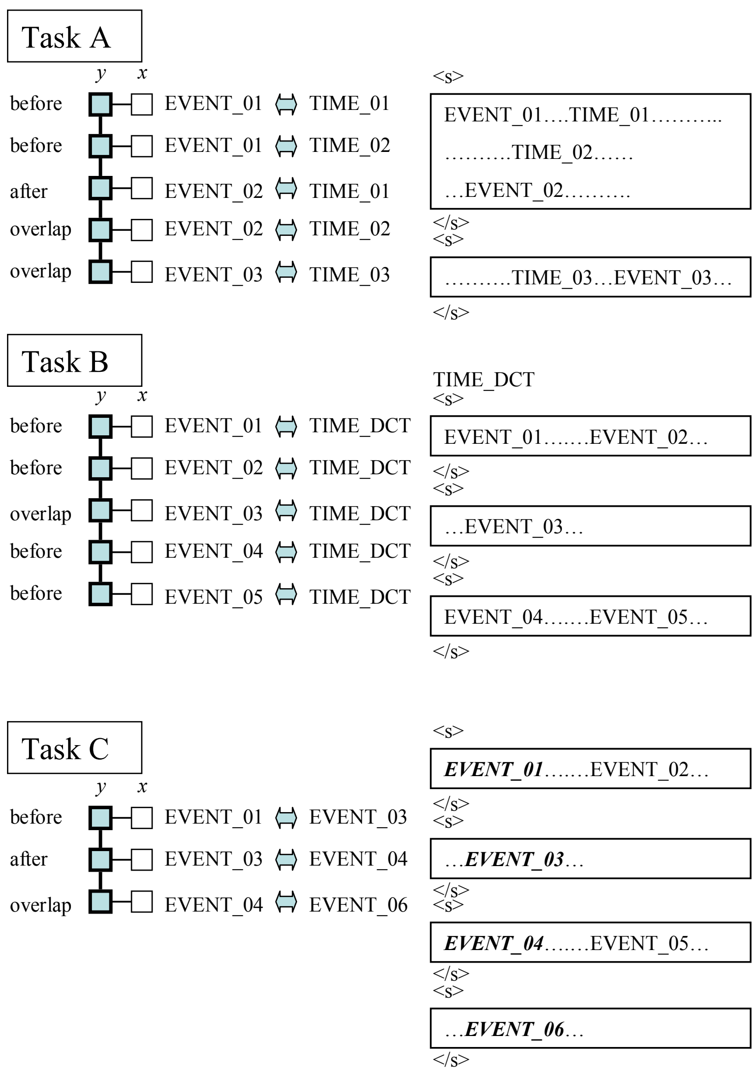
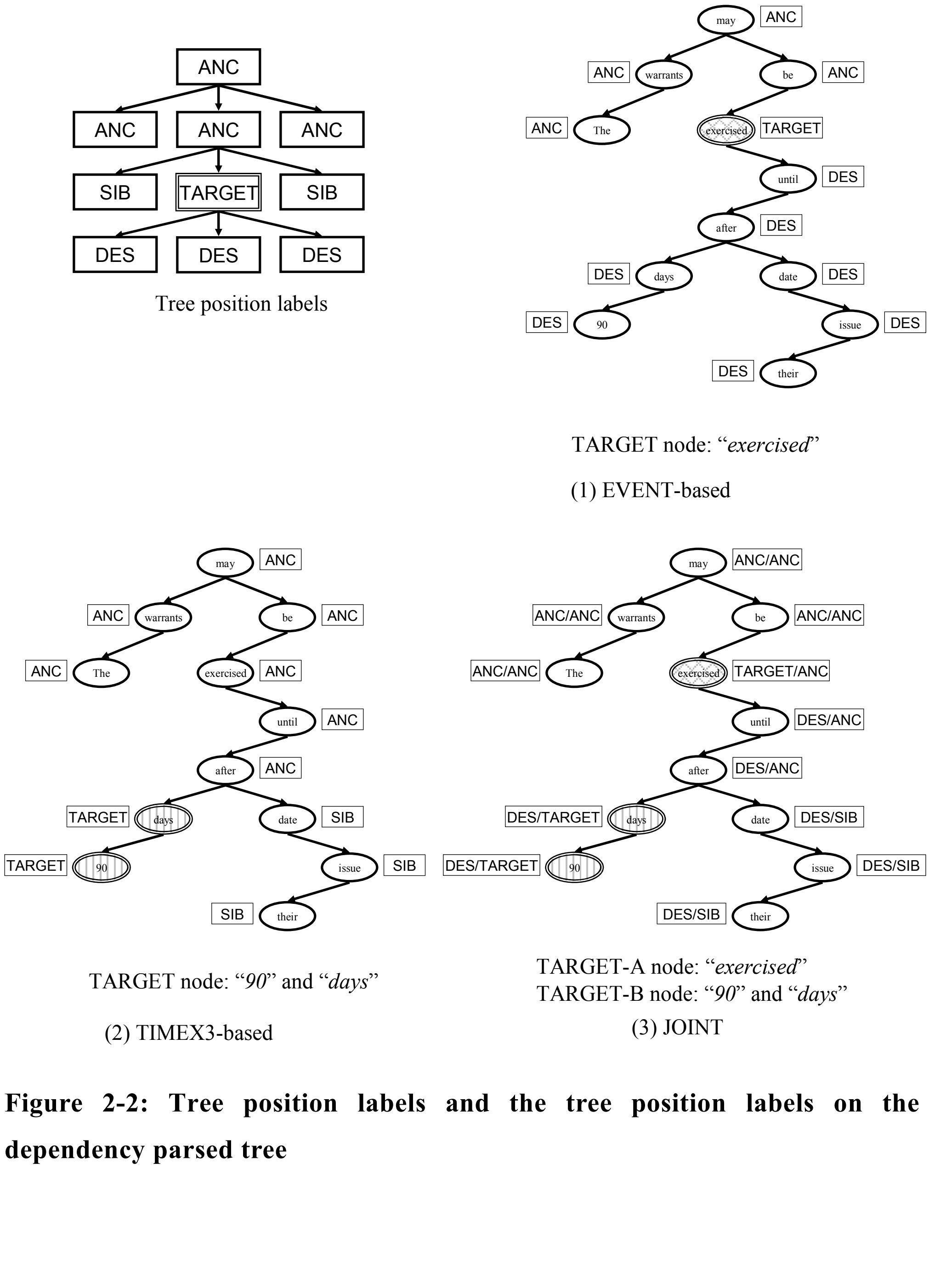


Figure 2-1: Sequence Labeling Models for Task A, B and C



Tree position labels

TARGET node: "*exercised*"

(1) EVENT-based

TARGET node: "*90*" and "*days*"

(2) TIMEX3-based

TARGET-A node: "*exercised*"
TARGET-B node: "*90*" and "*days*"

(3) JOINT

Figure 2-2: Tree position labels and the tree position labels on the example dependency parsed tree

Task	strict				Relaxed			
	P	R	F	Rank	P	R	F	Rank
Task A	0.61	0.61	0.61	2/6	0.63	0.63	0.63	2/6
Task B	0.75	0.75	0.75	2/6	0.76	0.76	0.76	2/6
Task C	0.49	0.49	0.49	5/6	0.56	0.56	0.56	6/6

Weight for scoring						
	B	O	A	B-O	O-A	V
B	1	0	0	0.5	0	0.33
O	0	1	0	0.5	0.5	0.33
A	0	0	1	0	0.5	0.33
B-O	0.5	0.5	0	1	0.5	0.67
O-A	0	0.5	0.5	0.5	1	0.67
V	0.33	0.33	0.33	0.67	0.67	1

- strict scoring
  - Precision: SYScorrect / SYS
  - Recall: SYScorrect / GOLD
- relaxed scoring
  - Precision : w * SYScorrect / SYS
  - Recall: w * SYScorrect / GOLD

Table 2-3: Results of the shared task and the weight for scoring

2.3.3 Other systems for the shared task

Six teams participated in the TempEval tasks (Verhagen et al., 2007 [73]). Three of the teams used statistics exclusively, one used a rule-based system and the other two employed a hybrid approach. This section gives a short description of other participating systems.

CU-TMP (Bethard and Martin, 2007 [5]) trained three support vector machine (SVM) models, one for each task. All models used the gold-standard TimeBank features for events and times as well as syntactic features derived from the text. Additionally, the relation types obtained by running the task B system on the training data for Task A and Task C, were added as a feature to the two latter systems. A subset of features was selected using cross-validations on the training data, discarding features whose removal improved the cross validation F-score. When applied to the test data, the Task B system was run first in order to supply the necessary features to the Task A and Task C systems.

LCC-TE (Min, Srikanth and Fowler, 2007 [60]) automatically identifies temporal referring expressions, events and temporal relations in text using a hybrid approach, leveraging various NLP tools and linguistic resources at LCC. For temporal expression labeling and normalization, they used a syntactic pattern matching tool that deploys a large set of hand-crafted finite state rules. For event detection, they used a small set of heuristics as well as a lexicon to determine whether or not a token is an event, based on the lemma, part of speech and WordNet senses. For temporal relation discovery, LCC-TE used a large set of syntactic and semantic features as input to a machine learning components.

The **USFD** system (Hepple, Setzer and Gaizauskas, 2007 [32]) uses an off-the-shelf Machine Learning suite (WEKA), treating the assignment of temporal relations as a simple classification task. The features used were the ones provided in the TempEval data annotation together with a few features straightforwardly computed from the document without any deeper NLP analysis.

WVALI's (Puscasu, 2007 [66]) approach for discovering intrasentence temporal relations relies on sentence-level syntactic tree generation, bottom-up propagation of the temporal relations between syntactic constituents, a temporal reasoning mechanism that relates the two targeted temporal entities to their closest ancestor and then to each other, and on conflict resolution heuristics. In establishing the temporal relation between an event and the Document Creation Time (DCT), the temporal expressions directly or indirectly linked to that event are first analyzed and, if no relation is detected, the temporal relation with the DCT is propagated top-down in the syntactic tree. Inter-sentence temporal relations are discovered by applying several heuristics and by using statistical data extracted from the training corpus.

XRCE-T (Hagège and Tannier, 2007 [30]) used a rule-based system that relies on a deep syntactic analyzer that was extended to treat temporal expressions. Temporal processing is integrated into a more generic tool, a general purpose linguistic analyzer, and is thus a complement for a better general purpose text understanding system. Temporal analysis is intertwined with syntactic semantic text processing like deep syntactic analysis and determination of thematic roles. TempEval specific treatment is performed in a post-processing stage.

2.4 Machine learner: Support Vector Machines

All the experiments in our research, includes dependency structure analysis and temporal relation identification, use support vector machines (SVMs) (Vapnik, 1998 [72]) as the machine learner. More details (such as feature selection and the labels of classifications) will be describes in the following chapters. We introduce basic description of SVMs in this section.

SVMs are a binary classifier based on a maximum margin strategy that search for hyperplanes with the largest margin between positive and negative samples (see Figure 2-3). Suppose we have a set of training data for a binary classification problem: $(\boldsymbol{x_1}, \boldsymbol{y_1})...(\boldsymbol{x_n}, \boldsymbol{y_n})$, where $\boldsymbol{x_i} \in R^n$ is the feature vector of the i-th sample in the training data and $y_i \in \{+1,-1\}$ is the label of the sample. The goal is to find a decision function $f(x) = sign(\sum_{z_i \in SV} a_i y_i K(\boldsymbol{x}, \boldsymbol{z_i}) + b)$ for an input vector $\boldsymbol{x}$. The vectors $z_i \in SV$ are called support vectors, which are representative examples. Support vectors and other constants are determined by solving a quadratic programming problem. $K(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function which maps vectors into a higher dimensional space. We use the polynomial kernel: $K(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x} \cdot \boldsymbol{z})^d$. To extend binary classifiers to multi-class classifiers, we use a pair-wise method which utilizes ${}_nC_2$ binary classifiers between all pairs of the classes (Kreßel, 1998 [39]). We use Libsvm (Lin, 2001 [52]) in our all experiments.

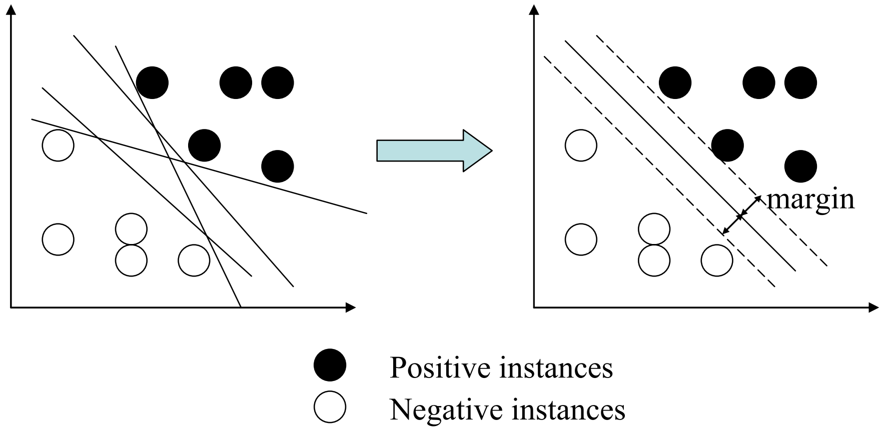


Figure 2-3: Maximize the margin in SVM

2.5 Summary

In this chapter, we describe several preliminary investigations before we will describe our proposed method in next chapter. These investigations are used for designing the annotation criteria. First, we review several related researches, especially the researches that deal with temporal relation analysis based on machine learning.

Second, we introduce the well-known English temporal information annotating guideline - TimeML and we investigate the distribution in the data of Timebank. In this section, we describe an important related research—TimeML— that defines a temporal information annotation guideline. We investigate the distribution of the tags in the temporal relation tagged corpus—Timebank, which is tagged by TimeML standard. We first introduce TimeML standard then describe our investigation in Timebank. Second, we investigate the distribution of temporal information in the timebank. The conclusion of our investigation shows that only considering adjacent event pairs will lose much important information, therefore considering both adjacent viewpoint and syntactic viewpoint enables us to acquire most important temporal relation between events.

Third, we describe a temporal relation identification shared task- TempEval: Temporal Relation Identification shared task. Although the target language of this shared task is different from the one of this thesis, we apply our proposed methods that are used in

Chinese temporal relation analysis for this shared task. The experiment results and the feature design also give us ideas to refine our proposed methods. We first introduce the shared task then describe our system for this shared task and the results of the system. Although our proposed system for this shared task does not achieve best accuracy, the results of all participants are close to each other. The common point of our method and other participants is that many participants used parsing results for this shared task. It shows the importance of syntactic analysis for temporal relation analysis.

Chapter 3 Strategy of Chinese Temporal Relation Annotation

In this chapter, we describe our proposed method- "an automatic temporal relation annotating system based on dependency structure for Chinese". First, we introduce the dependency structure and discuss the advantage of adopting dependency structure for temporal relation analysis. Second, we describe the temporal relation analysis in the viewpoint of dependency structure. Finally, we describe the construction of our temporal relation annotating system.

3.1 Dependency structure and temporal relation

Our proposed method adopts a viewpoint of dependency structure for temporal relation extracting. In this section, we introduce the dependency structure and discuss the advantage of using dependency structure.

3.1.1 Dependency representation and phrase representation of syntactic structure

In natural language processing, there are two major syntactic structure representations. One is a phrase structure representation and the other is a dependency structure representation. In general, a phrase structure representation may be found more suitable for languages with fixed word order patterns and clear constituency structures. Alternatively, using dependency representations can find more suitable for languages which allow flexible word order. In such languages, linearisation is controlled more by pragmatic than by syntactic factors. For example, for analyzing some Slavonic languages and Italian can benefit from a dependency structure representation than phrase structure representation. In Chinese language processing, attempts have been made to use dependency structure representation for Chinese sentence parsing successfully.

The two possibilities are mentioned here, Dependency and simple Phrase Structure grammar models are certainly not the only options available to annotate a corpus. Other approaches, such as Lexical functional grammar (LFG) and complex phrase structure grammar models (e.q. Generalized Phrase Structure Grammar (GPSG) and Head-Driven Phrase Structure Grammar (HPSG)), are also successfully developped. However, the reason why only phrase representation and dependency representation are covered here is that by now these two models have a certain tradition in corpus annotation; and they have been used to annotate corpora both manually and automatically. Though it is true that HPSG parsers and corpora exist, the existing HPSG parsers are not robust enough and of sufficiently wide coverage to serve as a basis for corpus annotation.

Dependency representation is concerned directly with individual words. The dependency grammar is composed by asymmetric head-modifier (governor-dependent) relations between words. In some researches (Hudson, 2000 [35]), arrows indicate the dependency relation between two words point from a head word to its modifier word. However, we define the arrows from modifier to head in our research. This is because one of our dependency analyzers (see section 4.1) is a bottom-up parser. We use this definition to illustrate the head-modifier event pairs in temporal relation annotating. That is, one kind of the temporal relations that we focus on is the temporal relation between the focus event and its ancestor event (see section 5.4).

The representative example of a dependency structure and a phrase structure were shown in Figure 3-1. In the above part of this figure, it shows the phrase structure of the instance sentence: “鄭成功 / 收復 / 臺灣 / 的 / 偉大 / 功業 (*The great triumph that Cheng Cheng-Kung recaptured Taiwan.*)”. This phrase structure and the labels can also be represented in a labeled bracketed structure as follows: [NP[S[S 鄭成功 / 收復 / 臺灣 S] 的 S]偉大 / 功業 NP]. In the below part of Figure 3-1, alternatively, it shows the dependency structure of the sentence. Arrows point from the modifier to the head. The dependency relation between each word pair can be extracted from the dependency structure. The dependency relations of this structure are shown as {(鄭成功, 收復), (臺灣, 收復), (收復, 的), (的, 功業), (偉大, 功業)}. Here, each bracket means (modifier, head).

In exploring the dependency structure of a sentence, there are some basic components of a transformational-generative formalism. The mathematical properties of Dependency Grammar (Tesniere, 1959 [71]) are studied by Gaifman (Gaifman, 1965 [28]) and Hayes (Hayes, 1964 [31]). Following their footsteps, Robinson (Robinson, 1970 [69]) formulates four axioms of dependency structures. Huang (Huang, 1982 [34]) derives a five axiom. These five axioms are for the well-formedness of dependency structures:

Axiom 1: One and only one element is independent.

Axiom 2: All other elements depend directly on some element.

Axiom 3: No elements depend directly on more than one other element.

Axiom 4: If element *A* depends directly on element *B* and other element *C* intervenes between them (in linear order of string), then C depends directly on A or on B or some other intervening element.

Axiom 5: An element cannot have modifiers lying on the other side of its own head.

In Chinese language processing, dependency structure can make perfect sense when one is parsing “surface” strings. These axioms can be concluded as that each word in a general dependency structure only has a head word and the dependency structure is projective. Generally, Chinese sentences accord with these axioms.

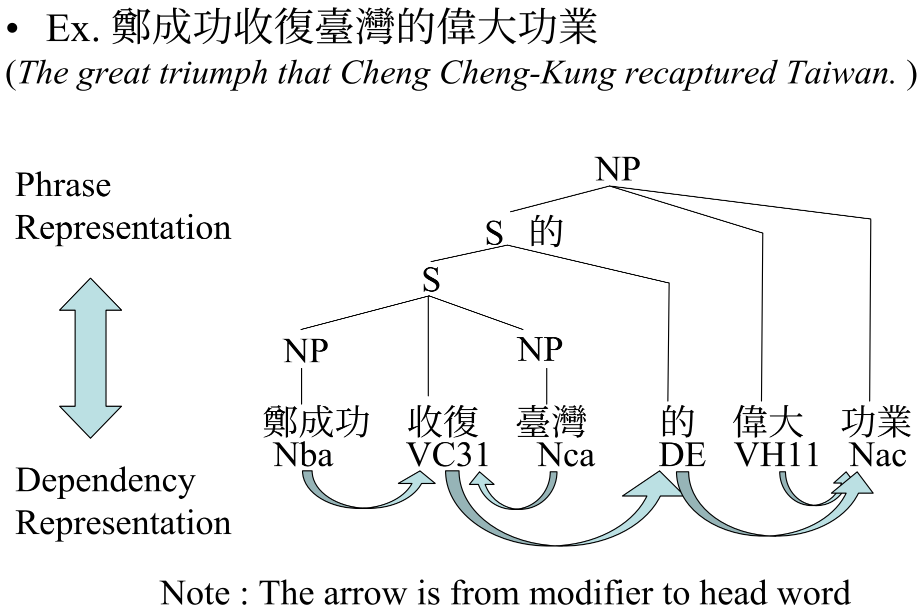


Figure 3-1: The example of dependency and phrase structure

3.1.2 Using dependency structure for temporal relation analysis

We wish that we can identify all possible temporal relations in a news article. It means that the annotator or the machine learning models need to analyze all combinations of events. The table in Figure 3-4 shows all combinations of the events in the diagram and our system try to fill this table. However, to annotate relations of all event pairs is time-consume. For *n* events in an article, C_2^n relations should be considered[14]. For example, if an article contains 50 events, there are 1225 event pairs (C_2^{50}) should be considered.

A simple method for reducing the manual efforts of temporal relation annotating is to consider only adjacent event pairs then to extend these relations by inference rules. For

[14] We assume that the inverted relation pairs, such as “event A occurs before event B” and “event B occurs after event A”, are different, because the combination C_2^n only calculates a single direction of temporal relations, for example, the relation from event A to event B is extracted, but the relation from event B to Event A is not considered. However, our method would extract two directions of temporal relation (the relation that from Event C to Event A is possible). If a relation between a combination event pair is extracted, we extend the inverse relation automatically (if event A occurs before event B, then event B occurs after event A).

example, Figure 3-2 (a1) illustrates an adjacent event sequences, we can only annotate the temporal relation between an event and its preceding adjacent event and we induce a long distance temporal relation using inference rules (Figure 3-2 (a2)). Then we can acquire all temporal relations in the events. However, events in most of articles cannot be considered as an adjacent event sequence. Some events could be hypothetical events (see section 5.3) which do not have understandable temporal relation to its preceding events (Figure 3-2 (b1)), then the hypothetical events "segment" this adjacent event sequence to several fragments. The inference rules cannot be adopted in several event fragments (Figure 3-2 (b2)) for acquiring more long distance temporal relations because the relation between fragments is unknown. If we know that the event does not have a temporal relation to its preceding event previously (the event 3 in Figure 3-2 (b3)), we can annotate a temporal relation between the event and more preceding event to connect the segments. This example remains us of an idea that we need to consider not only adjacent event sequence but also annotate a structure of events (see section 3.2.2).

We apply the dependency structure for temporal relation analysis because the viewpoint of dependency structure can describe the relation between head word and its modifier. As we discuss in section 2.2.2, subordinate event pairs correspond with the head-modifier relations in a dependency structure. We can refer these head-modifier relations to analyze the temporal relation. For example, to decide whether an event is an actual world event or a hypothetical world event (see section 5.3), the head event of the focus event is important clue. We can use dependency structure to describe that temporal expressions affect directly to their head event. In addition, the sibling relation in the dependency structure corresponds to coordinate relation. To apply the dependency structure in temporal relation analysis can therefore deduces the evaluation and the cost for constructing a temporal relation annotating corpus manually. We describe our proposed method that to apply the viewpoint of dependency structure to annotate temporal relations in next section.

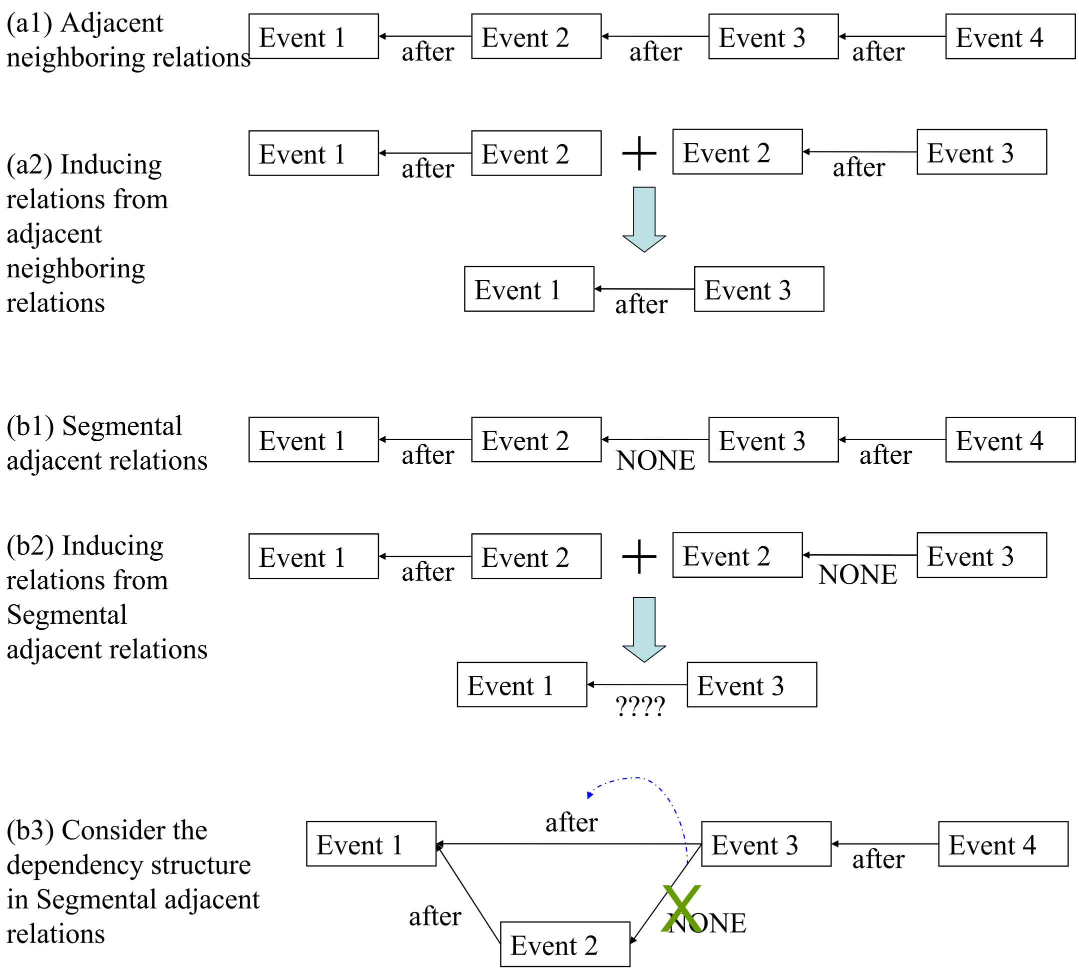


Figure 3-2: Continue / segmental adjacent event pairs and re-link the segmental adjacent events

3.2 The temporal relations of events in dependency structure

We propose a machine learning based temporal relation annotating system for Chinese. Before we construct the system, we annotate a temporal relation annotated corpus that is based on the viewpoint of the dependency structure. In this section we introduce our proposed method for developing a Chinese temporal relation annotating system. Our idea is introduced from the data analysis of TimeBank and our annotating guideline is based on TimeML (see section 2.2.1). TimeBank (annotated according to TimeML guideline) includes all understandable temporal relations between two entities[15] and is annotated manually. However, to annotate full temporal information on a newswire text requires large human effort and cost. To reduce the human effort, we introduce several constraints on the original TimeML. First, we limit the definition of events to verbs. Second, we focus on three types of event pairs in a complete graph according to dependency structure and use inference rules to extend relations.

3.2.1 The definition of the events

First, we limit the definition of events to verbs. According to the description of the TimeML guideline for English, the elements that can be regarded as events include verbal noun, normal nouns (noun phrases), verbs, adjectives, predicative clauses and prepositional phrases can be regarded as events. This definition of events corresponds with the grammatical types of event instances in Chinese. All event types in Chinese sentence are shown as following examples, where the italicized words are regarded as events in the sentences:

- Noun phrases: *第二次世界大戰* (***World War II***)
- Verbal nouns: 他/在/*電話*/中/說…(In ***the telephone call***, he said…)
- Verb: 我/*買了*/一具/電話 (I ***bough***t a telephone)
- Adjectives: 金融/市場/運行/*平穩* (The function of financial market is ***smooth.***)
- Predicative clauses: 政府/協助/工廠/*改進/設施* (The government helped the factory ***that amended their equipments***)
- Prepositional phrase: 他/*在*/*爭奪*/*決賽權*/*時*/落敗 (He was eliminated ***when he contended for passing the preliminary***.)

The noun phrase "第二次世界大戰 (World War II)" is a named entity and could mean both an event and a temporal expression. Many named entities that describe a historical occurrence representing events. We do not focus on the named entities events because the

[15] TimeBank includes the relations between two temporal expressions, two events and a event-temporal expression pair

named entity recognition is still a difficult task and to understand the event meaning of named entity needs world knowledge.

Many normal nouns and noun phrases can describe events, however, to recognize whether nouns and noun phrases represent events or not is difficult in Chinese articles. For example, a word "電話 (telephone)" could mean a telephone machine in the example "我/買了/一具/電話(I bought a telephone)", or could mean a telephone call in the example "他/在/電話/中/說…(In the telephone call, he said…)". The meaning (events or normal nouns) of most nouns are ambiguous in the result of morphological analysis. Therefore, we do not consider the temporal relation of noun events and noun phrase events in our work.

The usages of adjectives have similar property. An adjective can be either an event (describes a statement) or a modifier for a noun depends on the context. However, in Penn Chinese Treebank, the POS-tags of adjective include attributive adjectives (JJ) and predicative adjectives (VA).

In other types of events (predicative clauses and prepositional phrases), to recognize these entities from a context needs the chunking techniques. It is complicated that to recognize these event entities when we extract the events automatically. However, we can focus on the verbs instead of the predicative clauses and prepositional phrases. The phrase / clauses usually have the hierarchical structure of verbs. For example, in the sentence "政府/協助/工廠/改進/設施 (The government helped the factory ***that amended their equipments***)", the string "工廠/改進/設施 (to amend their equipments)" is a verb phrase and can be regarded as an event. Using the dependency structure, we can extract the word dependency relation between the focus verb and its ancestors / descendants words. These dependency relations include the structure of the event phrases and the event clauses. Therefore, we can acquire the event candidate of the predicative clauses and prepositional phrases by considering the verbs and their dependency structure.

It is difficult to recognize events from all event candidates except for verbs. However, following the preceding discussion, we can focus on verbs then acquire most of the events in articles. To simplify the process of recognizing events, we only regard verbs as events.

It should be noted that we do not limit the domain of verbs. In the related research (Li, et al., 2005 [50]), they manually created a dictionary which includes the common verbs in Chinese financial news articles and recognizing the event using the dictionary. However, our original data do not limit the domain of articles. Our system deals with all verbs in corpus for applying our system to the multi-domain articles.

3.2.2 Three types of event pairs

Second property of our research is that we focus on three types of event pairs in the complete graph. The first one is the adjacent event pairs. The second and third types are the head-modifier event pairs and the sibling event pairs in dependency structure tree representation of a sentence. The first type (adjacent event pairs as the discussion in section 2.2.2) and the other two types (head-modifier or sibling event pairs as the in section 2.2.2) are not exclusive. According to our investigation of TimeBank, subordinate event pairs are head-modifier relations and coordinate event pairs are sibling relations. Using dependency structure can extract these relations from sentences therefore we can acquire the most important temporal relations in a sentence.

The three types of pairs are shown in Figure 3-3. The example phrase “停止/撥付/財政債卷/安排/的/資金/並/起訴 (To stop providing funds that were prepared by financial bond, and to prosecute...) has four events: “停止 (to stop)”, “撥付 (to provide)”, “安排 (to prepare)” and “起訴 (to prosecute)”. The temporal relations of all possible event pairs are shown in the row “All possible temporal relations” of the table in Figure 3-3. For example, the temporal relation: {安排,撥付,before}, means that the event “安排(to prepare)” occurs before the event “撥付(to provide)”.

The adjacent pairs of these events are {停止-撥付, 撥付-安排, 安排-起訴} and these relations are shown in the row “Temporal relations of Adjacent event pairs”. However, the relation of the adjacent event pair “安排-起訴” is not useful information for understanding the main story of the article because the event “安排 (to prepare)” is a subordinate event of the event “撥付 (to provide)” and it describes a past event as a supplement of the event “撥付 (to provide).” The temporal relation between events “停止 (to stop)” and “起訴 (to prosecute)” is more useful than the relation between events “安排 (to prepare)” and “起訴 (to prosecute)” because events “停止 (to stop)” and “起訴 (to prosecute)” are coordinate events.

In the example in Figure 3-3, a native annotator can recognize that the temporal relation between “安排 (to prepare)” and “起訴 (to prosecute)” is “before”. However, many event pairs like this example do not have an explicit temporal relation. To analyze this kind of event pairs ("安排 (to prepare)" and "起訴 (to prosecute)"), we should consider not only the adjacent observation of events but also dependency structure of sentences to acquire the correct temporal information. As the discussion in the preceding section, the adjacent chain (adjacent links) will be disconnected if an adjacent event pair does not have understandable relation. The dependency structure can be used to connect the fragments of the adjacent chain.

The row "Temporal relations of Head-modifier event pairs" in the table shows the temporal relations of the head-modifier event pairs. We can determine these head-modifier event pairs as the subordinate relations. For the event "起訴 (to prosecute)", the most important information is the relation between the coordinate events "停止 (to stop)" and "起訴 (to prosecute)". We define the event pairs that share a head event as a sibling event pair and are shown in the row "Temporal relations of Sibling event pairs" of the table. It should be noted that some adjacent event pairs are also head-modifier event pairs or sibling event pairs. The event pairs {撥付-停止，安排-撥付} are both adjacent event pairs and are head-modifier event pairs. Naturally, the event pair should have the same temporal relation in different viewpoint (in an adjacent pair or in a head-modifier pair).

Figure 3-4 illustrates a diagram of the three types of event pairs in adjacent sentences. There are two sentences with twelve events (from the first event e1 to the last event e12) in the figure and the polygons with dashed-lines show the boundary of sentences. The broken-line links show the adjacent event pairs (from Ll-1 to Ll-11). The dotted-line links show the head-modifier event pairs (from Hl-1 to Hl-10) and the curve links show the sibling event pairs (from Sl-1 to Sl-6). The table in Figure 3-4 lists all combinations of events and the annotated temporal relations fill the lattices. Some adjacent event pairs overlap head-modifier event pairs or sibling event pairs. The lattices with more than one link mean the event pairs can be regarded as both adjacent event pair and head-modifier event pair, or both adjacent event pair and sibling event pair[16]. The goal of our research is

[16] According the definition of dependency structure, an event pair is impossible to be both a sibling relation and head-modifier relation.

to fill all lattices in the table. We use inference rules (see next section) to fill the table as possible as we can.

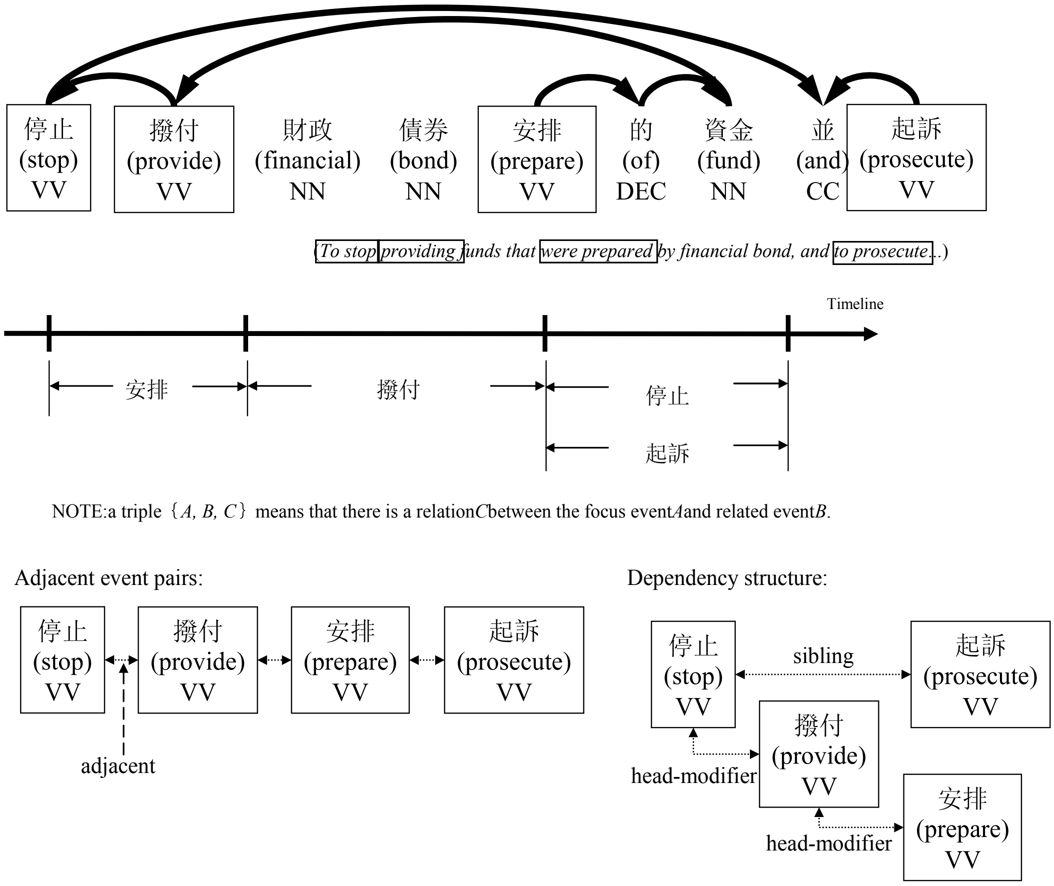
Because a dependency structure describes the structure of a single sentence, most of the three types of temporal relation links are in local structures (in their sentence). For connecting the temporal relations across adjacent sentences, the adjacent event pair links and the sibling event pair links can be used to connect these adjacent sentences. In Figure 3-4, the sibling relation link Sl-4 and the adjacent relation link Ll-7 connect two adjacent sentences. The link Ll-7 connects the last event in "sentence 1" and the first event in "sentence 2". Generally, if an event pair describes temporal related occurrences or statements, the event pair is also an adjacent event pair in an article. Therefore even though an event pair crosses an adjacent sentence pair, we assume that the event pair describes an understandable temporal relation[17].

Another viewpoint for connecting adjacent sentences is to connect the matrix events[18] of the sentences. The link "Sl-4" is the temporal relation between the event "e5" and the event "e11". These events are the matrix events of "sentence 1" and "sentence 2". If we assume that each article has a dummy root and this root is the parent of all matrix events, the relation between the event "e5" and the event "e11" is a sibling event pair. We can use the inference rules on the connecting relations (Sl-4 and Ll-7) to deduce the temporal relations that cross the adjacent sentence or to deduce longer relations.

---

[17] Actually, not all adjacent sentence pairs accordance this assumption, many sentence pairs describe different occurrences. But we consider that this adjacent links provide useful clues for connecting adjacent sentences.

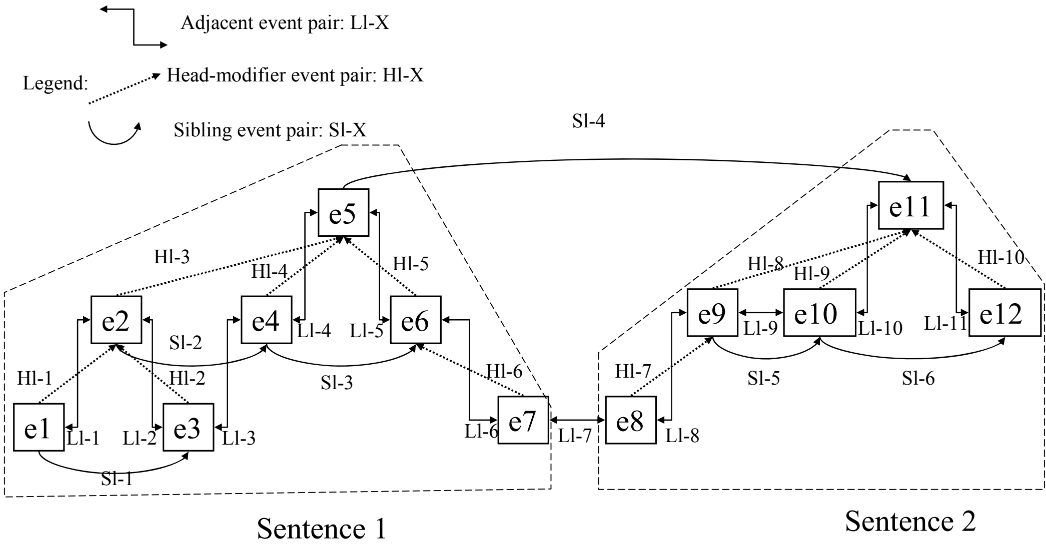
[18] A matrix event is the event of the main verb in the sentence. This definition is defined in TempEval shared task, please see section 2.3.1



Relation Types	Examples
Temporal relations of adjacent event pairs	{安排,撥付,after}, {起訴,安排, after}, {撥付,停止, before}
Temporal relations of head-modifier event pair	{安排,撥付,after}, {撥付,停止, before}
Temporal relations of sibling event pair	{停止,起訴,simultaneous}
Extend event relations using inference rules	{停止,安排,after}, {撥付,起訴,before}
True temporal relations	{安排,撥付,before}, {安排,停止,before}, {安排,起訴,before}, {撥付,停止,before}, {撥付,起訴,before}, {停止,起訴,simultaneous}

Note: The attributes in a bracket are {focus event, comparable event, temporal relation type}

Figure 3-3: The temporal relations in the example phrase



	Modifier event												
		e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12
Head event	e12										Sl-6	Ll-11, Hl-10	
	e11					Sl-4				Hl-8	Ll-10, Hl-9		
	e10									Ll-9, Sl-5			
	e9								Ll-8, Hl-7				
	e8							Ll-7					
	e7						Ll-6, Hl-6						
	e6				Sl-3	Ll-5, Hl-5							
	e5		Hl-3		Ll-4, Hl-4								
	e4		Sl-2	Ll-3									
	e3	Sl-1	Ll-2, Hl-2										
	e2	Ll-1, Hl-1											
	e1												

Figure 3-4: A diagram of the three types of event pairs and connecting the sentences

3.2.3 Use of inference rules

After annotating these relation tags, we use inference rules (See Table 3-1) to extend the temporal relations. We only consider simple logical relation to define these inference rules. For example, if an event A occurs before an event B, and the event B occurs before an event C, then the event A occurs before the event C. The empty lattices are ambiguous relations then we cannot deduce more relations.

The row “Extend event relations use inference rules” in Figure 3-3 shows the temporal relations extended by inference rules. By annotating the three types of temporal relation and using the inference rules to extend the temporal relations, we do not need to annotate all possible event pairs but we can acquire a number of useful temporal relations.

	The relation between event B and event C								
The relation between event A and event B	AFTER	BEFORE	DURING	INCLUDE	SIMULTANEOUS	OVERLAPPED-BY	BEGUN-BY	END-BY	OVERLAP
AFTER	AFTER			AFTER	AFTER	AFTER	AFTER	AFTER	
BEFORE		BEFORE		BEFORE	BEFORE		BEFORE	BEFORE	BEFORE
DURING	AFTER	BEFORE	DURING		DURING				
INCLUDE				INCLUDE	INCLUDE	OVERLAPPED-BY	INCLUDE	INCLUDE	OVERLAP
SIMULTANEOUS	AFTER	BEFORE	DURING	INCLUDE	SIMULTANEOUS	OVERLAPPED-BY	BEGUN-B	END-BY	OVERLAP
OVERLAPPED-BY	AFTER	BEFORE			OVERLAPPED-BY				
BEGUN-BY	AFTER			INCLUDE	BEGUN-BY		OVERLAPPED-BY	INCLUDE	
END-BY		BEFORE		INCLUDE	END-BY		INCLUDE	END-BY	OVERLAP
OVERLAP		BEFORE			OVERLAP		INCLUDE		
	The relation between event A and event C								

Table 3-1: Inference rules

In section 2.2.2, we presented that most of the temporal relations between events in English are the three types that we defined. We expect that these three types of links (Adjacent event pairs, Head-modifier event pair and Sibling event pair) in Chinese are more important than other links. In section 5.4, we describe our temporal relation annotation guideline for Chinese. Section 5.5.2 shows the distribution of tags in our corpus and section 6.1.2 the coverage of the links that is annotated by our proposed method.

In the previous research (Mani, et al., 2006 [55]), the inference rules could adopt some syntactic or semantic features[19] of event pairs to extend more inference rules. For using syntactic / semantic feature, it needs experimental linguistic knowledge to make an induction and we do not collect the linguistic knowledge yet. In our research, therefore, we use the inference rules that only adopt unambiguous relations without syntactic / semantic features.

3.3 Construction of the temporal relation annotating system

In this section, we introduce the construction of our temporal information analyzer. Figure 3-5 illustrates the process of our temporal relation analysis. Our system includes three parts – "morphological analyzer", "preliminary processes" and "temporal relation identification". The first part is a Hidden Markov Model based morphological analyzer (GOH, 2006 [29]), which segments the input text into words and gives POS-tags to each word. Then the second part of our system- "preliminary processes", addes the information to the output token sequence of the morphological analyzer for temporal relation analyzing. The preliminary processes include: "SIGNAL word classifier", "dependency structure analyzer", "simple temporal expression recognizer" and "event detector". The third part is "temporal relation identification". It includes three steps: "Events' Temporal Property Identifier", "Events' Temporal Relations Identifier" and "Long Distance Relations Analyzer". Here, the system deduces long distance relations by inference rules that we describe in section 3.2.3.

The percentage in each part means the accuracy of the part in the system. The 100% accuracy in the block "Morphological Analyzer" is that we use the morphological analyzed corpus. The number that is not 100% means the accuracy of a machine learner. Because our

---

[19] Such as the "POS" tag and the "TENSE" tag are used for creating inference rules in (Mani, et al., 2006 [55]).

proposed method adopts dependency structures, the dependency structure analyzer is a core part of our system. We describe the dependency structure analyzer in Chapter 4. The preliminary process “SIGNAL word classifier” uses a machine learning classifier to detect the SIGNAL word in the article. We describe the SIGNAL word in section 5.2.3 and the implement in section 6.1.3. The preliminary process “simple temporal expression recognizer” uses hand-writen simple rules to detect the numerical temporal expressions. The precision is therefore 100% in our experiments. Whereas the precision is therefore 100%, we cannot estimate the recall since we don’t have manually numerical temporal expression annotated corpus. We describe this process in section 6.1.4. The final preliminary process “event detector” is as the description in section 3.2.1; it selects words with verb POS-tags as the candidates of events. Certainly, the accuracy is 100%.

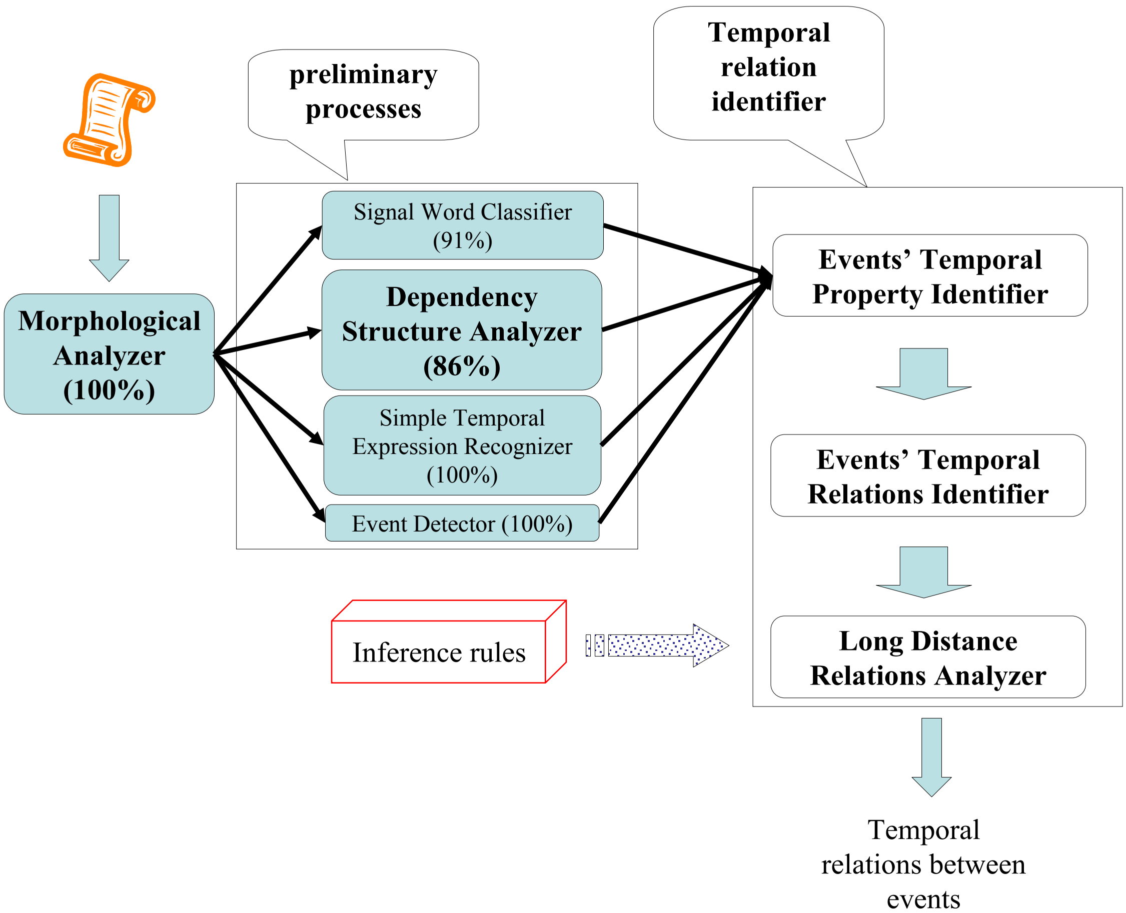


Figure 3-5: The processing flow of our temporal information analyzer

The three steps of temporal relation annotating are our main proposed methods. This process needs an annotated corpus for training a machine learning model. Therefore, the first work in our research is to annotate a temporal relation annotated corpus following our definition manually. This work is described in Chapter 5. After an annotated corpus is constructed, we train a machine learning model instead of the manual annotating to our system. The experiment are shown in Chapter 6

3.4 Summary

In this chapter, we describe our proposed method- “an automatic temporal relation annotating system based on dependency structure for Chinese”. Our proposed method adopts a viewpoint of dependency structure for temporal relation extracting. First, we introduce the dependency structure and discuss the advantage of adopting dependency structure for temporal relation analysis.

Second, we describe the temporal relation analysis in the viewpoint of dependency structure. We propose a machine learning based temporal relation annotating system for Chinese. We introduce our proposed method for developing a Chinese temporal relation annotating system. Our idea is introduced from the data analysis of TimeBank and our annotating guideline is based on TimeML. To reduce human effort, our proposed method includes several constraints on the original TimeML. First, we limit the definition of events to verbs. Second, we focus on three types of event pairs in a complete graph according to dependency structure and use inference rules to extend relations.

Finally, we describe the construction of our temporal relation annotating system. This system includes several preliminary processes - a Hidden Markov Model based morphological analyzer, “SIGNAL” word classifier, dependency structure analyzer, simple temporal expression recognizer and event detector. Then the temporal relation identifoer includes three steps: “identifying the temporal properties of events”, “identifying the temporal relations of events” and “deducing long distance relations”. Finally, the system deduces long distance relations by inference rules in section 3.2.3. The temporal relation annotated corpus is described in Chapter 5 and the performance of our system is experimented in Chapter 6.

Chapter 4 Dependency Analyzer for Chinese

In our temporal relation analysis system, a dependency analyzer is a central unit. The performance of the dependency analyzer will affect the performance of the temporal relation analysis. Until 2004, several well-known phrase / dependency structure parsers for English (Charniak, 2001 [9]; Collins and Roark, 2004 [21]) and Japanese (Kudo and Matsumoto, 2003 [43]) were released. However, there was no published Chinese dependency analyzer that can be used for our research of temporal relation analysis. We should construct an effective dependency analyzer for Chinese. Now, we developed a machine learning based dependency analyzer for Chinese. In this chapter, we describe the construction of the dependency analyzer. First, we introduce two algorithms of dependency analysis and compare the performance of the algorithms for Chinese. Second, we consider the properties of Chinese then propose some methods to improve the performance of the dependency analyzer.

4.1 Algorithms: Nivre's algorithm and MST parsing algorithm

Many syntactic analyzers for English have been implemented and have demonstrated good performance (Charniak, 2001 [9]; Collins, 2004 [21]; Ratnaparkhi, 1999 [69]). However, implementation of Chinese syntactic structure analyzers is still limited, since the structure of the Chinese language is quite different from other languages. We had implemented the dependency analyzer by two algorithms- Nivre's algorithm (Niver, 2004 [62]) and maximum spanning tree algorithm (McDonald, 2006 [59]). In this section, we introduce the algorithms that we implemented in our machine learning-based syntactic structure analyzer. Then we verify the practicability of the algorithms and discuss the excellence / deficiency of the algorithms in Chinese dependency structure analysis.

4.1.1 Introduction of Nivre's algorithm

We utilize a deterministic bottom-up algorithm and a maximum spanning tree algorithm for dependency relation construction. Deterministic methods of dependency structure analysis are proposed for Japanese (Kudo, 2002 [43]), for English (Yamada, 2003 [78]; Nivre, 2004 [62]) and for Norwegian (Nivre, 2007 [63]). In our previous research (Cheng, 2005 [13]), we adopt Yamada's method and Nivre's method to implement Chinese dependency analyzer and compare the performance of these algorithms. We find that Nivre's algorithm is better than Yamada's algorithm in Chinese dependency analysis in our preliminary research (Cheng, 2005 [13]). We adopt the Nivre's algorithm to constructing the bottom-up dependency analyzer for Chinese.

Figure 4-1 describes Nivre's algorithm and Figure 4-2 illustrates the operation in the algorithm. In Nivre's algorithm, the analyzer's configurations are represented by a triple $\langle S, I, A\rangle$. S and I are stacks, S keeps the words being in consideration. I keeps input tokens yet to be analyzed. A is a list of dependency relations that are determined during the parsing process. Given an input token sequence W, the analyzer is initialized by the triple $\langle nil, W, \phi\rangle$. The analyzer estimates the dependency relation between two tokens (the last token t in S and the first token n in I). The algorithm iterates until the list I becomes empty. When the list I becomes empty, the analyzer stops the iteration and outputs the word dependency relation A.

There are four possible operations to the next configuration:

- **Right:** In the current triple $\langle t|S, n|I, A\rangle$ (t is the top element and S is the remaining element in a stack), if there is a dependency relation that the word t depends on word n, the analyzer extends A with $(t \rightarrow n)$, removes t from S, and the configuration now becomes the triple $\langle S, n|I, A \cup \{(t \rightarrow n)\}\rangle$.

- **Left:** In the current triple $\langle t \mid S, n \mid I, A \rangle$, if there is a dependency relation that the word *n* depends on the word *t*, the analyzer extends *A* with $(n \rightarrow t)$, pushes *n* onto the stack *S*, and the configuration now becomes the triple $\langle n \mid t \mid S, I, A \cup \{(n \rightarrow t)\} \rangle$.

In the current triple $\langle t \mid S, n \mid I, A \rangle$, if there is no dependency relation between *n* and *t,* the analyzer checks the following conditions.

- **Reduce**: If there are no more words n' ($n' \in I$) which may depend on *t*, and *t* has a parent on its left side, analyzer removes *t* from the stack *S*, and the configuration now becomes the triple $\langle S, n \mid I, A \rangle$.
- **Shift:** If no above three conditions are satisfied, then push *n* onto the stack *S*, and the configuration now becomes the triple $\langle n \mid t \mid S, I, A \rangle$.

These operations are depicted in Figure 4-2. Given an input sentence of length *N* (words), the analyzer is guaranteed to terminate after at most *2N* actions. The dependency structure given at the termination is well-formed if and only if the subtrees are connected (Nivre, 2004 [62]).

It should be noted that the definition above is slightly different from the original algorithm (Nivre, 2004 [62]). Each word of input sentence becomes a token. The token includes the word, the POS, the information of its children, and other useful information. These become the features for the classifiers which determine the four operations.

<u>Nivre algorithm</u>

Input Sentence: $\{(w_1, p_1), (w_2, p_2), \ldots\ldots (w_n, p_n)\}$;

Initialize:I $= \{(w_1, p_1), (w_2, p_2), \ldots\ldots (w_n, p_n)\}$, S={}, A={};

Start:

While $|I| \geq 1$

do

x = ***get contextual features*** (**I**$_1$***;S***$_{Last}$) ;

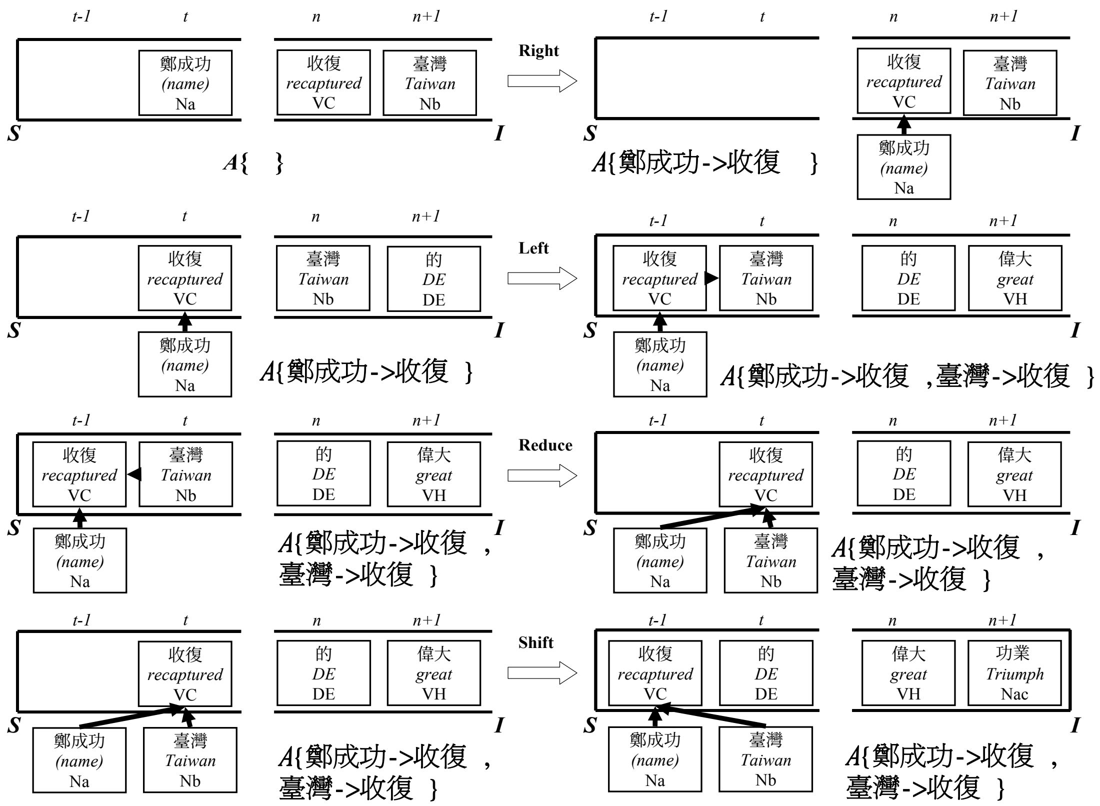
y = ***estimate operation*** (***model;*** **x**) **;**

//The operations are described in Figure 4-2

construct subtree (**I**$_1$***; S***$_{Last}$***; y***) **;**

end;

Figure 4-1: Nivre algorithm



Example: **鄭成功收復臺灣的偉大功業(“The great triumph that Cheng Kheng-Koug recaptured Taiwan.”)**

Figure 4-2: Four operations of Nivre’s algorithm.

4.1.2 Introduction of maximum spanning tree algorithm

The main idea of maximum spanning trees dependency analyzer (McDonald, 2005 [59]) is that the dependency parsing is the search for a maximum spanning tree in a directed graph. Eisner (1996 [26]) adopts maximum spanning tree algorithm for projective dependency parsing ($O(n^2)$). McDonald et al. (2006 [60]) implements the parsing algorithm with the Margin Infused Relaxed Algorithm (MIRA) to analyze the dependency structure of non-project languages such as Czech. We implement maximum spanning tree algorithm with Support Vector Machines (SVMs) to analyze the dependency structure of Chinese sentences.

A directed graph $G=(V,E)$ includes the vertex set $V=\{v_1,\ldots\ldots v_n\}$ and the set $E\subseteq[1:n]\times[1:n]$ of pairs (i, j) of directed edges $v_i \rightarrow v_i$. An edge that directs from v_i to v_j has a score $s(i, j)$. Because graph G is directed, $s(i, j)\neq s(j, i)$. A *maximum spanning tree* (MST) of G is a tree $T \subseteq E$ that maximizes the value $\sum_{(i,j)\in T} s(i,j)$ where the tree T and the graph G have similar vertex set. The maximum projective spanning tree of G is constructed similarly except that it can only contain projective edges relative to some total order on the vertices of G. The MST problem for directed graphs is also known as the maximum arborescence problem.

We define an input sentence that we want to analyze as the directed graph $G=(V,E)$ where:

$V=\{v_0=\text{root},\ v_1,\ldots\ldots v_n\}$

$E=\{(i,j):i\neq j,(i,j)\in[0:n]\times[1:n]\}$

The vertexes in the graph G correspond to the words in the sentence. A dummy node v_0 is added to the vertex set to explain the root node of the dependency tree. It is clear that dependency trees for input sentence correspond to the spanning trees, since both kinds of trees are required to be rooted at the dummy root and reach all the words in the sentence. Hence, finding a (projective) dependency tree with highest score is equivalent to finding a maximum (projective) spanning tree in the complete directed graph G. For searching the MST of the graph G efficiently, McDonald (2005 [62]) adopt the Chu-Liu-Edmonds algorithm (Chu and Liu, 1965 [17]; Edmonds, 1967 [25]) to search the MST of the graph G. We follow McDonald’s method to construct a dependency analyzer. We will describe the MST parsing based on Chu-Liu-Edmonds algorithm by using the following example.

Figure 4-3 describes the MST parsing algorithm and Figure 4-4 illustrates an example that analyzes a Chinese sentence. To find the highest scoring non-projective tree for a input token squence in Figure 4-4, “鄭成功 / 收復 / 臺灣 (*Cheng Cheng-Kung recaptured Taiwan.*)”. We simply construct the graph G and run it through the Chu-Liu-Edmonds algorithm. The resulting spanning tree is the best non-projective dependency tree.

In step 1 (Figure 4-4: step 1), we construct the graph G and estimate the score of each edges. The edges in the graph regard to that the initial vertex depend on the terminal vertex. After the MST is found, each vertex other than the root node should have at most one out-going edge. We estimate the score of edges by using support vector machines. We discuss this in next section.

In step 2 (Figure 4-4: step 2), for each vertex, the analyzer selects the out-going edge (except the edges that terminate at the root node) which has maximum score. For example, the selected edge of the vertex “臺灣(*Taiwan*)” is the edge (“臺灣”, ”收復”) because its score is maximum than other out-going edges. After the analyzer selects the maximum score edges of vertexes, we get a subgraph G_M . If there is no circle in the graph G_M , the graph G_M is MST and the analyzer output the dependency structure. Otherwise, the analyzer goes to the step 3 to resolve the circle in the graph. In this example, the vertexes (收復, 鄭成功) constitute a circle. The analyzer memorizes these vertexes in a circle to execute the next step.

Step 3 (Figure 4-4: step 3a and step 3b) includes three processes. First, for each vertex in the circle (收復, 鄭成功), the analyzer searches an outside vertex (root, 臺灣) with maximum path score. The outside vertex (with maximum path score) lets the path that starts from the circle vertex to the outside vertex has maximum score. For example, the outside vertex of the circle vertex “收復” is the vertex “root” and the vertex “臺灣”. The analyzer estimates the score of the path (收復, 鄭成功, 臺灣) and the path (收復, 鄭成功, root). The score is *s*(收復, 鄭成功, 臺灣)=10+8=18 and *s*(收復, 鄭成功, root)=10+5=15 (refer to Figure 4-4 step 1). The outside vertex with maximum path score of the circle vertex “收復” is the vertex “臺灣”. Similarly, the outside vertex of the circle vertex “鄭成功” is the vertex “root”.

Next, the analyzer selects an outside vertex with maximum path score for this circle. In this example, the maximum score path is (鄭成功, 收復, root). Therefore, we find an appropriate path to correct the circle. We shave the edge (收復, 鄭成功) and add a edge (收復, root) to the graph. That is, the correct parent of the word “收復” is the root node. Finally, we get a MST in Figure 4-4: End. This is the dependency structure of the input sentence “鄭成功 / 收復 / 臺灣 (*Cheng Cheng-Kung recaptured Taiwan.*)”. The directed edges illustrate the “modifier->head” relations and the vertex that out-going to the vertex “root” is the root node of the sentence.

MST parsing based on Chu-Liu-Edmonds algorithm

Input Sentence as the vertex set: $V=\{v_0,v_1,v_2,......v_n\}$, v_0 is a dummy root of the sentence.

Search Maximum Spanning tree from a directed complete graph *G = (V,E),* where *E* is set of edges of all vertex pairs:

Step 1:

$\forall v_i \in V,\ v_j \in V$

Estimate the score *s(i, j)* of all edges $(v_i,v_j) \in E$

Step 2:

Find a edge set $M=\{(v^*,v) : v \in V,\ v \neq v_0\ v^*=\text{argmax}_{v'} s(v',v)\}$

Step 3:

while ***circle***$(G_M=(V,M))$==true

C_{circle} = ***circle*** $(G_M=(V,M))$, where C_{circle} is the vertex set.

contract$(G_M=(V,M), C_{circle})$

Return the graph (a dependency structure) $G_M=(V,M)$

circle$(G_M=(V,M))$

$\forall v_i \in V$

If the path $v_i\ ,v_j\ ,v_k\ ......v_n$ is a circle,

Return the vertex set $C_{circle} = (v_i\ ,v_j\ ,v_k\ ......v_n\ )$

contract$(G_M=(V,M), C_{circle})$

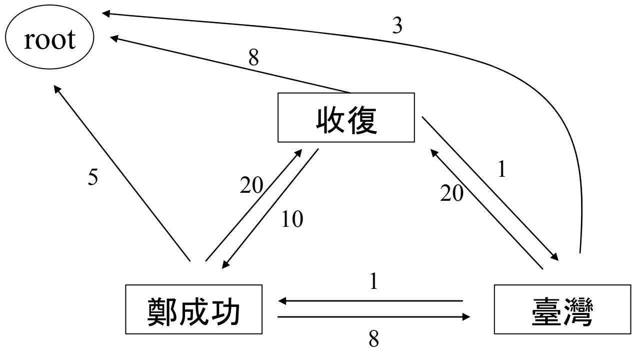
$\forall v_i \in C_{circle}$

Find a maximum score path $P_{out} =(v_i\ ,v_j\ ,......v_n\ ,v_{m\text{-}out})$ from v_i to the node $v_{m\text{-}out}$. Where $v_{m\text{-}out} \in V - C_{circle}$ and $v_{m\text{-}out}$ is a first outside node of the circle.

Shave a edge $(v_n\ , v_i\ ) \in M$

Return G_M

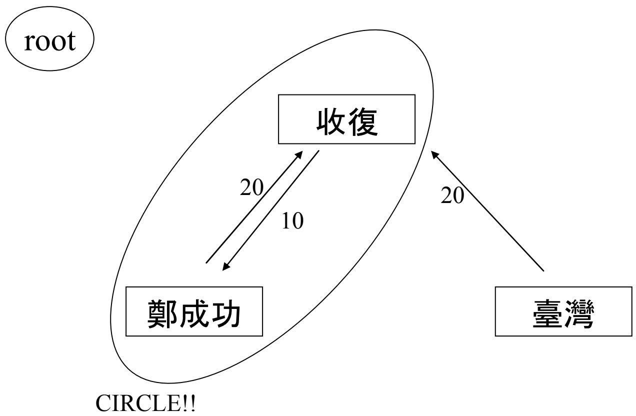
Figure 4-3: MST parsing based on Chu-Liu-Edmonds algorithm



Step 1:

■Let the nodes (words) of the sentence as a vertex set of a complete directed graph

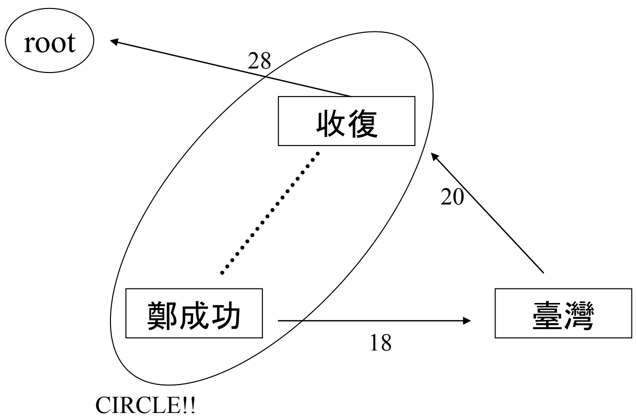
■Estimate the score of each edge of the complete directed graph.



Step 2:

■Select the maximum output edge of every vertex in the graph. (Exclude the edges to the root node)

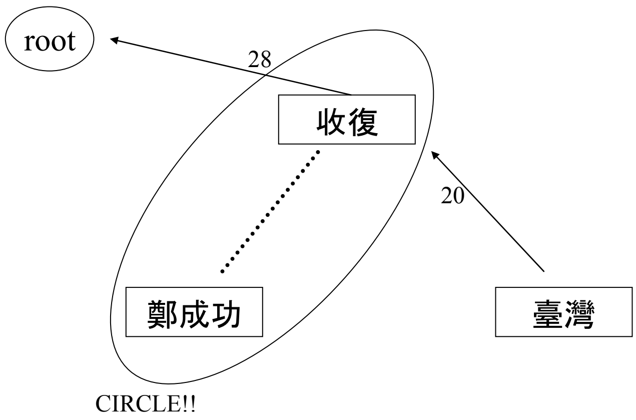
■Search the circles in the graph (the vertex(**收復**, **鄭成功**) is a circle)



Step 3a:

■Estimate the score of the path that start from a vertex in the circle to the first outside node.

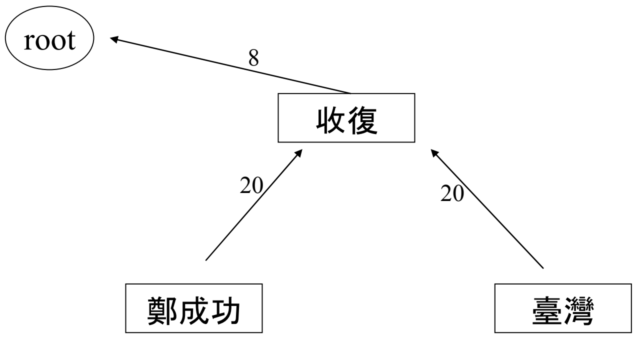
■The first outside node of the vertex ”**鄭成功**“ is the root node; the first outside node of the vertex ”**收復**” is the node “**臺灣**”.



Step 3b:

■Select the out-going edge that has maximum score. (select the edge (**鄭成功**, **收復**))

■Shave the repugnant edge. (shave the edge (**收復**, **鄭成功**))



End:

■Return the dependency structure.

Figure 4-4: An example of using MST parsing algorithm to analysis the dependency structure of Chinese.

4.2 Implement a multi-lingual dependency analyzer

We adopt both Nivre's algorithm and MST algorithm to construct a Chinese dependency parser. Before we implement the dependency analyzer, we are interesting in the performance of using these algorithms in multi-lingual dependency analysis. To compare the performance of these parsing algorithms in multi-lingual dependency analysis helps us to decide an appropriate approach for Chinese dependency analyzer. In this section, we first describe the system construction of dependency analyzer with these two parsing algorithms. Then we experiment the parsers in the multi-lingual dependency analysis shared task at the Conference on Computational Natural Language Learning (CoNLL) 2006 and 2007.

4.2.1 Implement the dependency analyzer

Nivre algorithm:

In Nivre's algorithm, the analyzer decides an optimum operation for a token pair in parsing process. We regard this task as a supervised classification task. A supervised classification task needs training and testing data which consist of annotated data instances. Each instance in the training set contains one "target value" (class label) and several "attributes" (features). The goal of a classifier is to produce a model which predicts target value of data instances in the testing set which only give the attributes.

We select the support vector machines (see section 2.4) as the classifier. We have tried other machine learning methods, such as memory based learning or maximum entropy method, to construct the dependency analyzer (Cheng, 2005 [13]). The performance of SVMs is better than using others in Chinese dependency analysis. This is because that SVMs can adopt combining features automatically (using the polynomial kernel), whereas other methods should add combining features manually. To extend binary classifiers to multi-class classifiers, we use the pair-wise method, which utilizes ${}_nC_2$ binary classifiers between all pairs of the classes (Kreel, 1998 [39]). Therefore SVMs classifier outputs the optimum operation.

Figure 4-5 illustrates the features for deciding the optimum operation in the Nivre's algorithm with SVMs. In our method, the analyzer considers the dependency of two nodes (n,t) which are in current triple. The nodes include the word, the POS-tag and the information of its children (the word and the POS-tag of the children). The context features we use are 2 preceding nodes of node t (and t itself), 2 succeeding nodes of node n (and n itself), and their children nodes. The distance between nodes n and t is also used as a feature.

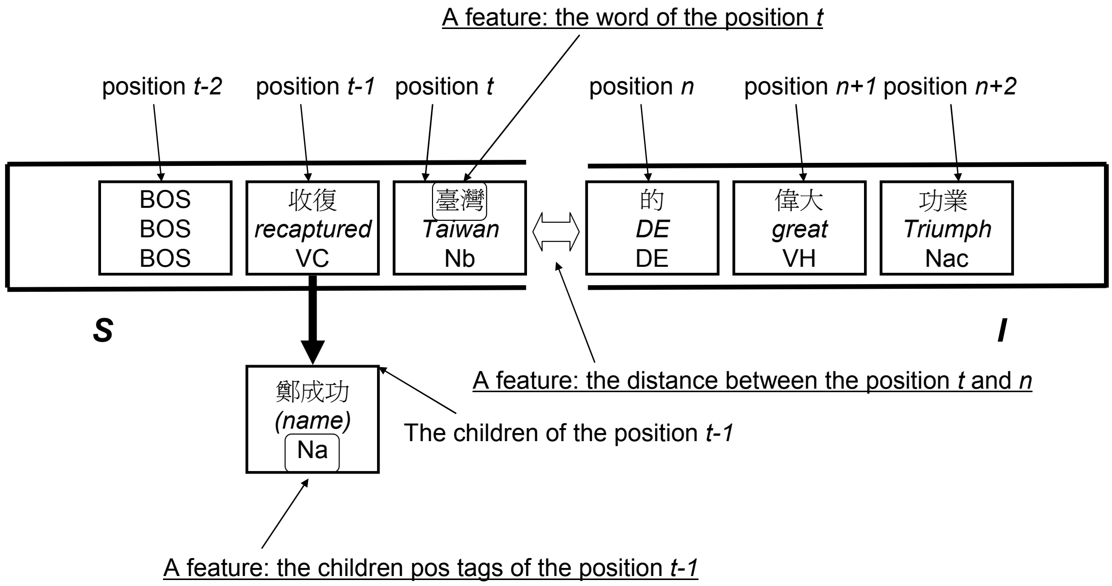


Figure 4-5: The features using for SVMs in the Nivre algorithm

MST parsing algorithm:

The concept of the MST parsing algorithm is different form Nivre's algorithm. Nivre's algorithm needs a classifier for deciding optimum operation. The MST parsing algorithm needs a calculator for estimating the score of all edges. In McDonald's research (2006 [58]), they adopt Margin Infused Relaxed Algorithm (MIRA)(Crammer and Singer, 2003 [23]) in their parser. For estimating the score more accurate, we adopt SVMs in our MST dependency analyzer. Original SVMs classifier is a binary classifier. The output of SVMs is the distance for an input test vector. The analyzer needs SVMs to estimate the score of edges. Therefore, we use a simple sigmoid function ($f(\mu)=\frac{1}{1+e^{-\mu}}$) that translates the decision value to probability (the value is 0~1). In Figure 4-4: step 1, the SVMs estimates the probability that a focus word depends on another word. For example, to estimate the score of the edge (收復, 鄭成功) means to estimate the probability that the word "收復" depends on the word "鄭成功". In our training data for SVMs learner, there exist dependency relations (ex: (鄭成功, 收復) and（臺灣，收復) is a positive instance and other word pairs (ex: (鄭成功，臺灣), (收復，臺灣), (臺灣，root)…) are all negative instances for machine learning.

Figure 4-6 illustrates the features that are used for SVMs in MST parsing. Because the MST parsing algorithm cannot refer the partial dependency tree dynamically (that is, before the MST graph is found, the dependency relations between all word pairs are uncertain), we only use the information of each word (the word and the POS-tag). To estimate the probability of the word *t* depends on the word *n*, we select several windows to extract the features. The features are: the focus word *t* and *n* ("鄭成功" and "偉大"); the preceding 2 words of *n* (nil word "BOS") / *t* ("臺灣" and "的") and the succeeding 2 words of *n* ("收復" and "臺灣") / *t* ("功業" and "EOS"); the words between the focus words *n* and *t* ("收復", "臺灣" and "的"). The nil words "BOS" and "EOS" mean that the positions are outside of the sentence.

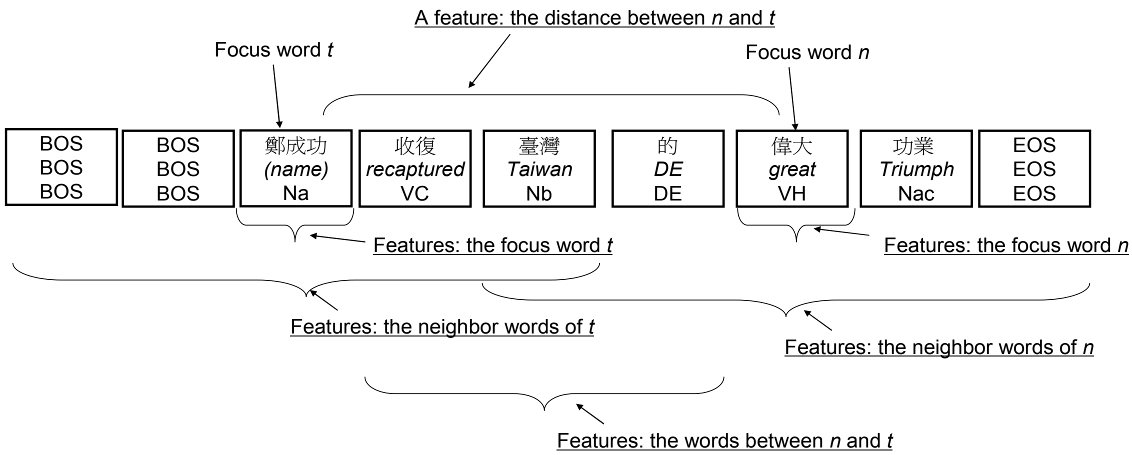


Figure 4-6: The features using for SVMs in the MST parsing algorithm

4.2.2 Experiment in the multi-lingual dependency analysis shared task

We are interested in the performance that using these methods in multi-lingual dependency analysis. If a dependency analyzer has better performance in multi-lingual dependency analysis, we can trust that to adopt it in our Chinese dependency analyzer will achieve better performance. Recently, the multi-lingual dependency analysis task is concerned. In Conference on Computational Natural Language Learning (CoNLL) 2006 (Buchholz and Marsi, 2006 [7]) and 2007 (Nivre et al., 2007 [63]), the shared task deals with the multi-lingual dependency analysis. We use the data of CoNLL-2006 and CoNLL-2007 to evaluate the performance of our dependency analyzer.

The shared tasks of CoNLL-2006 and CoNLL-2007 aim to define and extend the current state of the art in dependency parsing. Ideally, a parser should be trainable for any language, possibly by adjusting a small number of hyper parameters. The shared task provided the community with a benchmark for evaluating their parsers across different languages.

The shared task is to assign **labeled** dependency structures for a range of languages by means of a fully automatic dependency parser. Some gold standard dependency structures against which systems are scored will be **non-projective**. A system that produces only projective structures will nevertheless be scored against the partially non-projective gold standard. The input consists of (minimally) tokenized and part-of-speech tagged sentences. Each sentence is represented as a sequence of tokens plus additional features such as lemma, part-of-speech, or morphological properties. For each token, the parser must output its head and the corresponding dependency relation. Although data and settings may vary among language, the same parser should handle all languages. The parser must therefore be able to learn from training data and to handle multiple languages. The languages that used in shared task are:

CoNLL-X (2006): Arabic, Chinese, Czech, Danish, Dutch, German, Japanese, Portuguese, Slovene, Spanish, Swedish, Turkish and Bulgarian.

CoNLL-2007: Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian and Turkish.

We adopt the two dependency analyzer introduced in section 4.1 in the shared task. All these experiments are implemented on a Linux machine with XEON 3.0GHz dual CPUs and 16.0GB memory. The results of our experiments are shown in Table 4-1. The official evaluation metric in both shared task was the labeled attachment score (LAS), i.e., the percentage of tokens for which a system has predicted the correct HEAD (the head word) and DEPREL (the semantic role of the word), but results we also reported for unlabeled attachment score (UAS), i.e., the percentage of tokens with correct HEAD, and the label accuracy (LA), i.e., the percentage of tokens with correct DEPREL. Because we focus on the performance of the word dependency analysis, we only use the unlabeled attachment score (UAS) to evaluate the performance of our system. We compare our two dependency analyzer that based on Niver's algorithm and MST algorithm from the best analyzer in the participants of each shared task[20].

The results show that in most languages, our MST analyzer (MST parsing algorithm with SVMs) has better performance that our Nivre analyzer. One reason is that the Nivre's algorithm cannot deal with the non-projective dependency relation directly. However, our MST analyzer is not restricted to analyze the projective dependency relations. Another reason is that an error of deciding the operation in Nivre's algorithm infects other operation decision. Because our Nivre analyzer uses the information of the partial tree as machine learning features, an error operation will cause the partial tree incorrect. Therefore, the operation decisions that occur after the error will use the incorrect partial tree as features. This error will multiply more decision error latter.

We show that the performance of our MST analyzer is better than the performance of our Nivre analyzer. However, the analyzing speed of our MST analyzer is slow and it consumes large memory[21]. Because SVMs should estimate all edges of a directed graph, it is an $O(n^2)$ algorithm. The Nivre analyzer is $O(n)$ and consumes less memory than MST analyzer. Therefore, considering the practicality of our temporal relation analyzer, we

[20] The best analyzer in CoNLL-X (2006) shared task is McDonald's analyzer (McDonald et al., 2006 [58]) and the best analyzer in CoNLL-2007 shared task is Nakagawa's analyzer (Nakagawa, 2007 [61]).

[21] For example, our MST parser needs 2Gb memory for analyzing Chinese data in CoNLL-2007 and the analysis time is 4 seconds / sentence. However, our Nivre analyzer only needs 300 mb memory and the analysis time is 0.1 second / sentence in the same data.

cannot forsake the Nivre's algorithm. In next section, we consider the problems of our Nivre analyzer and propose two implements for Chinese dependency analysis. If our Nivre analyzer can be improved by these improvements, our temporal relation analyzer can require less memory for analyzing the word dependency relation.

Language in 2006	Our Nivre analyzer	Our MST analyzer	Best 2006 (McDonald et al., 2006)
Arabic	77.74	79.10	79.34
Chinese	89.46	91.17	91.07
Czech	83.40	87.42	87.30
Danish	88.64	89.36	90.58
Dutch	75.49	83.61	83.57
German	87.66	89.62	90.38
Japanese	93.12	92.56	92.84
Portuguese	90.30	91.08	91.36
Slovene	81.14	84.43	83.17
Spanish	85.15	86.72	86.05
Swedish	88.57	87.77	88.93
Turkish	74.49	76.04	74.67
Bulgarian	91.30	92.04	92.04
Average	84.60	86.57	86.61

Language in 2007	Our Nivre analyzer	Our MST analyzer	Best 2007 (Nakagawa, 2007)
Arabic	82.53	86.38	86.09
Basque	79.33	81.63	81.04
Catalan	91.39	93.10	92.86
Chinese	85.89	88.57	88.88
Czech	82.22	85.88	86.28
English	86.89	89.57	90.13
Greek	80.31	83.60	84.08
Hungarian	78.19	83.91	82.49
Italian	85.60	87.15	87.91
Turkish	82.83	86.00	85.77
Average	83.52	86.58	86.55

Table 4-1: The unlabeled attachment score (UAS) of the multi-lingual experiments

4.3 Improvement of the Nivre algorithm based analyzer for Chinese

In this section, we present a method for improving dependency analyzer of Chinese that based on Nivre's algorithm. We find that there are two problems in our Nivre analyzer and propose two methods to address them. One problem is that some operations cannot be addressed only using local feature. We utilize the global features to address this. The other problem is that this bottom-up analyzer doesn't use top-down information. We supply the top-down information by constructing SVMs based root node finder to address this problem. Experimental evaluation on the Penn Chinese Treebank Corpus shows that the proposed extensions improve the parsing accuracy significantly.

4.3.1 Improvement for the Nivre analyzer

Improvement (a): Using global features and two-step process

The bottom-up dependency analyzer based on Nivre's algorithm has two problems. First, some operations in the algorithm needs long distance information. However, the long distance information cannot be available if we assume a context of a fixed size in all operations. In the algorithm, the operation **Reduce** needs the condition that the node *n* should have no child in *I*. However, it is difficult to check this condition. In a long sentence, the modifier of the focused node *n* may be far away from *n*. Moreover, some non-local dependency relations may cause this kind of error.

In this deterministic bottom-up dependency analysis, we can generally consider the process as two tasks:

- **Task 1**: Does the focused word depend on a neighbor node?
- **Task 2**: Does the focused word may have a child in the remaining token sequence?

In the Task 1, the problem can be resolved by using the information of the neighbor nodes. This information is possibly the same as the features that we described in Figure 4-5. However, these features may not be able to resolve the problem in task 2. For resolving the problem in task 2, we need the information of long distance dependency. In Figure 4-7, the analyzer is considering the relation between focused words "告訴 (*tell*)" and "他 (*he*)".

The features used in this original analysis are the information of words “請 (*please*)”, “告訴 (*tell*)”, “他 (*he*)”, “何時 (*what time*)” and “準備 (*prepare*)”. These features are “local features”. The correct answer in this situation is the operation “**Shift**”. It is because the word “告訴 (*tell*)” has a child “出發 (*start*)” which is not yet analyzed and the focused words don’t depend on each other. However, the local features do not include the information of word “出發 (*start*)”. Therefore, the analyzer possibly estimates the answer as the operation “**Reduce**”. To resolve this problem, we should refer some information of long distance dependency in machine learning. The information about long distance relations is defined as “global features”. We select the words which remain in stack *I* but are not considered in local features as global features.

We cannot use the global features immediately because the global features are not effective in all operations. Therefore, we propose a two-step process in our Nivre analyzer. First, the analyzer uses only the local features (as illustrated in Figure 4-5) to decide the optimum operation. If the result is “**Reduce**” or “**Shift**”, the analyzer leaves the decision to another machine learner that makes use of global features. Then the analyzer outputs the final answer of this analysis process.

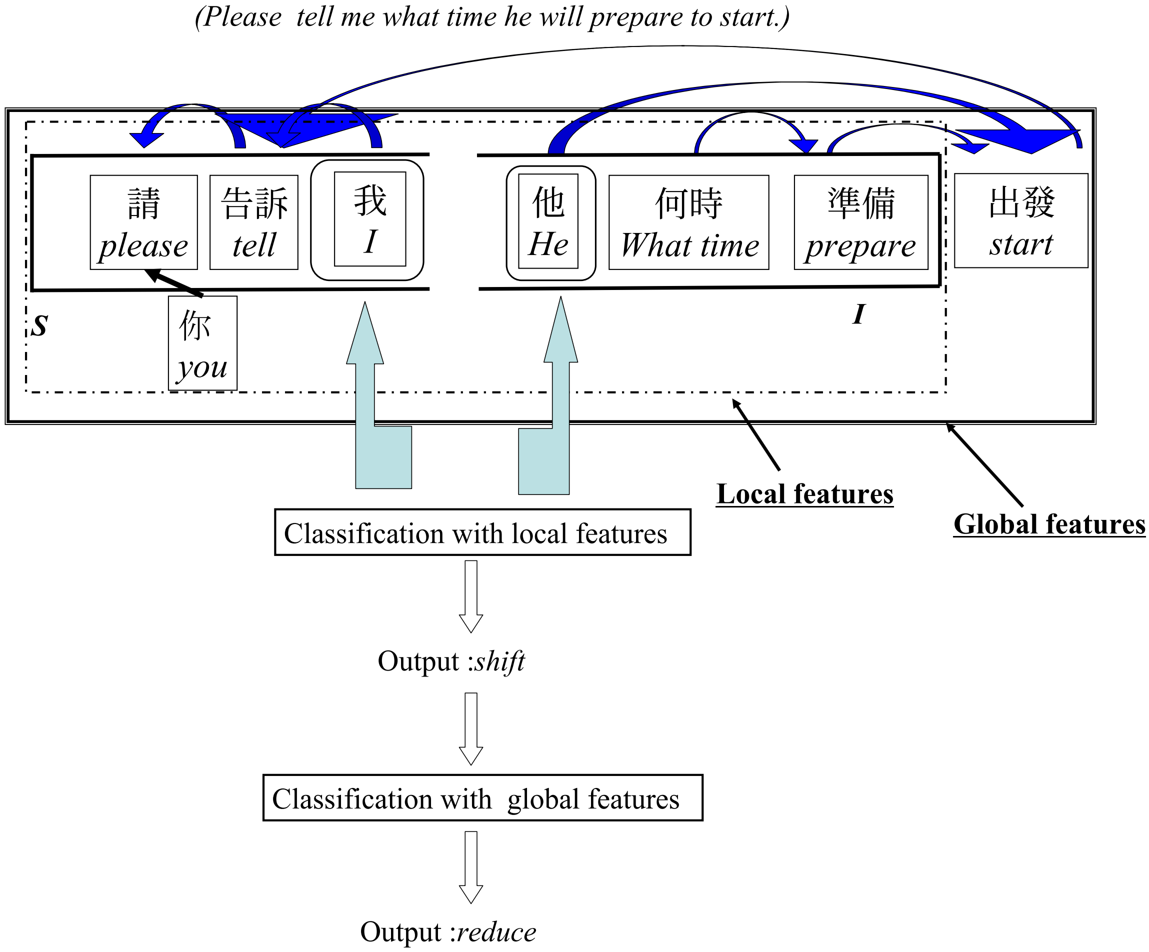


Figure 4-7: An example of the ambiguity of deciding the long distance dependency relation and using two-steps classification dependency relation

Figure 4-7 illustrates an example of using two-step classification for analyzing dependency relation. In this example, the local features are surrounded by dotted line and the global features are surrounded by solid line. The analyzer used local features to analyze the operation of this situation. The result is the operation "***shift***". The analyzer then selected the global features to analyze again and the output is the operation "***reduce***". The final result of this situation is the operation "***reduce***"**.**

Improvement (b): Using the root node finder and dividing the sentence

In the fundamental definition of dependency structure, there is one and only one head word in a dependency structure. An element cannot have dependents lying on the other side of its own governor (projective dependency tree). There are many languages that do not observe these constraints. However, we observed these constraints in the well-known Chinese treebanks (the Penn Chinese Treebank and the CKIP treebank) observe these constraints. We propose the improvement that is based on these constraints.

The second problem of our Nivre analyzer is that the top-down information isn't used in the bottom-up approach. We construct a SVM-based root node finder in our system to supplement the top-down information. In Isozaki's work for English (Isozaki et. al, 2004 [36]), they adopted a root finder in their system to find the root word of the input sentence. Their method used the information of the root word as a new feature for machine learning. Their experiments showed that information of root word was a beneficial feature. However, we think the information of root word can be used not only as the feature of machine learning, but also can be used to divide the sentence. Therefore, the complexity of the sentence can be alleviated by dividing the input sentence.

These peculiarities imply that the head word divides the phrase into two independent parts and each part does not cross the head word. In Figure 4-8, the original input sentence has a root word (the head word of phrase) "與 (*and*)". We can divide this sentence into two sub-sentence "出國 (*exodus*) / 去 (*do*) / 進修 (*study*) / 與 (*and*)" and "與 (*and*) / 到 (*go*) / 國外 (*foreign country*) / 去(*do*) / 旅行 (*visit*)". Both these sub-sentences share the root word "與 (and)". We can conceive that to analyze the dependency structure of the full sentence is to analyze the dependency structure of two sub-sentences. Our Nivre analyzer is a bottom-up deterministic analyzer. Instinctively, the accuracy of analyzing short sentence is significantly better than analyzing long sentence. Thus the performance of the Nivre analyzer can be improved by this method.

To use the root node, we should construct the root finder. Similarly to Isozaki's work, we use machine learner (SVMs) to construct the root finder. We refer to the features which are used in Isozaki's work and investigate other effective features. The performance of our root node finder is 90.71%. This is better than the root accuracy of our analyzers (see Table 4-2).

The tags and features of the root finding are shown in Figure 4-10. We extract all root words in the training data and tagging every word to show that it is root word or not. The features for machine learning of root finder include the contextual features (the information about the focused word, the two preceding words, and two succeeding words) and the word relation features (the words which are in the outside of the window). Other effectual features include the Boolean features “a root word has been found in the SVM-based tagging” and “the focus word is the first/last word of sentence”.

When we use the root finder to analyze the root word of the sentence, we do not know the structure of input sentence (either the phrase structure or the dependency structure). It may look odd that the root finder can analyzes the root word without any information of the structure. However, this analysis is practicable. Naturally, the root word of a sentence is usually a verb (about 61% of sentences have a verb as the root word in our testing corpus). For example, in the example 1 of Figure 4-9, “我 / 去 / 學校 (*I go to school*)”, we know the POS-tags are “noun, verb, noun” thus we can find that the root word is “去 (go)”. However, many sentences include more than one verb or the root word is not verb (in NP or PP…etc.). We can not only choose the verbs as root word directly. To decide the root word of complex sentences, there are some special word/POS relations that can be used to estimate the root node of a sentence. Considering the example 2 in Figure 4-9, the sentence has a verb “收復 (*recapture*)”, but the special word “的 (*DE*)” is in the right side of the verb “收復 (*recapture*)”. The special word “的 (*DE*)” resembles a preposition and it is always the last word of DE-phrase. Therefore, the verb “收復 (*recapture*)” is possibly in the 的 (*DE*)-phrase and the verb cannot be the root word.

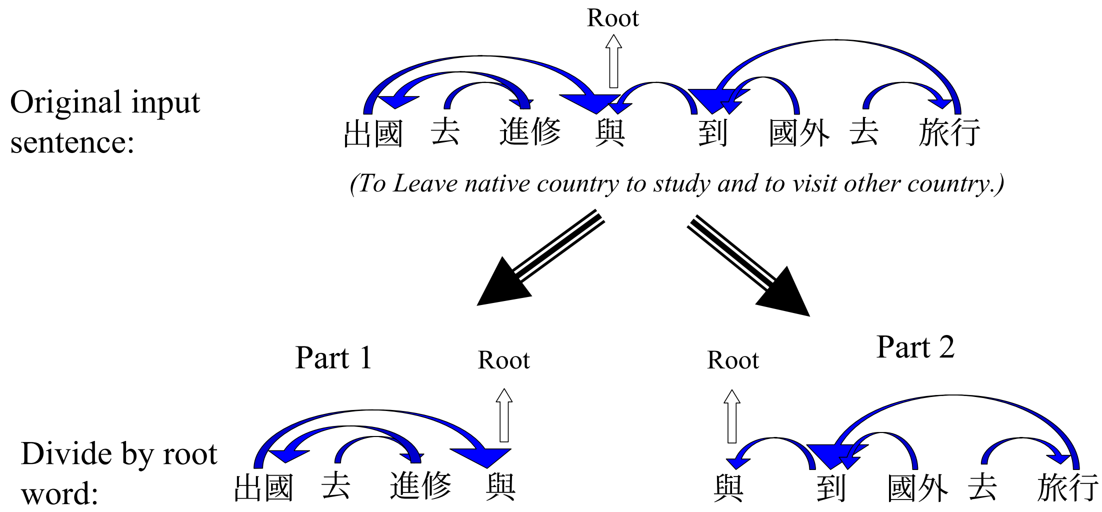


Figure 4-8: Dividing the phrase into two phrases by the root word

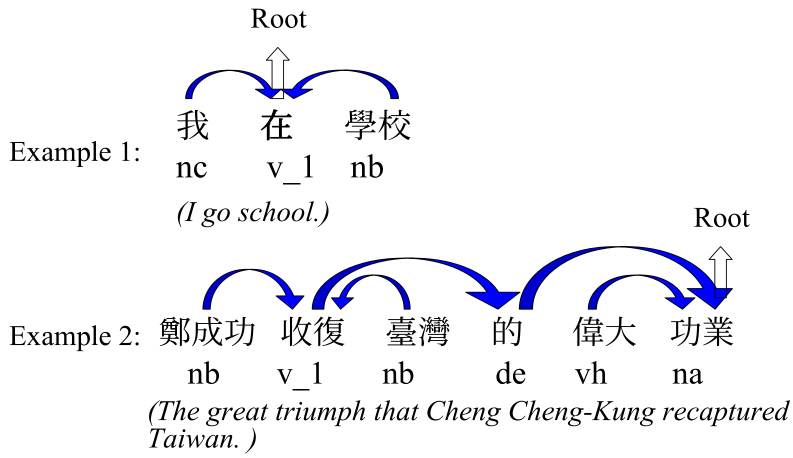


Figure 4-9: The examples of analyzing the root words of sentences

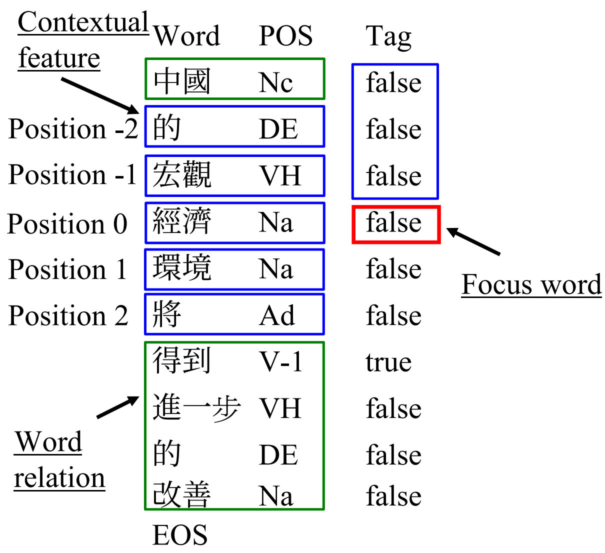


Figure 4-10: The features and tag set of root finder

4.3.2 Experiment of the improvement

The Chinese data in CoNLL shared task is CKIP Treebank (Chen et al., 1999 [10]). The average length of the sentence in CKIP Treebank is 5.9 words / sentence and the sentences are usually the fragments of documents. That is, not all texts in CKIP Treebank are complete. Our temporal relation analyzer deals with the news articles. Therefore, we should experiment the Chinese dependency analyzer in a Treebank that has complete documents and the sentences are longer. We use Penn Chinese Treebank 5.0 (Xue et al., 2002 [76]) in our dependency analysis experiments. This Treebank is represented by phrase structure and doesn't include the head information of each phrase. We convert the Treebank by using the head rules (Cheng, 2005 [14]). The training corpus includes about 377,408 words for learning and 63,886 words for testing.

In the Chinese dependency analysis experiments, we use the measures that are different from CoNLL shared task. Because we will use the dependency structure of the sentence to extract the temporal relations, we focus on not only the accuracy in the word dependency relations but also the accuracy in the complete sentences. The performance of our dependency structure analyzer for Chinese is evaluated by the following three measures:

- Dependency Accuracy: $(Dep.\ Acc.) = \frac{\#\ of\ correctly\ analyzed\ dependency\ relations}{\#\ of\ dependency\ relations}$

- Root Accuracy: $(Root.\ Acc.) = \frac{\#\ of\ correctly\ analyzed\ root\ nodes}{\#\ of\ clauses}$

- Sentence Accuracy: $(Sent.\ Acc.) = \frac{\#\ of\ fully\ correctly\ analyzed\ clause}{\#\ of\ clauses}$

Our experimental results are shown in Table 4-2. First row in the table is the result of our Nivre analyzer (Nivre algorithm with SVMs), second and third row show the effects of the proposed extensions. The fourth row is the result of combining the two extensions. Because the increasing of accuracy that using these proposed extension looks like non-significant, we use McNemar test to confirm the significance of the methods. The McNemar test proves that using the proposed methods improve our Nivre analyzer significantly.

The fifth row in Table 4-2 is the accuracy of using our MST analyzer. The final row is the accuracy of using the root node finder in MST analyzer. We compare our proposed extensions in both our Nivre analyzer and our MST analyzer. However, the proposed extension that using global features and two-steps analysis cannot be adopt in our MST analyzer. The reason is that our MST analyzer estimates the score of edges; it cannot combine with two-steps analysis. And, evidently, our MST analyzer uses the global features of the focus word pair. We do not adopt the two-steps analysis in our MST analyzer. We only combine the root node finder into our MST analyzer. Because we use the root node finder to dividing a sentence, both of the analyzers analyze the two sub-sentences that are divided by using root node.

The second row of Table 4-2 shows that dividing the process of classification as two steps can improve our Nivre analyzer slightly. The reason is that the sentences in corpus do not include many the long distance relations. Another reason is the distribution of operations. The instances of operations in our experimental corpus are not balanced. The operation "***reduce***" is the least (7.8%) and it is far less than other operations. Therefore the instances for creating the model of operation "***reduce***" are not satisfactory.

In the experiment of utilizing root finder, we tried to adopt the root information to the analyzer. However, the performance is worse than using our Nivre analyzer with out this method. The third row of Table 4-2 shows that dividing the sentence into two sub-sentences can improve our dependency analyzer. Using root finder and dividing sentence can reconstruct some mistakes in sentences. Certainly, the performance of the root finder influences the analyzer strongly. To improve the accuracy of the root node finder can increase the performance of using this improvement in our Nivre analyzer. The third row of Table 4-2 shows the results of combining the two proposed methods (using global features and root node finder) to improve our Nivre analyzer. Combining two methods can increase the dependency accuracy better than using either one of the methods.

The final two rows in Table 4-2 show the results of the experiments that using our MST analyzer with and without dividing the sentence. The dependency accuracy of our MST analyzer is better than our Nivre analyzer that combines with improvements. However, in the sentence accuracy, our MST analyzer is worse than our Nivre analyzer. As we discuss in section 4.2.2, the error in our Nivre analyzer will "multiply". An error operation will multiply more errors in the latter analysis. Inversely, the errors in using our Nivre analyzer centralize in some sentences. However, the errors in using our MST analyzer distribute equally in all sentences. For example, if our MST analyzer analyzes a sentence that includes twenty words and the output dependency structure includes one incorrect word, the dependency accuracy of the sentence is 19/20=0.95 but the sentence accuracy is 0/1=0.

The result of using the root node finder and dividing the sentence to improve our MST analyzer is not significant. The reason is that this improvement is used to resolve the error propagation problem of our Nivre analyzer. Because this error propagation problem does not occur in our MST analyzer, this improvement cannot improve our MST analyzer significantly.

For constructing our temporal relation analyzer, we require that the dependency analyzer has high performance both in dependency accuracy and in the sentence accuracy. Also, considering the hardware requirement of using our Nivre analyzer and our MST analyzer, using our Nivre analyzer with the two improvement methods is available. Or we can improve the calculation speed of our MST analyzer. The future work of our dependency analyzer includes four points. First, we can improve the performance of the root finder. Second, we should construct a useful prepositional phrase chunker, because the prepositional phrase is a major error source of our Nivre analyzer. The original analyzer tends to let the preposition governing a partial subtree of the full phrase. Intuitively, if we can extract the prepositional phrases from sentence, the complexity of the sentence will decrease. Thus an important task is how to chunk the prepositional phrase in the sentence. Finally, to improve the training / analysis speed of SVMs is an important future work. Because we use SVMs in our MST analyzer, the instances of training data is unbalance (the negative instances are huge and the positive instances are few) and the amount of the training instances multiply by the length of sentences. If we can use some methods to resolve these problems, our MST analyzer will become more useful.

	Dep. Acc.	Root Acc.	Sent. Acc.
Nivre algorithm with SVMs	85.25	86.18	59.98
Nivre analyzer with two-step process	85.44	86.22	60.1
Nivre analyzer with root node finder	86.13	**90.94**	**61.33**
Nivre analyzer with two-step process and root node finder	86.18	**90.94**	**61.33**
MST parsing algorithm with SVMs	87.76	88.36	57.37
MST analyzer with root node finder	**87.80**	**90.94**	58.69

Table 4-2: The experimental results of the improvement for Chinese dependency analyzer.

4.4 Summary

In this chapter, we describe the construction of the dependency analyzer which is a dependency analyzer is a central unit. We developed a machine learning based dependency analyzer for Chinese. First, we introduce two algorithms of dependency analysis and compare the performance of the algorithms for Chinese. We had implemented the dependency analyzer by two algorithms- Nivre's algorithm (Niver, 2004 [62]) and maximum spanning tree algorithm (McDonald, 2006 [59]). To compare the performance of these parsing algorithms in multi-lingual dependency analysis helps us to decide an appropriate approach for Chinese dependency analyzer. We experiment the parsers in the multi-lingual dependency analysis shared task at the Conference on Computational Natural Language Learning (CoNLL) 2006 and 2007.

Second, we consider the properties of Chinese then propose some methods to improve the performance of the dependency analyzer. We present a method for improving dependency analyzer of Chinese that based on Nivre's algorithm. We find that there are two problems in our Nivre analyzer and propose two methods to solve them. One problem is that some operations cannot be solved only using local feature. We utilize the global features to address this. The other problem is that this bottom-up analyzer doesn't use top-down information. We supply the top-down information by constructing SVMs based root node finder to address this problem. Experimental evaluation on the Penn Chinese Treebank Corpus shows that the proposed extensions improve the parsing accuracy significantly. However, the result of using the root node finder and dividing the sentence to improve our MST analyzer is not significant. The training / testing speed of our MST analyzer is time-consume. Therefore, we apply the Nivre analyzer into our temporal relation identifier but will try to resolve the time-consume problem of our MST analyzer.

Chapter 5 Constructing a Temporal Relations Tagged Corpus

Main theme of this thesis is constructing a machine learning based temporal relation analyzer for Chinese. We need a temporal relation tagged corpus for the machine learner. We first annotate a basic corpus manually, then training the temporal relation analyzer on the corpus. In this chapter, we describe the guideline of our temporal relation tagged corpus. In section 5.1, we introduce the environment of our annotating work. In section 5.2, we describe the attributes that exist in our corpus. Because our corpus focuses on the temporal relations between verbs, we need to observe the behavior of the verbs in Chinese. In section 5.3, we describe the verb-event classification. In section 5.4, we introduce the possible temporal relation types between two events. In section 5.5, we report the progress of our annotating work until now. Finally, we compare our criteria and TimeML in section 5.6.

5.1 Basic data and annotation tools

Our temporal relation analyzer focuses on the dependency structure of the sentences to analyze the temporal relations between events. To recognize subordinate event pairs and head-modifier event pairs in a sentence, we need a dependency parsed corpus for using the information of the dependency structure. We used the Penn Chinese Treebank (Palmer, et al., 2005 [65]) as the original data. Since, the Penn Chinese Treebank does not include the head-modifier relations; we transformed phrase structures into dependency structures using head rules (Cheng, 2005 [14]). The head rules decide the head word of each phrase in the phrase structure, and then the phrase structure becomes a dependency tree. We annotate the temporal attributes and the temporal relations of events on a part of the Penn Chinese Treebank. Our corpus contains 151 Chinese news articles with 7239 events, 1945 sentences and 49691 tokens.

The punctuation "，" usually can be used in the semantic ending of a sentence in Chinese. To distinguish the meaning of the punctuation mark "，" is difficult. We define that the end mark of a sentence is the punctuation "。" (a full stop) in our corpus. Because a sentence in the Treebank could include several clauses which denote independent events, the average length of sentences in the Penn Chinese Treebank is 27 words (507222 words / 18782 sentences). This is a property of the news articles. Therefore, we require that the dependency analyzer of our system should robust in analyzing the long sentences (See section 4.3).

We introduce the XML format for our data like TimeBank. Using XML format administers to the modification and the publication. Similar to the annotating work of TimeBank, we use the XML editor "<oXygen/>[22]" for our annotating work. Figure 5-1 illustrates the window of the XML editor "<oXygen/>". It includes three sub-windows: "Token sequences window", "Attribute information window" and "Original text window". "Original text window" shows the original XML data. Because the information in this window is hardly visible, our annotators almost do not need to refer to this window in annotating work. "Token sequences window" is more visible for our annotators. This window shows the word sequences of the text. The elements in the token sequences window include "EVENT" and "WORD". The annotating targets (events) are tagged with an attribute name "EVENT" (for example, the element "EVENT 報名 w11" is annotating target), and the element name "WORD" is the information attributes of the word. The annotating work that our annotators will do is to annotate the annotating targets "EVENT". If the annotator clicks an annotating target in the token sequences window, "Attribute information window" will show the information of this element. An example of the

[22] http://www.oxygenxml.com/

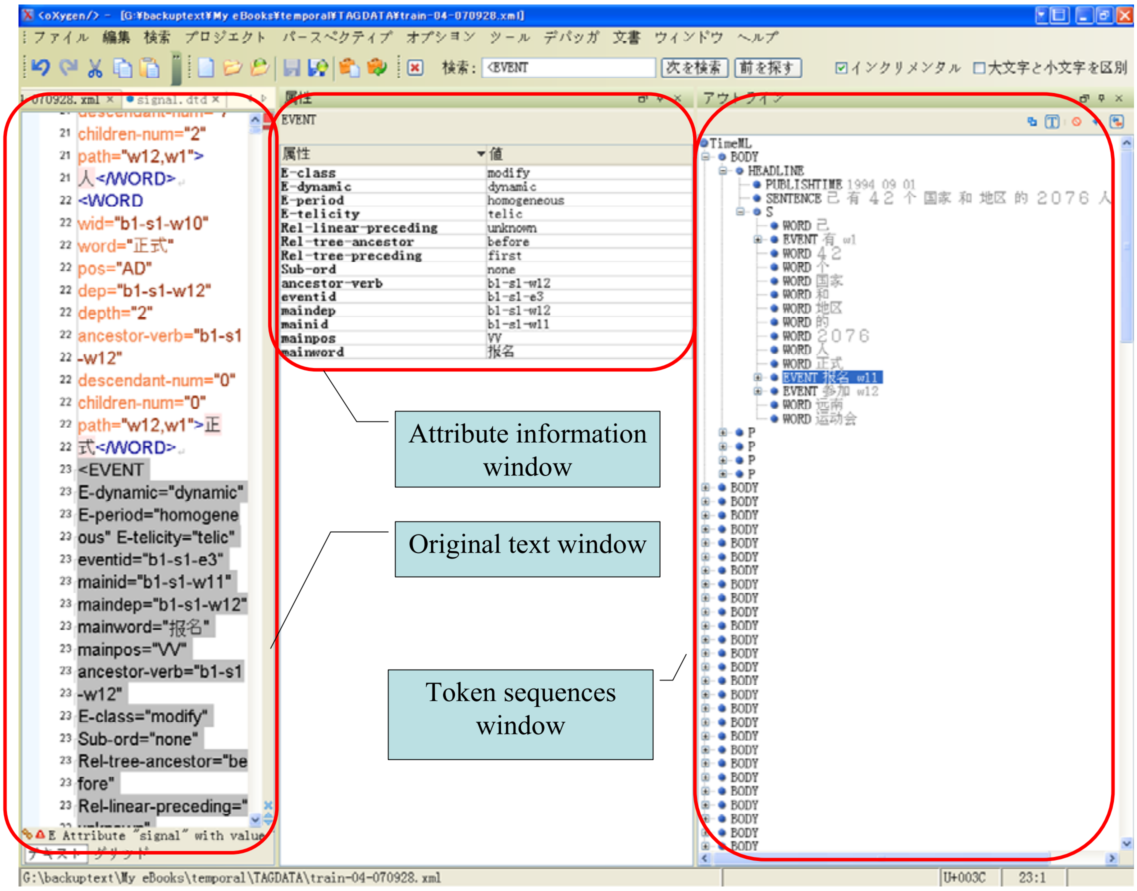


Figure 5-1: The working window for the annotator

Figure 5-2: Attribute windows for annotators.

attribute information window is illustrated in Figure 5-2. If the annotator clicks a word element, the attribute information window will show as the left side of Figure 5-2. The annotators refer the information in this attribute information window and annotate the attributes of the annotating target element.

5.2 The temporal Information Annotation Guideline

We annotate the two types of the temporal attributes of events: the properties (event class, dynamic, period and telicity) and the temporal relations for limited event pairs (adjacent event pairs, head-modifier event pairs, sibling event pairs and subordinate relations). Some information of words and events can be annotated automatically, such as the POS-tag, head word, the path to the root of the sentence, and so on. The annotator refers to the automatic annotated information to decide the most appropriate attributes of the temporal relations and temporal properties of each event. Figure 5-2 shows the attribute windows of an element in a token sequence. We introduce the attributes of morphological-syntactic information in section 5.2.1-section 5.2.3 and the attributes of an event in section 5.2.4.

5.2.1 The attributes of morphological-syntactic information

The left side window in Figure 5-2 shows the morphological information and the dependency information of a word, and the definition of these attributes is described in Table 5-1. The most right column of Table 5-1 is an example of the morphological-syntactic information that is extracting from a dependency structure. The morphological information attributes are the basic information of a word, these include: "wid (word ID[23])", "word", "POS" and the special attributes ("TMP", "verb-class" and "signal"). Some morphological information, such as the "word" tag and the "POS" tag are

[23] In our corpus, the format of word ID and event ID is "bxx-sxx-w(e)xx". The prefix character "b" means the id number of the text. The character "s" means the id of the sentence that the focus word / event exists. The character "w / e" means the id of the focus word / event.

similar to the original Treebank. The “TMP” tag refers to the phrase tag “*-TMP[24]” in the Treebank. The attributes “verb-class” and “signal” will be explained in section 5.2.2 and the section 5.2.3. All of these morphological information attributes are analyzed automatically.

example:

已/有/42個/國家/和/地區/的/2076人/正式/報名/參加/本/運動會

(2076 people from 42 nations and regions have registered officially to participate this gymkhana.)

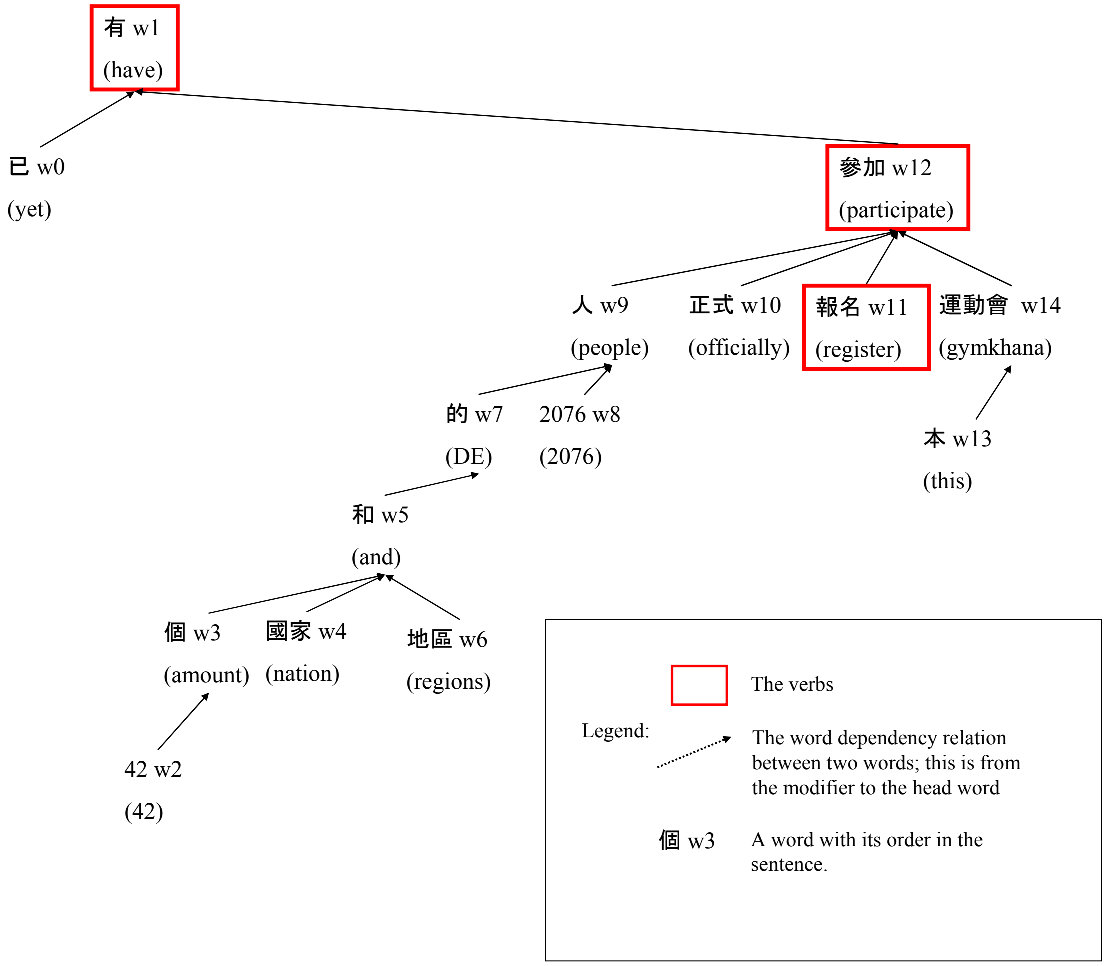


Figure 5-3: An example of using the information extracting from a dependency structure.

---

[24] The Treebank includes “NP-TMP” (nominal phrase), “PP-TMP” (prepositional phrase), “LCP-TMP” (Localization phrase) and others.

The other attributes in Table 5-1 are dependency information. Figure 5-3 illustrates an example of using the information extracting from a dependency structure. The word dependency relation is explained by the arrows in the figure. The arrows point from the modifier to its head word. We extract the dependency information of the word from the dependency structure. The attribute "dep" is the head word ID of the focus word. In the example in Table 5-1, it is the word "b1-s1-w5" that corresponds to the word "和 w5" in Figure 5-3. The attribute "path" of a word "國家" is the path that starts from this focus word to the root node of the dependency structure. That is, the path {"和 w5", "的 w7", "人 w9", "參加 w12", "有 w1"}. The attribute "ancestor-verb" is the ancestor verb of the focus word. It is the first verb that exists in the root path that we describe here. Therefore the attribute "ancestor-verb" is different from the attribute "dep". In this example, the first verb in the path is the verb "參加 w12". The attribute "depth" is the length of the root path. It is five in this example. The attributes "children-num" and "descendant-num" mean the number of children and the descendants of the focus word in the dependency structure. These attributes are both zero in this example. In other word, such as the word "和 w5" in the figure, it has three children ("個 w3", "國家 w4", "地區 w6") and four descendants ("個 w3", "國家 w4", "地區 w6" and "42 w2").

The other morphological information of a word includes the attributes "verb-class" and "signal". We introduce these attributes in next two sub-sections.

Attribute	Definition	Example
The dependency information		
ancestor-verb	The ancestor verb of the focus word	b1-s1-w12
children-num	The number of children of the focus word	0
dep	The head word ID of the focus word	b1-s1-w5
depth	The depth of the focus word in the dependency tree	5
descendant-num	The number of descendants of the focus word	0
Path	The path from the focus word to the root of the dependency tree	w5,w7,w9,w12,w1
The morphological information		
TMP	Is the focus word a part of a temporal expression? (yes or no)	No
pos	POS tag	NN
signal	Is the focus word a signal word? (yes or no)	No
verb-class	The temporal meaning class of the verb ("state","change","action" and "mental")	Non-verb
wid	The ID of the focus word	b1-s1-w4
word	The focus word	國家

Table 5-1: Attributes of the word "國家".

5.2.2 The attribute "verb-class" of a word

The attribute "verb-class" is a concept class of verbs. The verbs in Penn Chinese Treebank include four POS-tags (VV, VA, VC and VE). For giving more semantic information of verbs, we define four classes of the verbs to describe the activity concept: "state", "change", "action" and "mental." This verb class is different from the verb event classification that we describe in section 5.3. The verb class here is the essential action type of a verb. It does not depend on the context and is an intuitive image when a native speaker read the verb. For example, when a native speaker reads the verb "爆炸[25] (to explode)", she/he will image that a bomb or gunpowder is exploding then on fire.

We classify manually 23979 verbs that are extracted from GOH's research (2006 [29])[26] into these four classes. The definition and number of each classes is shown in Table 5-2.

[25] This word can be both a noun and a verb, but we consider the verb case here.

[26] In GOH's research, they collected the words in Penn Chinese Treebank and CKIP corpus as a basic dictionary. Then, they extended the dictionary to more than 160,000 words by their proposed method.

The class “state” describes a statement or a static situation, such as “齊全 (well-appointed)” and “保有 (to hold on)”. Most of the verbs of this class are the verbs with the POS-tag “VA (predicative adjective)”. These words are adjectives and become predicates in the sentences. The class “change” describes the change of statement, such as “變成 (to become)” and “長大 (to grow up)”. These verbs usually explain the occurrence that a statement transforms to another. The class “mental” describes a psychological action or state, such as “認爲 (to think)” and “討厭 (to hate)”. The last verb class is “action”. This class includes normal actions and uncertain verbs. The normal actions such as “爆炸 (to explode)” are dynamic occurrences. These verbs are most common class in the dictionary. Because some verbs can not be classified into single verb class, these verbs are grouped into the class “action”. For example, the verb “駐防 (to garrison)” could be a statement that troops stay in some place or could be an action that the troops are garrisoning the place. A native speaker cannot decide the class of this verb when he is reading it. However, this kind of verbs tend to be an action in a context, therefore we group these verbs into the class “action”.

Actually, we require a lexicon of event semantics, such as Lexical Conceptual Structure (Jackendoff, 1992 [38]), to classify the verbs in our dictionary. However, there is no Chinese lexicon with event semantic information which covers the verbs in our dictionary. Therefore, we classify the verbs to the four classes manually before we annotate the corpus. The temporal relation annotators are not required to classify the verbs when they annotate the corpus.

We assume that the concept of verb class is important information for recognizing the temporal relations between two events. The verbs “射擊 (to shoot)” and “認爲 (to assume)” are in different classes. The class of the former verb “射擊 (to shoot)” is “action” and it usually means a time-bounded action (short period or instantaneous). The class of the latter verb “認爲 (to assume)” is “mental” and it usually means a mental statement with long continuance.

It should be noted that the “verb-class” needs not to mean temporal properties (such as the occurrence period of a verb) of the verb. The temporal property of a verb could change in contexts. For example, someone may think that a mental verb “認爲 (to assume)” is a statement without time boundary. However, in the following sentence:

- “六歲/時/, /我/*認爲*/真有/聖誕老人 (when I was six years old, I *believed* that Santa Claus exists.)”

The verb “認爲 (to assume)” has an approximate period “六歲時 (in six years old)”. Therefore, the annotators need not to annotate the events by referring to the verb class attribute. It is a feature for our annotating system (see section 6.2).

Verb class	Definition	example	Number
state	a statement or a static situation	齊全, 保有	2160
change	the change of statement	變成 , 長大	950
mental	a psychological action or state	認為, 討厭	1187
action	normal action and uncertain verbs	射擊,爆炸	19681

Table 5-2: The definition of the verb class and the number of each class.

5.2.3 The attribute “SIGNAL” of a word

A SIGNAL is a textual element that makes explicit the relation between two temporal entities. SIGNAL definition in TimeML includes temporal prepositions, temporal conjunctions and prepositions signaling modality. Briefly, the original signals are composed by prepositions or conjunctions. A SIGNAL word could mean temporal or non-temporal relations depending on the contextual information. In the sentence in Figure 1-3:

- “我/昨天/早上/六點/**起床**/, /**吃過**/早餐/後/**搭**/公車/上學” (I **waked up** at 6 A.M. yesterday, **ate** breakfast and then **go to** school by bus.)

The word “後” (after, then) is a SIGNAL word and describes that the event “吃早餐”(to eat breakfast) occurs before the event “搭公車” (by bus, to take the bus). However, the same word “後” (after, then) in the sentence “屋/後/有/個/花園” (There is a garden behind the house) means a location relation. As we discuss in Chapter 1, temporal adverbs in Chinese news articles sometimes be abbreviated. Hence verbs lack the tense / aspect information itself for analyzing the temporal relations. However, the signals in news articles are scarcely abbreviated. The signals are important clues for temporal relation analysis.

Candidate words of SIGNAL in Chinese are limited. We collect these SIGNAL candidates according to the POS-tag standard of CKIP's corpus (CKIP, 1993 [19]), which list the SIGNAL candidates in the POS-tag "Ng" (Localizer) and some prepositions. According to the introduction of the POS-tags in CKIP Treebank, they listed all prepositions that occurred in the corpus and given the usage of each preposition. We refer to the usages then select the temporal prepositions as the candidates of SIGNAL words. Table 5-3 lists the candidates of the SIGNAL words. The signals in CKIP Treebank can be identified according to their POS-tags. However, these SIGNAL candidates in Penn Chinese Treebank are not listed and spread in the prepositions ("P"), conjunctions ("CC") and localizers ("LC"). The POS-tags in Penn Chinese Treebank do not define detailed classes. We cannot identify which words are signals according to their POS-tags. For recognizing the SIGNAL automatically, we use the SIGNAL candidate list shown in Table 5-3 to annotate the words that correspond to the "possible signal" list. A SIGNAL candidate word has an attribute with two values: "time" and "non-time" to describe if it is a temporal SIGNAL word or not.

The attribute "SIGNAL" is tagged automatically in our temporal relation analyzing system. Therefore we need to train the machine learner with a tagged corpus (see section 6.2). This SIGNAL tagged corpus is identical to our temporal relation corpus. Our system refers to Table 5-3 to extract all candidate words of SIGNAL in corpus. Then we require annotators to classify these candidate words manually. The result that to distinguish the use of SIGNAL words in different context by a machine learning classifier and the automatic SIGNAL word tagger which based on machine learning is described in Chapter 6.

For applying the SIGNAL words in automatic temporal relation annotating, we define the intuitional meaning of SIGNAL words. These meanings are also listed in Table 5-3. We assume that the intuitional meanings are the relation between the ancestor event and the descendant event which is in the same clause with the SIGNAL word. For example, in the sentence "**吃過**/早餐/後/搭/公車/**上學** (after **ate** breakfast, I **went to school** by bus)", the SIGNAL word "後 (after)" indicates the ancestor event "吃過 (ate)" occurs after the descendant event "上學 (to go to school)". Therefore we define the intuitional meaning of the SIGNAL word "後 (after)" as "after". We divide them to four intuitional meanings--

"after", "before", "during" and "simultaneous". The SIGNAL words without the intuitional meaning do not have clear intuitional meaning.

Preposition											
signal	meaning	signal	meaning	signal	meaning	signal	meaning	signal	meaning	signal	meaning
自從	after	等到	after	到	before	截止	before	打從	after	有	
等	after	臨近	before	隔	after	待	before	打	after	距離	
正當	simultaneous	及至	before	趁	simultaneous	俟	before	直到	before	離	
逢	simultaneous	待到	before	趁著	simultaneous	留待	before	迄	before	距	
臨	simultaneous	從	after	乘	simultaneous	繼	after	至	before	趕	
臨到	simultaneous	每逢	simultaneous	延至	before	遲至	before	値	simultaneous	比及	
當著	simultaneous	在	simultaneous	乘著	simultaneous	於	simultaneous	當	simultaneous		
Localizer											
signal	meaning	signal	meaning	signal	meaning	signal	meaning	signal	meaning	signal	meaning
後	after	之後	after	以降	after	之時	simultaneous	而外		初	
之前	before	以後	after	之際	simultaneous	時	simultaneous	一樣		左右	
以前	before	當中	during	之初	simultaneous	開始	after	上去		外	
前	before	中間	during	起	after	以來	after	下來		末	
以下	after	之間	during	之內	during	也似		之左		之餘	
以上	before	間	during	以內	during	似		之右		之秋	
上	before	之中	during	來	after	一般		不等		前後	
過後	after	爲止	before	之交	simultaneous	中		上下		內	

Table 5-3: The candidate list of SIGNAL word.

5.2.4 The attributes of an event

The definition of EVENT is based on the TimeML (see section 2.2): an entity describes a situation of happen, occur, state and circumstance. However, we limit an event to the one expressed by a verb (we describe this more in section 5.3) in our guideline. According to the interpretation in the guideline of Penn Chinese Treebank, verbs serve as the predicate of a main clause or the embedded clauses in corpus (Xia, 2000 [77]). We assume that a verb in a clause can be thought as the representative entity of an event that the clause describes.

The right side window in Figure 5-2 shows the attributes of the focus event. Table 5-4 describes the attributes of an event. The attributes of an event include three parts:

- The information of the focus event (eventid, maindep, mainid, mainword, mainpos and ancestor-verb)
- The temporal properties of the event (E-class, E-dynamic, E-period, and E-telicity)
- The temporal relations (Rel-liner-preceding, Rel-tree-preceding, Rel-tree-ancestor and Sub-ord).

The attributes of the main verb are extracted automatically from the morphological information and the dependency information of the verb. Since we define that each verb is an event, the information of the main verb is quoted for the information of the event. The attributes "maindep, mainid, mainword, mainpos and ancestor-verb" correspond to the attributes of a word that we define in Table 5-1: "dep, wid, word, pos, and ancestor-verb". Annotators refer these attributes to annotate the temporal property and the temporal relation of the event.

Properties of an event are the temporal characteristics of the focus event. These are different from the verb class that we describe in section 5.2.2. In these properties, the attributes "E-dynamic, E-period, and E-telicity" roughly correspond to the classification of verbs in (Dorr and Olsen, 1997 [24]). These temporal characteristics mean the telicity, the dynamic characteristic and the occurrence period of an event. The examples and the definitions of each value are shown in following representation (the italicized words are the focus events):

- E-dynamic:

  state: the event describes a truth, a static result of an action and a mental situation
  Ex: 我/***知道***/他/是/學生 (I ***know*** that he is a student.)
  dynamic: the event describes an action and the process of an occurrence
  Ex: 他/***損壞了***/新/裝備 (He ***broke*** the new device.)

- E-period:

  durative: the event occurs in a boundary or non-boundary time period.

  Ex: 我/正在/***排隊***/買/遊戲/軟體 (I ***am queuing*** for buying the game soft.)

  instantaneous: the event occurs in a short time period that the period is close to zero.

  Ex: 炸彈/***爆炸了*** (the bomb ***explored***)

  repeat: a durative or a instantaneous event repeats by a time interval.

  Ex: 我/每天/***上學*** (I ***go to school*** every day.)

- E-telicity:

  telic: the occurrence time period of an event has a predicable ending point.

  Ex: 空地/上/正在/***興建***/公寓 (An apartment ***is building*** in the clearance)

  non-telic: the occurrence time period of an event does not have a predicable ending point

  Ex: 他/正在/***跑步*** (He is ***running***.)

  continue-state: an event describes a result statement of its occurrence.

  Ex: ***進駐***/首都/的/部隊 (The troops that ***garrisoned*** the capital.)

We think that these temporal properties of an event are useful features for machine learner to analyze the temporal relations. For example, if the focus event is an instantaneous event, it would not have a temporal relation "include"(see section 5.4) to another event. Or if an event is a state event, it usually has a long occurrence period then it would "include" other events in the article. These properties can describe the verb classification by Vendler (1967 [74]) or other verb classification (Li et al., 2005 [50]) by the combination of the attribute values. Table 5-5 illustrates the comparison between our temporal characteristics of an event and Vendler's verb classification. It should be noted that our temporal characteristics is the features of an "*event*" and other related research focus on "*verbs*". The reason that we do not adopt the concept of Vendler's verb classification is that the temporal characteristics of a verb change in different context.

For example, a quantifier or a temporal expression will change the telicity of the event, such as:

- 他/正在/***跑步*** (He is ***running***.)

- 他/正在/***跑***/三千公尺 (He is ***running*** for three kilo-meters.)

In the upper sentence, the reader recognizes an event—“someone is running”, but the finished time point of the event is uncertain (the man could run few minutes or several hours). We regard the event in this sentence as a non-telic event. However, in the lower case, the amount of the running is described (3000 meters). This event is a telic event. For example, an activity verb could have both telic and non-telic values that depend on the context. Therefore, the temporal properties of an event are ambiguous in the direct verb classification. We do not adopt Vendler’s verb classification instead of the temporal properties as the feature of temporal relation analysis. Before using the temporal properties for machine learning, we need to construct a machine learning model for annotating the temporal properties automatically. The annotators obey the definition of temporal properties to annotate these attributes.

Another attribute of the temporal properties is the attribute “E-class”, it is the actuality of the event, that is, the event is real a happened event or not, or annotators should consider the temporal relation of the event or not. This event class depends on the usage of verbs in different situations. We describe this attribute in next section. The temporal relation tags of an event (Rel-linear-preceding, Rel-tree-preceding, Rel-tree-ancestor, and Sub-ord) are main attributes that we require the annotators to annotate. Because we need more discussion for the temporal relation tags, we describe these attributes and define each value of attributes in section 5.4.

Attribute	Values	Definition
information of the main verb		
ancestor-verb		The ancestor verb of the main verb of the focus event
eventid		The ID of the focus event
maindep		The head word ID of the focus word
mainid		The ID of the main verb
mainpos		The POS tag of the main verb
mainword		The main verb
the temporal properties of the focus event		
E-class	actual, hypothetical	Actuality of the focus event
E-dynamic	state, dynamic	Activity of the focus event
E-period	repeat, durative, instantaneous	Period of the focus event
E-telicity	telic, non-telic, continue-state	Telicity of the focus event
the temporal relation tag of the focus event		
Rel-linear-preceding	Relations in **Figure 5-6**	Relation between the focus event and the linear adjacent preceding event
Rel-tree-preceding	Relations in **Figure 5-6**	Relation between the focus event and the sibling event
Rel-tree-ancestor	Relations in **Figure 5-6**	Relation between the focus event and the ancestor event
Sub-ord	introduce, explanation, condition, none, report, passive, possibility	Subordinate type between the focus event and the ancestor event

Table 5-4: Attributes of an event.

verb class / feature	state	activity	accomplishment	achievement
dynamic	state	dynamic	dynamic	dynamic
period	Durative	durative/ instantaneous/ repeat	durative	Durative/ instantaneous
telicity	non-telic/ continue-state	telic/non-telic	telic	telic/ continue-state

Table 5-5: The comparison between our temporal characteristics and Vendler's verb classification

5.3 Verb events and the event classification

Our research focuses on the relations between events and limits the events to the verbs. Verbs can be identified according to the POS tag of the word automatically (the POS-tag: VV, VA, VC and VE). Most of the verbs in treebank are the POS-tag "VV", it includes major verbs, such as raising predicates ("可能"(may be)), control verbs ("要"(want)), physical action ("飛"(fly)), psychological action ("討厭"(hate)), and so on (Xia, 2000 [77]). The POS-tag "VA" is the predicative adjectives, such as "齊全"(well-appointed). We consider the predicative adjectives as an event because these predicative adjectives usually describe a statement. The predicative adjectives can modify a noun in other context, but in these cases, the POS-tag of the predicative adjectives is an adjective "JJ". The difference can be analyzed in the step of morphological analysis or be distinguished in the original Treebank. Therefore we also recognize the predicative adjectives as a type of event according to the POS-tag. The POS-tag "VC" is the copula verb such as "是" (is). It describes a statement of a truth, such as the verb "是" (is) in the sentence: "我是學生" (I am a student), and we define these verbs as EVENT. The POS-tag “VE” describes the possessive or existential statement, such as the verb “有” (have) in the sentence “我有一本書” (I have a book). All these types of verbs are EVENTs and have the annotatable attribute in our criteria.

Verbs in an article include the events in actual world (which describe actual situations or actions) and the events in hypothetical world (which describe possible situations, imagination and the background knowledge). This definition is similar to the generalized Japanese modality of Kudo’s research (2004 [42]). The difference between our definition and Kudo’s generalized modality is that their generalized modality includes three types: “actual”, “actual-hypothetical” and “hypothetical”. “actual” in Kudo’s research is almost regarded to “actual world” in our definition. We group “actual-hypothetical” and “hypothetical” to “hypothetical world” in our definition. Therefore we only distinguish two different event types-“actual world” and “hypothetical world”.

In the first annotating working, we did not define the class of event types. We requires the annotators to decide the attributes of temporal relations of a verb by annotators’ knowledge but do not describe the difference definition between events (verbs) in actual world and hypothetical world. However, our annotators are confusing when they annotate the temporal relation corpus. In this section, we attempt to give the definition of actual / hypothetical world events (verbs). We investigate the usages of verbs in Penn Chinese treebank and classify the verbs to actual / hypothetical world.

5.3.1 Actual world and Hypothetical world

Because all verbs in our annotating work are regarded as events, the verbs as hypothetical world event are also included in the events. For example: (the italicized in our examples indicates the verbs)

- (a) 工業區/成立 **1**/後/大量/吸引 **2**/外資 (after the industrial estate **was established1**, it **attracted2** a great deal of foreign capital)
- (b) 工業區/成立 **1**/後/可能/大量/吸引 **2**/外資(after the industrial estate **is established1**, it can **attract2** a great deal of foreign capital)

The difference between example (a) and (b) is only without or with the word "可能 (can)". The word "可能 (can)" governs a verb phrase and explains a possible situation. It should be noted that verbs in Chinese do not have the morphological change according to the tense. The complete meaning of the examples should consider the global context of original sentence. The example (a) explains an actual world event that the industrial estate attracted a great deal of foreign capital. However, the word "可能 (can)" changes the phrase "大量/吸引/外資 (to attract a great deal of foreign capital)" as a hypothetical world event in example (b). The phrase presents a possibility and does not indicate an event in the actual world.

Considering the temporal relation between the verbs "成立(establish)" and "吸引(attract)", the temporal relation in the example (a) means that "the event 成立(establish) occurs before the event 吸引(attract)". Whereas in the example (b), the verb "吸引(attract)" indicates a possibility. We cannot make sure if it could happen. We think that the temporal relation in the example (b) is unidentifiable. Because we require the annotators to decide the temporal relation between the verbs "成立(establish)" and "吸引(attract)", we need to investigate the difference between the actual world and the hypothetical world. We address the issue by introducing the event types of verbs.

Aside from the problem of the actual world event and the hypothetical world event, the verbs in our temporal relation tagged corpus include some incomprehensible events. For deducing these problems, we investigate the different usages of verbs in the Penn Chinese Treebank then give a clear classification of event types. We use this classification of events as a clue for annotating the temporal relation between events. This event classification corresponds to the attribute "E-class" in Table 5-4. In section 5.3.2, we will investigate the usages of verbs in Penn Chinese Treebank and classify the event types according to their usages.

Figure 5-4[27] summarizes the event types of verbs. We divide the verbs to two rough types "hypothetical world" and "actual world". Each type includes several sub-types (we define the types in next section). Annotators annotate the attribute of event types (this is the attribute "E-class" that we describe in section 5.2.4) for each verb in the corpus. The definition of each event type in following sections is a guideline for our annotators. This

[27] The brackets with a character in each square refer to the example in 5.3.2.

attribute "E-class" has two values "actual world" and "hypothetical world". Although the types of values are coarse-grained, this attribute can describe whether a verb can be recognized as an event with understandable temporal relation on timeline or not.

However, the value "hypothetical world" of the event types means not only that the verbs with this value are temporal relation un-recognizable verbs, but also that the verbs with this value are "local recognizable" events. For example:

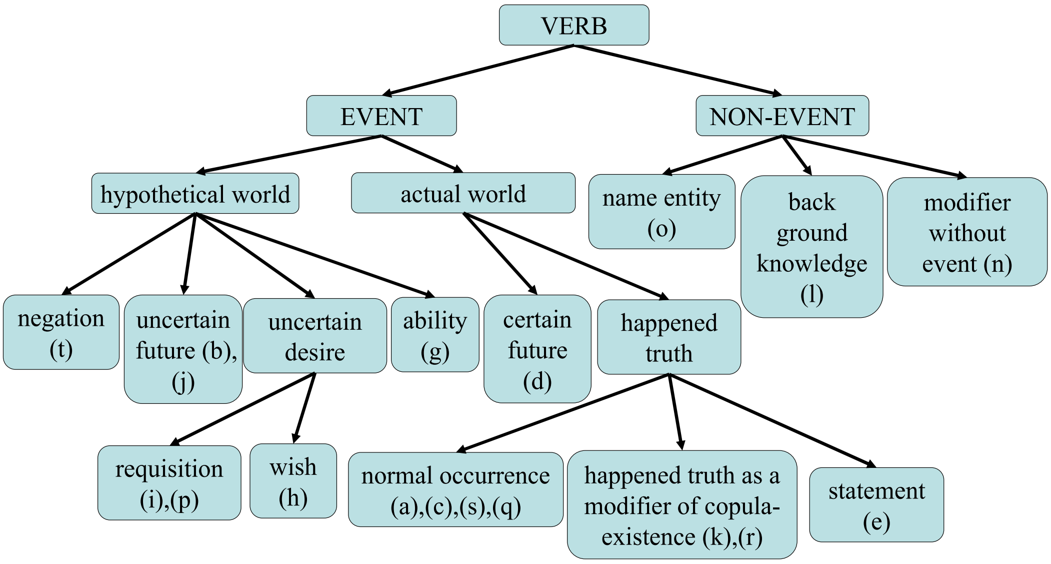
- (p) 他們/<u>**希望 1**</u>/政府/<u>**增加 2**</u>/預算/來/<u>**修補 3**</u>/堤防 (They <u>**wish1**</u> the government to <u>**increase2**</u> budget to <u>**repair3**</u> the bank)

The verb "希望 (wish)" governs the verb phrase "政府/增加/預算/來/修補/堤防 (the government increases budget to repair the bank)". Therefore the verb phrase describes a hypothetical world event (because we do not know if the government will repair the bank or not). However, considering the local context of the verb phrase, it includes two verbs that have a causal relation between them. The event "增加 (increase)" should occur before the event "修補 (repair)"[28]. The temporal relation between the two verbs exists on the local context. We do not ignore this kind of temporal relations and also require annotators to annotate them. The temporal relation between the verbs "增加 (increase)" and the verb "修補 (repair)" is not unknown but the temporal relation between the verbs "增加 (increase)" and the verb "希望 (wish)" is unknown.

Therefore, we can consider the attribute of event type is a "bridge" of going to the actual world or the hypothetical world. The event in the actual world means that an event can be ordered with the other occurred events on a timeline. The events in the hypothetical world cannot be ordered with the occurred events but can be ordered in their hypothetical world. It should be noted that the descendents of the "bridge" event of the hypothetical world also have the "E-class" value "hypothetical world". Figure 5-5 illustrates these concepts. The index on each event indicates the linear ordering of the event mentions in the article. The two events with rectangles are in actual world and the events with diamond shapes are in hypothetical world. The "E-class" of the events in two hypothetical worlds (hypothetical world 1 and hypothetical world 2) are all annotated as the value "hypothetical world".

---

[28] The government must increase the budget and pass the deliberation in the congress, and then the budget can be used to repair the bank.



Note: the characters in the brackets refer to the examples of each event type

Examples:

(a) 工業區/**成立1**/後/大量/**吸引2**/外資 (after the industrial estate **was established1**, it **attracted2** a great deal of foreign capital)

(b) 工業區/**成立1**/後/可能/大量/**吸引2**/外資(after the industrial estate **is established1**, it can **attract2** a great deal of foreign capital)

(c) 市場/**發生1**/火災 (A fire **occurred1** in the market.)

(d) 市政府/大樓/將於/年底/**完工1** (The construction work of the city hall will **finish1** at the end of the year.)

(e) 金融/市場/運行/**平穩1** (The function of financial market is **smooth1**.)

(g) 新港口/能/**停靠1**/大型油輪 (A big oil tanker can **berth1** at the new port)

(h) 他們/**希望1**/政府/**訂立2**/相關/法案 (They **wish1** the government to **legislate2** against affiliated bill)

(i) 政府/**要求1**/工廠/**改進2**/設施 (The government **requires1** the factory to **amend2** their equipments)

(j) 此/技術/**有助於1**/未來/**開發2**/新藥 (this technology can **help1** to **develop2** a new kind of medicine)

(k) 舊/法律/**是1**/三年前/**修訂2**/的 (The older version of bill **was1** **legislated2** at three years ago.)

(l) 該/公司/**是1**/世界上/最大/的/電力公司 (The company **is1** the largest electric power company in the world.)

(n) **提供1**/**新2**/的/動力 (**To provide1** a **new2** kind of power)

(o) "**解放1**/剛果/民主/同盟 (Alliance of Democratic Forces for **Liberating1** Congo-Zaire)"

(p) 他們/**希望1**/政府/**增加2**/預算/來/**修補3**/堤防 (They **wish1** the government to **increase2** budget to **repair3** the bank)

(q) 這批/企業/中/**有1**/十家/**完成了2**/公司/改組 (In these companies, **there are1** ten companies that **have completed2** the readjustment.)

(r) 武漢/**有1**/香港/投資/企業/2150家 (**There are1** 2150 Hong Kong capital companies in Wuhan.)

(s) 我/**有1**/一輛/車 (I **have1** a car.)

(t) 在/**發展1**/工商業/的/同時/,/ 他們/沒有/**放棄2**/農業 (During **development 1** industry and commerce, they have not **given up 2** the agriculture)

Figure 5-4: the event types of verbs.

There is no understandable temporal relation between actual world and hypothetical world. For example, the relation between the event 1 and event 3 is un-recognizable. The events in the same hypothetical world have their temporal relations for other events in the same hypothetical world. For example, the relation between the event 0 and event 2 can be annotated. However, a hypothetical world is independent of the other hypothetical worlds. Therefore, the temporal relation between event 2 and event 3 are understandable but the relation between event 3 and event 4 are unknown. We require our annotators to annotate the understandable temporal relations in each hypothetical world because the instances of the local context are useful training instances in analyzing the temporal relation between events in actual world by machine learning.

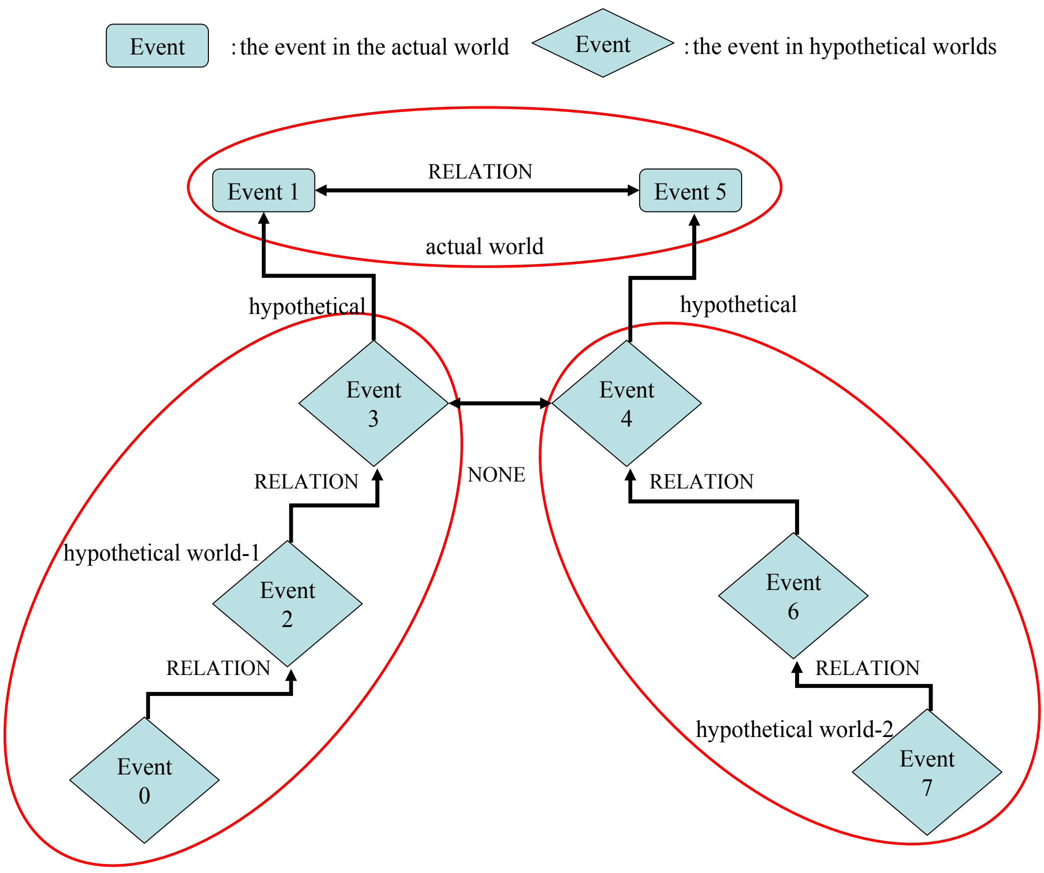


Figure 5-5: The actual world and hypothetical worlds in an article.

5.3.2 The classification of event types based on the usage of verbs

In this section we investigate the usages of verbs in different situations. We limit the events to the verbs according to the POS-tag of Penn Chinese Treebank. Therefore all the words tagged with the POS-tags “VA”, “VE”, “VC”, and “VV” are the “event candidates”. However, these POS-tags include not only the actual world events but also the hypothetical world event, the modifiers of nouns, and sub-segments of name entities. We will exemplify these situations and classify to actual world / hypothetical world. Referring to Figure 5-4, the rough types of event include the following sub-types:

- actual world: normal occurrence (example (a),(c)), statement (example (e)), certain future (example (d)) and happened truth as a modifier of copula (example (k))
- hypothetical world: ability (example (g)), wish (example (h)), requisition (example (i),(p)), uncertain future (example (b), (j)), modifier without event (example (n)), background knowledge (example (l)) and name entity (example (o))

SITUATION 1: Verbs of actual world events

The “event” that we want to annotate is an action or situation that has happened or will decidedly happen in the actual world. We define these events as the actual world event. For example:

- (c) 市場/<u>發生 1</u>/火災 (A fire <u>**occurred1**</u> in the market.)
- (d) 市政府/大樓/將於/年底/<u>完工 1</u> (The construction work of the city hall will <u>**finish1**</u> at the end of the year.)
- (e) 金融/市場/運行/<u>平穩 1</u> (The function of financial market is <u>**smooth1**</u>.)

The verbs in these examples are the actual world event. We want to distinguish between these events and other hypothetical world events.

The example (c) is a normal instance of an actual world event. The verb “發生 (happen)” in the sentence indicates an occurred fact. The verb “完工 (finish)” in example (d) is a confirmative result that will occur in future. The word “將於 (will)” indicate that the sentence describes a future statement. If there is no other statement that describes an accident in the context, we can trust that the event in the example (d) is an actual world event.

In Chinese, an adjective can be a predicative without the copula verb “是” (the copula corresponds to the verb “be” in English). The example (e) does not include a corresponding word of the verb “是”. The adjective “平穩 (smooth)” is a predication and explain an actual world situation. This kind of adjective is the POS-tag “VA” in Penn Chinese Treebank and also be defined as an actual world event.

SITUATION 2: Verbs of hypothetical world events

Some verbs indicate hypothetical world events. These verbs describe a possibility, a statement of ability, anticipation, a requisition and an inconclusive future. For example:

- (b) 工業區/成立 1/後/可能/大量/吸引 1/外資(after the industrial estate **is established1**, it can **attract1** a great deal of foreign capital)
- (g) 新港口/能/停靠 1/大型油輪 (A big oil tanker can **berth1** at the new port)
- (h) 他們/希望 1/政府/訂立 2/相關/法案 (They **wish1** the government to **legislate2** against affiliated bill)
- (i) 政府/要求 1/工廠/改進 2/設施 (The government **requires1** the factory to **amend2** their equipments)
- (j) 此/技術/有助於 1/未來/開發 2/新藥 (this technology can **help1** to **develop2** a new kind of medicine)
- (t) 在/發展 1/工商業/的/同時/,/ 他們/沒有/放棄 2/農業 (During **development 1** industry and commerce, they have not **given up 2** the agriculture)

The verbs “吸引 (attract)” in example (b) explains a possibility that “may” occur after a confirmative result “成立 (establish)” in future. We cannot order the temporal relation between the actual world event “成立 (establish)” and the possible event “吸引 (attract)” in the example (b), because we do not know if the event “吸引 (attract)” will come true.

The verb “停靠 (berth)” in example (g) explains an ability of the new port. The verb “停靠 (berth)” does not indicate a truth or a confirmative result. We could not confirm what time an oil tanker will berth at the new port. This verb is a hypothetical world event. The verb “訂立 (legislate)” in the example (h) and the verb “改進(amend)” in the example (i) explain a wish and a requisition. Even the government (in the example (h)) and the factory (in the example (i)) was required, there is no evidence that they will do the requirement. Although the wish and requisition will be realized in future, we cannot identify the time point of the realization of these events. Therefore we consider these verbs as the hypothetical world events.

The verb “開發 (develop)” in the example (j) explains an inconclusive plan in future. The developed technology can be used for a new development plan. However, we also cannot make sure if the development plan will be realized or not. We cannot identify the verb “開發 (develop)” on a timeline. Therefore this verb is a hypothetical world event.

The verb “放棄 (give up)” in the example (q) explains a negative fact. People “DO NOT” give up the agriculture according to a negative function word “沒有”. We can regard a hypothetical situation that they give up the agriculture, but this situation does not occur in actual world. Therefore, we define a negative fact as a hypothetical event.

These examples (from the example (b), (g) to (j) and (t)) indicate the hypothetical world events. However, as we introduce in the description of TYPE 1 (the examples (a) and (b)), the different types of events could include similar context in local structure. The difference between the example (a) and the example (b) is the word “可能 (can)”. To distinguish the actual world event and the hypothetical world event with similar local context, the dependency structure analysis is useful process (see Chapter 3).

SITUATION 3: Copula verb and Existence verb

There are two special POS-tags of verbs in Penn Chinese Treebank, VC and VE. These verbs are the copula and the existence verb in Chinese. The “VC” is copula verb (such as the verb “是 (be)”) that indicates existences and roughly (not perfectly) corresponds to the verb “be” in English. The “VE” verbs means an existence, such as the verb “有 (have)” is a verb with the tag “VE”. In TimeML, these copula and existence verb do not be considered as an independent verb. It is included in other verb phrase or in the nominal phrase that is an event. However, these special verbs are independent verbs following the Penn Chinese Treebank standard. We investigate how to deal with copula verbs and existence verbs. For example:

- (k) 舊/法律/是 1/三年前/修訂 2/的 (The older version of bill **was1** **legislated2** at three years ago.)

- (l) 該/公司/**是 1**/世界上/最大/的/電力公司 (The company **is1** the largest electric power company in the world.)

Considering the usage of copula in Penn Chinese Treebank, the sentences include copula verbs can be distinguished to two types. The copula verbs describe existences. These existences could be the verb phrases (the example (k)) or the nominal phrases (the example (l)). In the example (k), the verb phrase "三年前/修訂 (was legislated at three years ago)" is an event and the copula verb accentuates the existence of the verb phrase. Although there are two verbs in the example (k), the sentence only means the event in verb phrase "三年前/修訂 (was legislated at three years ago)".

Considering the dependency structure of the sentence, the copula verb is the root of the dependency structure and the verb phrase modifies the copula verb. In the example (k), we define the relation between the copula verb and the verb "修定 (to legislate)" as "copula-existence" (see section 5.4). Therefore we regard the copula verb in the example (k) as the main verb of the verb phrase "修訂 (to legislate)" and it is an actual world event[29].

The copula verb "是 (be)" in the example (l) accentuates the truth of the nominal phrase "世界上/最大/的/電力公司 (the largest electric power company in the world)". According the discussion in the previous paragraph, the meaning of this copula verb comes from the nominal phrase. We can recognize the nominal phrase as a truth at the time point "NOW" (the company is largest in the world now). However, this phrase does not indicate the existence period of the truth. We can regard it as the background knowledge and it does not include an event. To order this noun phrase with other actual world event on the time-line is impossible[30]. We also regard this copula verb as a hypothetical world event.

The POS tag "VE" means existence verbs in the treebank. We deal with the existence verbs similar to the copula verbs. The usages of the existence verbs also include two types:

---

[29] Whether the copula verbs are actual world events or hypothetical world events depend on the modifier verb phrases.

[30] We cannot know when the company became the largest one on the world. And other events in the context distribute in a shorter period on a timeline. Therefore to compare the existence period of the truth and other events is impossible. However, if a temporal expression with a passed time period in the context, the truth could have a boundary of occurrence time. Then the copula can be recognized as an actual world event.

- (q) 這批/企業/中/<u>有 1</u>/十家/<u>完成了 2</u>/公司/改組 (In these companies, **<u>there are1</u>** ten companies that **<u>have completed2</u>** the readjustment.)
- (r) 武漢/<u>有 1</u>/香港/投資/企業/2150 家 (**<u>There are1</u>** 2150 Hong Kong capital companies in Wuhan.)
- (s) 我/<u>有 1</u>/一輛/車 (I **<u>have1</u>** a car.)

The existence verb “有 (have, exist)” can be used to explain the possessory situation (the example (s)), an existence situation (the example (r)) and can be used to accentuate other event (the example (q)). The example (s) is a popular usage of the verb “有”. It simply describes the subject owns the object. The usage of the example (r) looks like the usage of the example (s) in Chinese. However, the meaning that this event focuses on is not the city Wuhan (武漢) “has (有)” the Hong Kong capital companies. The event focuses on the truth that the Hong Kong capital companies locate in Wuhan. This usage is similar to the usage of the copula verb “是” in the example (l).

The usage of the verb “有” example (q) is similar to the usage of the verb “是” in the example (k). In the dependency structure of the example (k), the verb “有” governs a verb phrase “完成了/公司/改組 (**have completed** the readjustment)”. We think that this verb “有” emphasizes the verb phrase. Therefore Therefore we regard the exstience verb “有” in the example (q) as the main verb of the verb phrase “完成了/公司/改組” and it represents an actual world event and the temporal relation between the verb “有” and the verb “完成” is the special value “copula-existence” (see section 5.4).

SITUATION 4: Non-event verbs

There are several situations that words have verbal POS-tag but cannot be recognized as events. These situations include the non-event predicative adjectives and the name entities. In Chinese, the adjectives can be the predicates and without other verbs. This kind of adjectives is predicative adjective and has a POS-tag “VA” in Penn Chinese Treebank. These predicative adjectives indicate statements. However, some instances of these predicative adjectives in the Treebank are close to the normal adjectives. We distinguish the difference between the predicative adjectives that describe situations and the predicative adjectives that are normal adjectives. For example:

- (e) 金融/市場/運行/<u>平穩 1</u> (The function of financial market is **<u>smooth1</u>.**)

- (n) 提供 1/新 2/的/動力 (**To provide1** a **new2** kind of power)

The adjective "平穩 (smooth)" in the example (e) indicates a statement. We regard this adjective as an actual world event. However, the adjective "新 (new)" in the example (n) is a modifier of the noun "動力 (power)". This adjective do not indicate a situation, therefore it is not an event.

Another situation of non-event verbs is the verb in a name entity. Following the POS-tagging strategy of Penn Chinese Treebank, a name entity is separated to several words and these words are tagged independently. For example:

- (o) "解放 1/剛果/民主/同盟 (Alliance of Democratic Forces for **Liberating1** Congo-Zaire)"

The full phrase in the example (o) is a name entity and the word "解放 (liberate)" has the POS-tag "VV" in Penn Chinese Treebank. However, this verb does not describe an actual event or hypothetical event. It is a substring of the name entity. We define this kind of verbs as non-event verbs.

5.4 The temporal relations between events

In section 3.2 and section 5.2.4, we describe that our proposal method is that constructing a temporal relation analysis system. This system analyzes the temporal relation between events (verbs). The attributes of an event include these temporal relations that are based on a viewpoint of the dependency structure. We describe the possible temporal relation types between in this section. We compare our classification to the definition of other related research in section 5.4.1. Two special types of temporal relation are described in section 5.4.2. Finally, we describe the subordinate relation between events in section 5.4.3.

5.4.1 The classification of temporal relations

Our definition of temporal relations is based on TimeML language and Allen's research (Allen, 1983 [1]). The original definition of Allen's temporal relations is the relations between two time-intervals. We define four types of temporal relations between two events -- Rel-linear-preceding, Rel-tree-preceding, Rel-tree-ancestor, and Sub-ord. First three relations correspond to the relations that we described in Section 3.2.2.

- The relation attribute “Rel-linear-preceding” refers to the adjacent event pairs.
- The relation attribute “Rel-tree-ancestor” refers to the head-modifier event pairs.
- The relation attribute “Rel-tree-preceding” refers to the sibling event pairs.

The possible temporal relations are shown in Figure 5-6. EVENT 1 is the focus event and EVENT 2 is the related event. The following examples describe the instances of all temporal relation types (The boldface words correspond to “EVENT 1” in Figure 5-6, and the italicized words are “EVENT 2”):

- AFTER (complementary relation group 1): 股票/指數/上漲/一/點, ***達到***/歷史/新高 (The share index **advanced** one point, and then ***accomplished*** a historical high value.)
- BEFORE (complementary relation group group 1): 股票/指數/***上漲***/一/點, 達到/歷史/新高 (The share index ***advanced*** one point, and then **accomplished** a historical high value.)
- BEGUN_BY (complementary relation group 2): 明年/政府/將/再/恢復 **1**/***撥付 2***/預算 (Government will **restart1** to ***appropriate2*** funds.)
- OVERLAPS (complementary relation group 2): 明年/政府/將/再/***恢復 2***/撥付 **1**/預算 (Government will ***restart2*** to **appropriate1** funds.)
- INCLUDES (complementary relation group 3): 會議/***進行 2***/時/議長/提出 **1**/表決案 (When the conference ***was in progress2***, the chairman **introduced1** a voting bill.)
- DURING (complementary relation group 3): 會議/進行 **1**/時/議長/***提出 2***/表決案 (When the conference **was in progress1**, the chairman ***introduced2*** a voting bill.)
- OVERLAPPED－BY (complementary relation group 4): 他/在/爭奪 **1**/決賽權/時/***落敗 2*** (He ***was eliminated2*** when he **contended1** for passing the preliminary.)
- ENDED_BY (complementary relation group 4): 他/在/***爭奪 2***/決賽權/時/落敗 **1** (He **was eliminated1** when he ***contended2*** for passing the preliminary.)

- SIMULTANEOUS: 該/市/產值/**<u>達到 1</u>**/兩百五十億, ***<u>佔 2</u>***/全省/總產值/的/百分之四十 (The output value of the city **<u>accomplished1</u>** 250 billion, and it ***<u>accounts for2</u>*** 40 percent of the gross output value of the province.)

Intuitively, the some temporal relations have another complementary relation. An obvious example is the complementary relation group between “AFTER” and “BEFORE”. If an event A occurs “BEFORE” an event B, we also can say that the event B occurs “AFTER” the event A. In the foregoing examples, these temporal relation types include four complementary pairs (complementary relation group 1 to 4) and “simultaneous”.

We group the temporal relation “overlapped-by” and “finished” in Allen’s definition into the temporal relation “OVERLAPPED-BY” in our criteria because there are few instances of “overlapped-by” in our experience and these “overlapped-by” and “finished” instances are intuitionally similar to each other. We also group the temporal relation “overlaps” and “start” into the relation “OVERLAP”. The group “AFTER” includes “after” and “met-by” in Allen’s definition. Similarly, the group “BEFORE” includes “before” and ”meet” in Allen’s definition. We group the relations because to distinguish “after” and “met-by” and to distinguish “before” and ”meet” are difficult. However, even though we group the intuitionally similar relations, the temporal relations in our criteria sometimes are intuitively ambiguous in some instances.

For example, in the example “明年/政府/將/再/**恢復**/***撥付***/預算 (Government will **restart** to ***appropriate*** funds.)”, the event pair has the temporal relation “begun_by” and “overlap” (depend on the focus event). The event “恢復 (restart)” is a simultaneous event and changes the state from “not appropriate” to the state “appropriate”. That is, the event “恢復 (restart)” precipitates the event “撥付 (appropriate)” to realize. The start time point of the event “恢復 (restart)” is also the start time point of the event “撥付 (appropriate)”. However, this recognizing process is not exclusive. Other interpretations are possible in different annotator’s intuition. Other annotators could recognize the example as following interpretation: The event “恢復 (restart)” is an administrative proceeding that the government should do before to appropriate the funds. Therefore, the government “cannot” use the fund before the administrative proceeding completed (if they do that, they will be indicted). Therefore the event “恢復 (restart)” must occur “before” the event “撥付 (appropriate)”.

The similar situations may occur in the example “他/在/**爭奪**/決賽權/時/***落敗*** (He ***was eliminated*** when he **contended** for passing the preliminary.)”. We can recognize that the event “爭奪 (contend)” is a competition and it finishes on the time point that scores are proclaimed. Because the winner and loser of the competition are determined by the order of scores, the event “落敗 (eliminate)” will occur with the score proclaimed. The temporal relation between the events “爭奪 (contend)” and “落敗 (eliminate)” is “ended_by” according to this recognizing process. However, another interpretation is that the competition may include several matches and it will finish when the final winner appeared. The player will be eliminated when he loses in a match but the competition does not finished. Therefore the event “落敗 (eliminate)” could be included in the event “爭奪 (contend)”.

We describe the difficulty of annotating the temporal relation between events in above-mentioned examples. These ambiguous instances decrease the consistence of the different annotators’ annotated data. To alleviate the un-consistence needs to consider the pragmatics, the complete context of the article and the world knowledge. For example, for distinguishing the use of verbs in complementary relation group 2, we need to understand what the author wants to explain in the event “恢復 (restart)”, a state change or a specific action. In the complementary relation group 4, we need to consider the type of the preliminary, a league match (such as a baseball game) or the tournament (such as prizefighting). However, to collect all possible situations of these ambiguous is difficult. We discuss the ambiguous instances case-by-case with our annotators.

Except for the grouped relations, other relations are similar as the TimeML and Allen’s definition. The last row in Figure 5-6 shows the non-temporal relations. It includes two un-comparable relations class “first” and “none” and three special types. The special temporal relation types -“ambiguous”, “hypothetical” and “copula-existence” are described in next section. The un-comparable relation class “first” is used for annotating several un-comparable situations. These un-comparable situations are defined following the restriction of the dependency structure. The class “first” means that the focus event does not have a comparable event. It includes following situations:

- The first event of a sentence does not have a preceding adjacent event, therefore the value of the attribute “Rel-linear-preceding” is “first”.

- The root node event in the dependency structure does not have an ancestor event, therefore the value of the attribute “Rel-tree- ancestor” is “first”.
- The single child event does not have brother, therefore the value of the attribute “Rel-tree-preceding” is “first”.

The un-comparable relation class “unknown” means that the temporal relation is unrecognizable. Some event pairs lack the necessary temporal information for recognizing their relation. Annotators annotate these event pairs with the class “none”. Most of the un-comparable instances are the relations between different hypothetical worlds. For example, the relation between event 3 and event 4 in Figure 5-5 is the relation between different hypothetical world (hypothetical world 1 and hypothetical world 2), and then the attribute of this relation is annotated with “unknown”.

Ordering Relation types	Our criterion	TimeML	Allen
EVENT 1 EVENT 2	AFTER	AFTER	after
EVENT 1 EVENT 2		IAFTER	met-by
EVENT 1 EVENT 2	OVERLAPPED－BY		overlapped-by
EVENT 1 EVENT 2		ENDS	finishes
EVENT 1 EVENT 2	DURING	DURING/IS_INCLUDED	during
EVENT 1 EVENT 2	BEGUN_BY	BEGUN_BY	started-by
EVENT 1 EVENT 2	SIMULTANEOUS	SIMULTANEOUS/IDENTITY	equal
EVENT 1 EVENT 2	INCLUDES	INCLUDES/DURING_INV	contains
EVENT 1 EVENT 2	ENDED_BY	ENDED_BY	finished-by
EVENT 1 EVENT 2	OVERLAPS		overlaps
EVENT 1 EVENT 2		BEGINS	starts
EVENT 1 EVENT 2	BEFORE	IBEFORE	meets
EVENT 1 EVENT 2		BEFORE	before
Non-temporal relation	first, ambiguous, end, copula-existence, hypothetical, unknown		

Figure 5-6: Relation definitions among our critera, TimeML and Allen's work.

5.4.2 The special types of temporal relation

The special types of temporal relation include three relation types-"ambiguous", "hypothetical" and "copula-existence". The relation "hypothetical" is annotated in the actual-hypothetical event pairs. For example, the event "成立 (establish)" and "吸引

(attract)” in example (b) (section 5.2.2) and the event “要求 (require)” and “改進 (amend)” in example (i) are the actual-hypothetical event pairs. The temporal relation type[31] is “hypothetical”.

The special relation type “copula-existence” is annotated in copula (existence)-phrase event pairs. The example (k) and (q) (section 5.2.2) are copula (existence)-phrase event pairs. The value of the attribute “Rel-tree- ancestor” in the event “修定 (to legislate)”(in example (k)) and in the event “完成 (to complete)” (in example (q)) is both “copula-existence”. Following our discussion in section 5.2.2, the event that is accentuated is the descendent of the copula (existence) verb. The dependency relation between the accentuated event and the copula (existence) verb is a head-modifier relation. Therefore the special temporal relation type “copula-existence” is limited to annotate the head-modifier relation attribute “Rel-tree- ancestor”. If a copula (existence) event pair is both an adjacent event pair and a head-modifier event pair in the dependency structure, annotators annotate the attribute “Rel-tree- ancestor” as the value “copula-exsitence” and annotate the attribute “Rel-linear-preceding” is the empty class “none”. This different annotating process follows our proposed definition of the three kind of temporal relations (see section 3.2 and 5.3.1). Following the definition of temporal relations, the event that is accentuated by a copula verb is a hypothetical world event and the copula verb is the actual world event. The events in hypothetical world cannot be ordered in the timeline in the actual world. We do not consider a temporal relation of an actual-hypothetical event pair in the adjacent view point.

The special relation type “ambiguous” is defined for dealing with some ambiguous temporal relation instances. An example is a news article that reports a historical conference which focuses on the past effective ruler “張學良 (Chang Hsüeh-liang)”[32]. This article therefore includes the events that is in connection with the action in the conference, and includes the events that “張學良 (Chang Hsüeh-liang)” had done. Both these events are in the actual world. That is, the events occurred in the past therefore they are all “truth”. However, the article lacks for the temporal information of Chang

---

[31] In these cases, we focus on the events “吸引 (attract)” and “改進 (amend)”.

[32] “張學良 (Chang Hsüeh-liang)” was the effective ruler of Manchuria and much of North China in 1928-1931.

Hsüeh-liang's actions. To order Chang Hsüeh-liang's actions is difficult and the temporal relation between a Chang Hsüeh-liang's action and a conference event is ambiguous. Again, some sentences in this article could include both the Chang Hsüeh-liang's actions and the conference events. For example:

- 他/在/今天/**舉行**/的/學會/中/**列舉**/張學良/在/東北/**興建**/鐵路/,/**開通**/航線/,/**成立**/學校/的/貢獻 (In the conference that **was held** today, he **listed** the contribution that Chang Hsüeh-liang **built** the railway, **opened** a new airline and **established** schools in Manchuria.)

This sentence includes two conference events: "舉行 (to hold)" and "列舉 (to list)", and includes three Chang Hsüeh-liang's contributions: "興建 (to build)", "開通 (to open)" and "成立 (to establish)". The temporal relation between the conference events is clear: the event "舉行 (to hold)" includes the event "列舉 (to list)", or the event "列舉 (to list)" occurs during the event"舉行 (to hold)". However, we cannot order three Chang Hsüeh-liang's contributions correctly because there is no more additional information in the sentence that can redounds to order the events[33]. Therefore the temporal relations between these events are ambiguous. We cannot predicate the relation between the event "興建 (to build)" and the event "成立 (to establish)".

Even though the temporal relations between Chang Hsüeh-liang's contributions are ambiguous, we are sure that the event "成立 (to establish)" certainly occurs "before" the event "舉行 (to hold)" because Chang Hsüeh-liang was a past ruler. In the annotating process, if the dependency relation between the event "成立 (to establish)" and "舉行 (to hold)" cannot be connected directly by the three types of our definition (Rel-linear-preceding, Rel-tree- ancestor and Rel-tree- preceding), we can try to use the inference rules to induce the temporal relation between this event pair. However, the ambiguous relation cannot be added to the inference rules then we cannot induce this temporal relation between "成立 (to establish)" and "舉行 (to hold)". Figure 5-7 illustrates this problem. The temporal relation between the event "成立 (to establish)"

[33] We cannot regard that the presenter listed Chang Hsüeh-liang's contributions in temporal order. Therefore, according the description in the sentence, the temporal relations between ambiguous events can be any possible temporal relation in Figure 5-6.

and ”舉行 (to hold)” can be induced by the path: “舉行 (to hold)”→“列舉 (to list)”→“成立 (to establish)”. However, ““列舉 (to list)”→“成立 (to establish)”” is ambiguous, therefore we cannot induce the temporal relation the path: “舉行 (to hold)”→“列舉 (to list)”→“成立 (to establish)”.

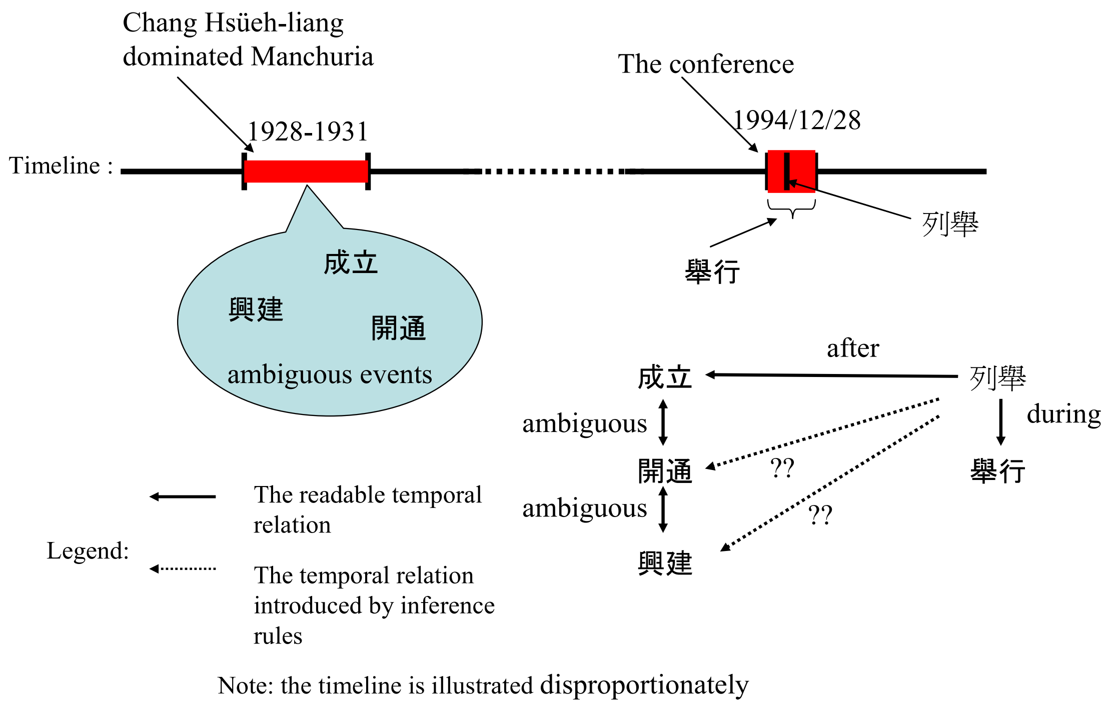


Figure 5-7: An example of the ambiguous temporal relations

For dealing with this kind of problem, we give a special definition to the class type “ambiguous” as following distribution: “the temporal relation between the focus event pair is uncertain, but in the inference rules, these ambiguous relations are regarded as a simultaneous relation”. That is, in the induced path: “舉行 (to hold)”→“列舉 (to list)”→“成立 (to establish)”, the temporal relation ““列舉 (to list)”→“成立 (to establish)”” is regarded as “simultaneous”. Therefore, we can induce the temporal relation between “舉行 (to hold)”→“成立 (to establish)” as the class “after”. Regarding the ambiguous relation as the type “ambiguous” or “simultaneous” depends on the context and world knowledge. We regard the example of the ambiguous relation as the type “simultaneous” according to the truth that Chang Hsüeh-liang dominated Manchuria in 1928-1931 and the conference was held on 1994. The length of this time interval is a useful clue for deciding the meaning of the ambiguous relations. We cannot model this deciding process; therefore we only discuss each instance to deal with this problem.

5.4.3 The sub-ordinate relation between events

The last relation attribute of an event (Table 5-1)-“Sub-ord” means the subordinate relation of a head-modifier event pair. We refer to TimeML in defining the subordinate relations. Annotators refer to the dependency structure of the focus event to recognize the subordinate event and its head event. Annotators annotate sub-ordinate relation to the event that has its head event in the dependency tree. The definition of the subordinate relations is described in Table 5-6. TimeML includes another link tag “ALINK” to annotate aspectual relations. We do not distinguish SLINK and ALINK and designate these two kinds of relations as the tag “Sub-ord”. Because the temporal relations include aspectual relations (such as BEGUN_BY and END_BY); the annotators can annotate the temporal relation between a sub-ordinate event and its head event to cover the SLINK and ALINK.

Our special temporal relation class “hypothetical” can be replaced instead of the SLINK class “modal” in TimeML (see section 2.2.1). Some sub-ordinate classes can be recognized by referring to the event class of verbs. For example, the class “possibility” can be annotated in the events that their verb use correspond to the example (i)-(l) in section 5.3.2. In the sub-ordinate class “report”, we define it to express the event that is described by another report event. The report events (for example, the verb “報導 (to report)” and “宣佈 (to announce)”) are the verbs that the subject describes something that they will do or have done. Because our original data is collected from news articles, the articles usually include the announcements and the reports of some people or some organization. These announcements could occur in long time before the publish time, or could be hypothetical event[34]. To annotate these report events can help to annotate the temporal relations of the sub-ordinate events.

---

[34] For example, an article describes that the tub-thumper announced that the government will build a bridge. We can identify the time point that the tub-thumper told about the detail of the bridge. If the context of the

The examples of all subordinate class are described as following sentences (the italicized words is the focus event in annotating work):

- Introduce: 明年/政府/將/再/**恢復**/***撥付***/預算 (Government will **restart** to ***appropriate*** funds.)
- Explanation:國會/**修訂**/三年前/***頒布***/的/舊/法律/ (The congress **legislated** the old version of bill that ***was enacted*** at three years ago.)
- Condition: 工業區/***成立***/後/大量/**吸引**/外資 (after the industrial estate ***was established***, it **attracted** a great deal of foreign capital)
- Report: 他/在/今天/**舉行**/的/學會/中/***列舉***/張學良/在/東北/**興建**/鐵路/,/**開通**/航線/,/**成立**/學校/的/貢獻 (In the conference that **was held** today, he ***listed*** the contribution that Chang Hsüeh-liang **built** the railway, **opened** a new airline and **established** schools in Manchuria.)
- Passive: 兵馬俑/**被**/***稱爲***/世界/第八奇跡 (Mausoleum of the First Qin Emperor ***is called*** "The 8th Wonder of the Ancient World").
- Possibility: 他們/**希望**/政府/***訂立***/相關/法案 (They **wish** the government to ***legislate*** against affiliated bill)
- Hypothetical: 此/技術/**有助於**/未來/***開發***/新藥 (this technology can **help** to ***develop*** a new kind of medicine)

---

details does not include enough information, we can regard the details as the hypothetical events. Actually, many report events have this kind of situation in our experience.

subordinate class	definition
introduce	The focus event is introduced by the head event.
explanation	The focus event explains the head event.
condition	The focus event occurs if the head event is true.
report	The head event is a "report" event.
passive	The focus event is passive of the head event.
possibility	The head event describes a possibility of focus event.
hypothetical	The focus event is a bridge event of a hypothetical world

Table 5-6: Definition of subordinate class

5.5 The corpus observation

We are annotating our temporal relation tagged corpus. The original data is Penn Chinese Treebank. For each verb in the corpus, we annotate the temporal properties of and the temporal relation tag of events by using the morphological information and dependency structure. We show an example in section 5.5.1 to illustrate the work that our annotators are doing. In section 5.5.2, we report the progress of our annotating work and show the distribution of each attribute.

5.5.1 An example of annotating an event

Figure 5-8 describes the attributes of the events in the example sentence of Figure 3-3. These attributes of the main verb of each event are annotated automatically. The annotator refers to this information to annotate the temporal properties and the temporal relation tag of events.

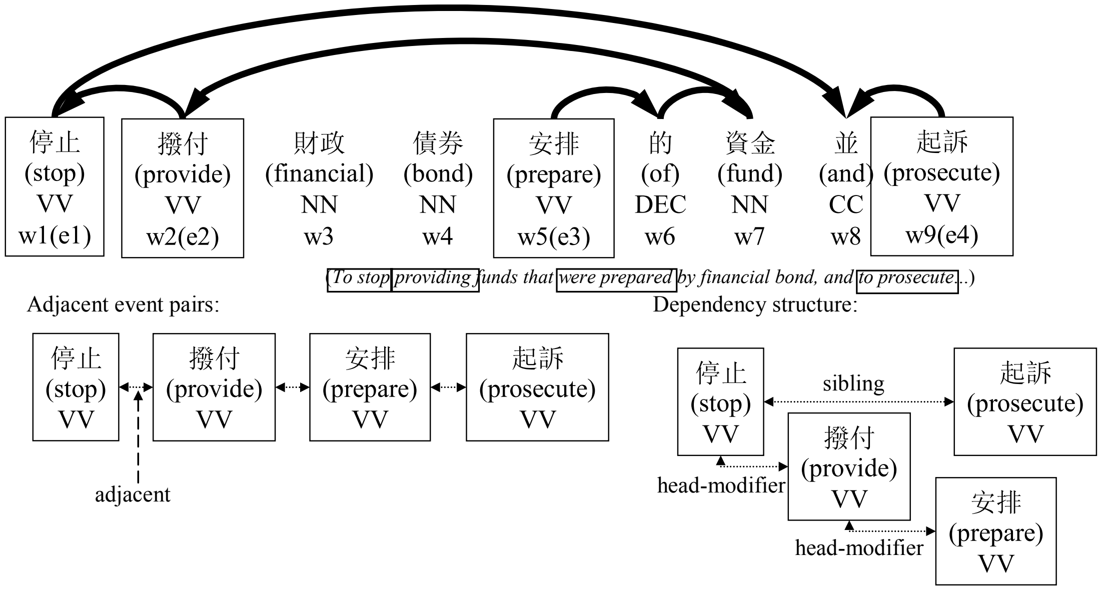
First, the annotator decides an appropriate value of the attribute “E-class”. It is important for annotating the temporal relation of an event. Annotators should know whether the focus event is in an actual world or in a hypothetical world before annotating the attributes “Rel-linear-preceding” and “Rel-tree-ancestor”. If the focus event has a different “E-class” value against its head event or its adjacent event, the two focus events are in different worlds (one event is in actual world and another event is in hypothetical world). The temporal relation between these focus event pair should be “hypothetical” or “none”. According to the description in the section 5.3.1, if a local tree includes a hypothetical world event as its root node, all other events in the local tree are regarded as hypothetical events. Therefore, to annotate the attribute “E-class” of the descendant events in such a local tree, referring the “E-class” value of the root node is necessary. In annotating the attribute “Rel-tree-preceding”, the temporal property “E-class” is not important information, because in most cases, the temporal relation between two sibling events is in the same world. Other temporal properties of an event are annotated independently. That is, annotating the values of attributes “E-dynamic”, “E-period” and “E-telicity” does not need the information of other attributes.

After annotating the temporal properties of the focus event, annotators annotate the temporal relation and sub-ordinate attributes of the focus event. Annotators refer the dependency structure and the “E-class” of the focus event to do this to annotate these attributes. All these attributes are annotated independently, that is, annotating each relation attributes do not need to consider other attributes[35].

Considering the dependency structure in this example (Figure 5-8), the events “停止 (to stop)” and “起诉 (to prosecute)” are coordinate events. The annotator regards that the suitable value of the attribute “Rel-tree-preceding” of the event “起诉 (to prosecute)” is “SIMULATANEOUS[36]”. The attributes “Rel-tree-preceding” of the other events are “first” because each of these events is an only child. The ancestor event of the events “拨付 (to provide)” and “安排 (to prepare)” are the same as their linear adjacent events, therefore the value of the tag “Rel-linear-preceding” is the same as the tag “Rel-tree-ancestor”. Because the event “停止 (to stop)” is the first event and is the root event of the dependency tree, it has neither its linear adjacent events nor its ancestor event. Therefore the values of the attributes “Rel-linear-preceding” and “Rel-tree-ancestor” are “first”. In annotating the sub-ordinate relation attribute “sub-ord”, the event “安排 (to

[35] The attribute “sub-ord” can only be annotated on the events that have their head event. This is a natural constraint following the dependency structure.

[36] Actually, the conjunction word “並 (and)” in Chinese usually means that the temporal relation between this coordinate events is “SIMULATANEOUS”.



event / Attribute	停止 (stop)	撥付 (provide)	安排 (prepare)	起訴 (prosecute)
information of the main verb				
ancestor-verb	end	w1	w2	end
Eventid	e1	e2	e3	e4
Maindep	w8	w1	w6	w8
Mainid	w1	w2	w5	w9
mainpos	VV	VV	VV	VV
mainword	停止	撥付	安排	起訴
the temporal properties of the event				
E-class	actual world	actual world	actual world	actual world
E-dynamic	dynamic	dynamic	dynamic	dynamic
E-period	instantaneous	durative	instantaneous	instantaneous
E-telicity	telic	telic	telic	telic
the temporal relation tag of the event				
Rel-linear-preceding	first	END_BY	BEFORE	AFTER
Rel-tree-preceding	first	first	first	SIMULTANEOUS
Rel-tree-ancestor	first	END_BY	BEFORE	first
Sub-ord	none	explanation	explanation	none

Figure 5-8: The attributes of the events in Figure 3-3

prepare)” is a sub-ordinate event of its head event “撥付 (to provide)”. And the event “撥付 (to provide)” is the subordinate event of the event “停止 (to stop)”. The annotator annotates the sub-ordinate relation attribute of these two events and do not annotate to other events (because they do not have any head event).

This procedure of temporal relation annotating does not need to consider all combinations of events therefore it can reduce manual efforts. In our annotating work, the average work time for annotating an article is twenty-five minutes. This is shorter than the work time of TimeML annotating (several hours per article).

5.5.2 The corpus distribution

In this section, we report the distribution of the corpus annotation. We annotated a part of Penn Chinese Treebank and investigated the distribution of each attribute. The Penn Chinese Treebank 5.0 contains 507,222 tokens, 18,782 sentences, and 890 articles. We will automatically analyze these attributes in the future. But we need a manually tagged training data to construct machine learning models (see section 6.2). We use a part of the Penn Chinese Treebank (ten percent of the corpus) to construct a basic data set. Because the consistency of the annotated corpus is not competent, we could not use it to get machine learning models before we repeat the annotating work to improve the consistency. This is time consuming but we can deduce the working time after our basic annotating system completed. The distribution of the attributes in our annotated corpus is summarized in Table 5-7. Because the distribution of temporal relations is uneven and some values of the temporal relation are data sparse, we only show the top five types of temporal relation attributes in the table.

Considering the tag “Rel-linear-preceding (adjacent event pairs)”, the relation classes “after / unknown / simultaneous / before” are the most possible relations for the adjacent event pairs. Because annotating the adjacent event pairs does not consider the actual-hypothetical worlds, an adjacent event pair is able to in different worlds. The non-relation type “unknown” has the second instances. Because we request the annotators to annotate as many temporal relations as possible, they used much world knowledge and contextual information in reading the articles to select a most suitable value of the temporal relations. Certainly, the most clear temporal relation is intuitively the simple relations: "after", "before" and "simultaneous". Our annotators tend to annotate the temporal relations using these simple relations. Therefore, the class "ambiguous" in tag "Rel-linear-preceding" is infrequent.

The relation class "first" of the tag "Rel-tree-preceding (sibling event pairs)" means the focus event does not have any sibling event because events in similar sentences are structured as a hierarchy structure. There are few sentences that have some events that modify the same event. Therefore, most events are singletons of their head events.

In the tag "Rel-tree-ancestor (head-modifier event pairs)," the root event of the dependency structure does not have a head event and the correct selection of the tag "Rel-tree-ancestor" in this case should be "first". The attribute type "hypothetical" means the focus event pair is a bridge between actual and hypothetical world therefore it has many instances in the corpus (it is the third type in this attribute).

Attribute						
E-class	value	actual world	hypothetical world			
	Number	4584	2936			
E-dynamic	value	State	dynamic			
	Number	2173	5347			
E-period	value	durative	instantaneous	repeat		
	Number	3305	3998	217		
E-telicity	value	Telic	non-telic	continue-state		
	Number	3721	3401	398		
Rel-linear-preceding (top five relations)	value	after	unknown	simultaneous	before	during
	Number	2423	2119	1393	1266	250
Rel-tree-preceding (top five relations)	value	first	after	before	simultaneous	during
	Number	4594	1015	656	577	257
Rel-tree-ancestor (top five relations)	value	first	before	hypothetical	simultaneous	After
	Number	1965	1850	1547	1013	581
Sub-ord (top five relations)	value	none	hypothetical	explanations	introduce	report
	Number	3245	1561	1523	556	432

Table 5-7: Distribution of the attributes

In the tag “sub-ord (subordinate relation),” the value of the most meaningful classes of subordinate relation are “explanations” and most of this attribute is the class “none”.

5.6 Difference between our guideline and TimeML

Our corpus guideline adopts many concepts and attribute values token from TimeML. However, our corpus criteria have several differential features that are different from TimeML. First, the goal of our research is to construct a machine learning based annotation system. All attributes can be annotated automatically after we complete a large corpus and train a machine learner. In TimeML, annotators need to extract the temporal entities and temporal relations by their knowledge, this work is time-consume. However, in our criteria, our annotators focus on the attributes of events and inference rules are used to extend temporal relations. We can create a large corpus according to our criteria with less manual efforts, which will be more difficult by using TimeML. Second, for recognizing the events of corpus automatically, we limit the events to the verbs, but the TimeML includes more syntactic event constituents. To limit the event only to verbs also reduce manual efforts and preserve the major part of all events.

When we use the temporal relation tagged corpus to train a machine learner, every attribute of our criteria can be trained. However, training the machine learner with a corpus tagged by the TimeML annotation scheme is more difficult than with our corpus. TimeML includes more difficult criteria. For example, the machine learner should identify the event phrase[37] (clause) in corpus. Following the introduction in section 2.3, participates of the shared task use TimeBank as the corpus to identify the temporal relations in articles. However, they are not required to identify event phrases because to identify event phrases needs chunking technology and world knowledge.

Finally, our criteria are based on dependency structure. TimeML does not consider any syntactic nor morphological information for their annotation. Our criteria can describe the

[37] The events in TimeML could be a noun phrase or verb phrase.

temporal relations of subordinate and coordinate event pairs clearly. In our experience, these criteria can provide annotators with useful information that help them to recognize the relations between events. Moreover, because verbs in Chinese do not have morphological change according to tense, to recognize the tense of an event needs the information of modifier of the verb, such as a temporal expression “昨天”(yesterday) or a temporal function word “已經”(have). This information directs to the temporal relations analysis for Chinese. This information can be provided by dependency structure whereas TimeML does not emphasize this. Therefore, our criteria are more applicable than TimeML to creation of a temporal relation tagged corpus of Chinese.

5.7 Summary

Our proposed method is constructing a machine learning based temporal relation analyzer for Chinese. In this chapter, we describe the guideline of our temporal relation tagged corpus. Figure 5-9 illustrates the flow of annotating the temporal relation annotated corpus. The section number in each block refers to its corresponding section. The square “automatic annotating” means that the processes are annotated automatically. The square “manual annotating” means the processes that the annotators do.

In section 5.1, we introduce the environment of our annotating work. We used the Penn Chinese Treebank (Palmer, et al., 2005 [65]) as the original data and we transformed phrase structures into dependency structures using head rules (Cheng, 2005 [14]). We annotate the temporal attributes and the temporal relations of events on a part of the Penn Chinese Treebank. Our corpus contains 151 Chinese news articles with 7239 events, 1945 sentences and 49691 tokens.

In section 5.2, we describe the attributes in our corpus. We annotate the two types of the temporal attributes of events: the properties (event class, dynamic, period and telicity) and the temporal relations for limited event pairs (adjacent event pairs, head-modifier event pairs, sibling event pairs and subordinate relations). Some information of words and events can be annotated automatically, such as the POS-tag, head word, the path to the root of the sentence, and so on. The annotators decide the most appropriate attributes of the temporal relations and temporal properties of each event. We introduce the attributes of morphological-syntactic information in section 5.2.1-section 5.2.3 and the attributes of an event in section 5.2.4.

In section 5.3, we describe the verb-event classification. Our research focuses on the relations between events and limits the events to the verbs. Verbs can be identified according to the POS tag of the word automatically (the POS-tag: VV, VA, VC and VE). All these types of verbs are EVENTs and have the annotatable attribute in our criteria. Verbs in an article include the events in the actual world (which describe actual situations or actions) and the events in hypothetical worlds (which describe possible situations, imagination or the background knowledge). We attempt to give the definition of actual / hypothetical world events (verbs). We investigate the usages of verbs in Penn Chinese treebank and classify the verbs to actual / hypothetical world.

In section 5.4, we introduce the possible temporal relation types between two events. In section 3.2 and section 5.2.4, we describe that our proposal method is that constructing a temporal relation analysis system. This system analyzes the temporal relation between events (verbs). The attributes of an event include these temporal relations that are based on a viewpoint of the dependency structure. We describe the possible temporal relation types between in this section. We compare our classification to the definition of other related research in section 5.4.1. Two special types of temporal relation are described in section 5.4.2. Finally, we describe the subordinate relation between events in section 5.4.3.

In section 5.5, we show an example to illustrate the annotating work and then we report the distribution of each attribute in our annotated corpus.. Finally, we compare our criteria and TimeML in section 5.6.

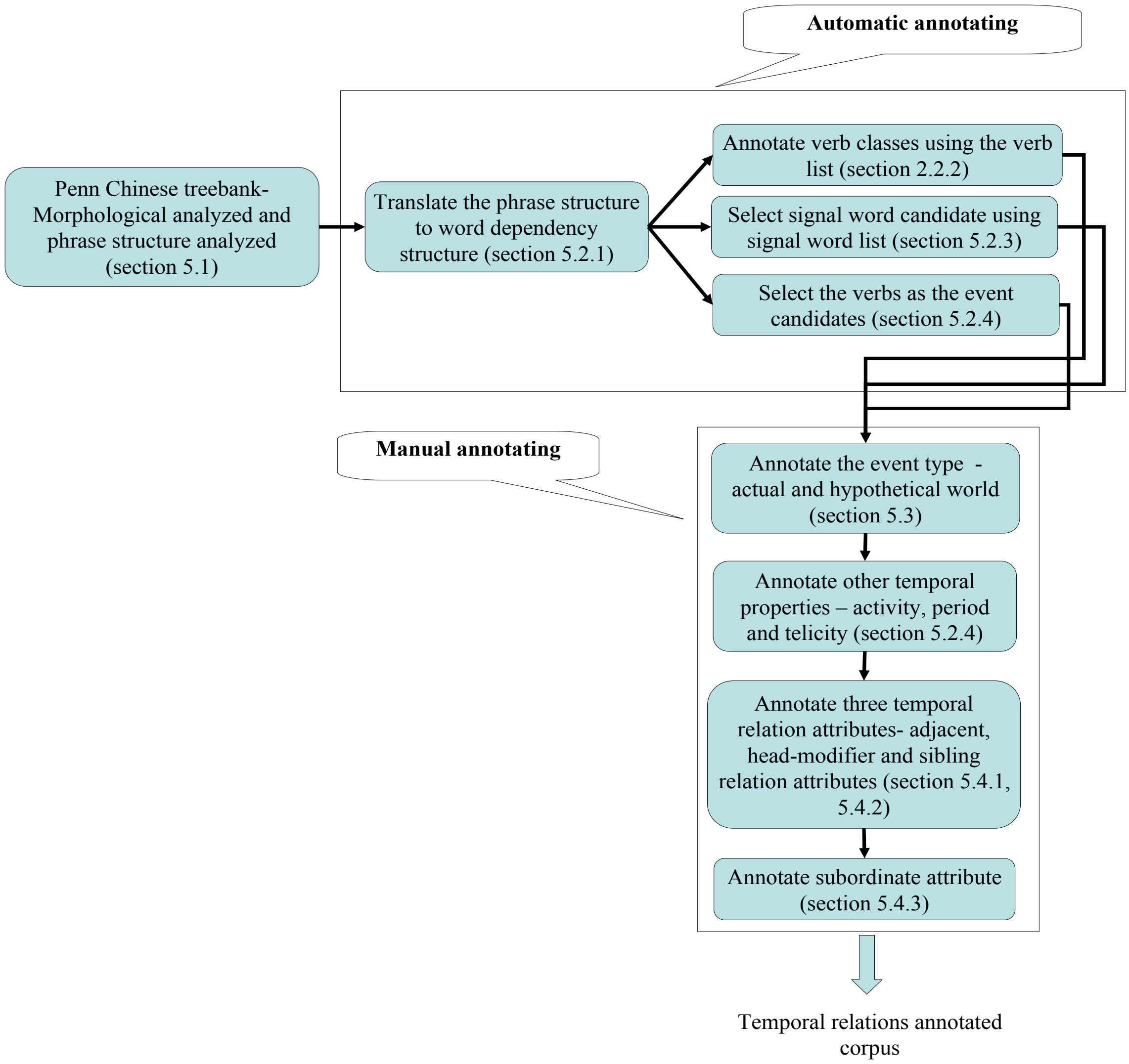


Figure 5-9: The flow of annotating a temporal relation annotated corpus

Chapter 6 Temporal Relation Annotating Using Machine Learning

In this chapter, we apply the temporal relation annotated corpus to train the automatic temporal relation annotator and evaluate the performance and coverage of our proposed system. First, we perform several preliminary experiments before we evaluate the performance of our system. These preliminary experiments can provide more useful features for machine learning experiments. Second, we evaluate the performance of our temporal relation annotating system. This evaluation includes the automatic annotation of the temporal properties and the temporal relation attributes of events. Finally, we examine the recall of our proposed system. We compare the result of our proposed system and a complete manual work to observe that how many temporal relations can be identified by our system.

6.1 Preliminary experiments

In this section, we investigate four preliminary experiments. First, we evaluate the consistency of our manually annotated temporal relation corpus. Second, we investigate the coverage of our proposed annotating criteria. We do not annotate the temporal relation of all possible event pairs. We limit annotation work on adjacent, sibling and head-modifier event pairs. We evaluate the coverage of our criteria with all possible event pairs. Third, we perform the experiment of a machine learning based SIGNAL word identification. We describe the feature selection and the results. Final preliminary experiment is simple temporal expression recognitions. We use a simple rule set to recognize the meaning of numerical expressions.

6.1.1 The manually annotated data

We need a small corpus consistency to observe the performance of our proposed criteria and system by comparing the annotation results by our proposed method and annotating all possible event pairs. We cannot compare two methods in a large corpus because annotating all event pairs manually is time-consuming. Therefore, we select 50 articles in Penn Chinese Treebank and only use the first two paragraphs of each article then annotating them by our criteria. The small corpus includes 732 events (verbs) and 5010 tokens. The distribution of attributes is shown in Table 6-1. This testing is used in all our experiments in this chapter.

The column "consistency" in the table is the consistency of our annotating work between two annotators. Our annotators annotate this testing corpus independently, then, compare the results to estimate the consistency. We train our annotators simultaneously; therefore they have hearly same annotating knowledge and technique. In the results, they have high consistency in the attributes "E-class" and "Rel-tree-preceding" because we give more clear definition of the "E-class" types and the attribute "Rel-tree-preceding" has syntactic restrictions. The sibling relations (the attribute "Rel-tree-preceding") do not exist in the single child of a head event. Annotators can identify these single children according the dependency structure. Therefore they can achieve high consistency in this attribute.

The consistencies of other attributes are lower than 90%. The lowest consistencies are the attributes "Rel-liner-preceding" and "E-telic" (82% and 81%). This result indicates the difficulty of the annotating work. Even though our annotators are taught same criteria, the annotating results are not completely consisting. To discuss the annotating results repetitively is important for the annotating work. This is time-consuming; even so, our criteria can reduce manual efforts.

Attribute	Consistency						
E-class	0.90	value	actual world	hypothetical world			
		Number	493	239			
E-dynamic	0.87	value	State	dynamic			
		Number	152	580			
E-period	0.84	value	durative	instantaneous	repeat		
		Number	303	414	15		
E-telicity	0.81	value	telic	non-telic	continue-state		
		Number	384	239	109		
Rel-linear-preceding	0.82	value	after	unknown	before	simultaneous	during
		Number	278	216	139	75	18
Rel-tree-preceding (top five relations)	0.95	value	first	after	before	simultaneous	during
		Number	507	119	74	20	12
Rel-tree-ancestor (top five relations)	0.86	value	before	first	hypothetical	after	simultaneous
		Number	189	180	163	81	63
Sub-ord (top five relations)	0.87	value	none	hypothetical	explanations	report	introduce
		Number	301	134	123	65	40

Table 6-1: The attribute distribution of the testing data

6.1.2 The temporal relation coverage of the proposed criteria

A question for our research is "how many temporal relations could be identified in the proposed method?", that is, the coverage of using our proposed method. For investigating the relation coverage of our proposed criteria, we annotate the testing corpus manually both by annotating all event pairs and by using our criteria. After annotating by our criteria, we use the inference rules shown in Table 3-1 to extend the relations. The coverage in this section can be regarded to the limit of our automatic annotating system.

For observing the coverage of different methods, we survey four methods to extract temporal relations. They are:

- Using the relations of the adjacent event pairs (RLP is an abbreviation of Rel-linear-preceding), the head-modifier event pairs (RTA is an abbreviation of Rel-tree-ancestor) and the sibling event pairs (RTP is an abbreviation of Rel-tree-preceding), then extending the relations by the inference rules (The column "RLP+RTA+RTP" in Table 6-2).
- Only using the relations of the adjacent event pairs with the inference rules (The column "RLP" in Table 6-2).
- Using the relations of the head-modifier event pairs and the sibling event pairs with the inference rules (The column "RTA+RTP" in Table 6-2).
- Using three kinds of event pairs without the inference rules (The column "RLP+RTA+RTP w/o inference rules" in Table 6-2).

Because the attribute distribution of the testing data is uneven, we reduce the nine main classes of temporal relations to five classes for experimental convenience. The classes {AFTER, OVERLAP_BY, BEGUN_BY} are reduced to the class "AFTER" and the classes {BEFORE, OVERLAP, ENDED_BY} are reduced to the class "BEFORE." According to our annotator's experience, these classes are ambiguous in many event pairs and they are few in the testing data; therefore we group the classes to reduce the ambiguity.

Table 6-2 describes the coverage of our proposed criteria. We regard the understandable relations of all event pairs as the gold standard (the row "True event pairs") and we compare the result of our method with the gold standard. The row "Recall" shows the coverage of each method.

The last column shows the case of using our criteria to annotate temporal relations without using the inference rules. The row "Extend event pair relations using the inference rules" in this column indicates the total amount of events that are annotated by our criteria. It should be noted that an adjacent event pair could be also a sibling event pair or a head-modifier event pair. These event pairs will be calculated twice in the two types of event pairs. Therefore the number of the relations that we extract by our criteria is not equal to the total number of the three kinds of relation types (RLP+RTA+RTP > Total event pairs).

Intuitively, the combination of events must include all relations that could be extracted. The relations that we extract by our criteria must be included in the gold standard. In Table 6-2, the row “Total extracted event relations” is included in the true event relation. However, in our preliminary investigation (this result is NOT included in Table 6-2); the annotator does not consider any syntactic structure or full context in annotating the event pairs and then the extracted event relations are not completely consistent to the true event relations. Because this testing data set was annotated by an annotator but not completed in one day, the annotator does not remember the viewpoint before when he annotates the same instance. The annotator annotated the event combination first and then annotated the three types of event pairs of our criteria. The intuitive reorganization of event relations could be inconsistent with the dependency structure. Therefore we re-annotated the testing data several times to comfirm the consistency of the relation attributes. This observation indicates the difficulty of constructing a corpus consistently.

According to our results, the recall of using the dependency viewpoint-“RTA+RTP” with the inference rules is better than the recall of only using the adjacent viewpoint-“RLP” with the inference rules. The hypothesis in Chapter 3 is confirmed in the result. The head-modifier event pairs and sibling event pairs can connect some fragment structures and can extract many important relations that the adjacent event pairs cannot extract. We use three types of event pairs and the inference rules and acquire 63% relations of the gold standard can be extracted. One reason is that we only consider the absolute inference rules. The inference rules in Table 3-1 include several empty blocks. We can add more inference rules that consider other syntactic or semantic information of events to extend the relations.

Another reason of the low coverage of using our criteria is that the relation increasing by inference rules is worse in long distance. Because we annotate all possible event pairs to be the golden data, it includes long distance relations (for example, the relation between the first event and the last event of the article). Our proposed method emphasizes the temporal relations in local sentences. We use the adjacent relation and the sibling relation to connect sentences, and then use inference rules to extend long distance relations. If the event pair does not have an appropriate inference rule to deduce more relation, the deduction chain will be segmanted. Therefore, using our criteria cannot achieve higher coverage. Someone may think that this result and observation are the disability of our proposed method. However, the necessity of long distance relation is arguable. Intuitively, temporal relations of long distance event pairs are relatively difficult to identify. Our annotators annotate the long distance event pairs as far as possible, but this is not necessary in normal NLP application. Our proposed method can extract most of the local temporal relations and which are more useful than long distance relations.

	RLP+RTA+RTP	RLP	RTA+RTP	RLP+RTA+RTP w/o inference rules
Relations of Adjacent event pair (The attribute Rel-linear-preceding-RLP)	702	702	0	702
Relations of Head-modifier event pair (The attribute Rel-tree-ancestor-RTA)	530	0	530	530
Relations of Sibling event pair (The attribute Rel-tree-preceding-RTP)	205	0	205	205
Total extracted event relations	1018	702	735	1018
Extend event relations by using inference rules	4166	2005	2871	1018
True event relations	6646	6646	6646	6646
Recall	0.63	0.30	0.43	0.15

Table 6-2: Results of the relation coverage

6.1.3 Tagging the word attribute "signal"

The SIGNAL words are useful clues for temporal relation analysis. We select the SIGNAL word candidates and define the intuitive meaning of them. For applying it in temporal relation automatic annotation, we need to prepare a machine learning based SIGNAL word identifier before constructing our system. We recognize the SIGNAL candidates in the training and testing data, and then train the machine learner with the annotated data.

Table 6-3 describes the distribution of SIGNAL words in training / testing data and the experiment results of the automatic identifying result. We use SVM as the machine learner and select the morphological and dependency information as features for SVM. The morphological information includes the POS-tag and the word of the tokens that are in the window[38]. The dependency information is the attributes of dependency structure of words in section 5.2.1. We compare the results of using the dependency structure and only using the morphological information of words in Table 6-3. The results show that using dependency structure for SIGNAL word identification is necessary and the accuracy of automatic identification is 91%. We use the output of this experiment as a feature for the automatically temporal relation annotating.

	Training data	Testing data
Temporal signal words	908	112
Non-temporal words	1443	132
Total candidates	2351	244

	Morphological information	Morphological + dependency information
accuracy	0.84	0.91

Table 6-3: The distribution of SIGNAL words and the experiment results

6.1.4 A simple temporal expression recognizer

Following the discussion in Chapter 1 and Chapter 2, identifying temporal expressions is a part of temporal information processing. The temporal expressions are fewer than the events in articles and then we do not identify the temporal relations by dealing with temporal expressions. Because the temporal expressions are important information, we wish to apply the information of temporal expressions as a machine learning feature for temporal relation annotating. However, to recognize the variable temporal expression is difficult. We only apply most simple numerical expressions and some special expression to our system. The distribution of simple numerical expressions in training and testing data is

[38] The "window" in this experiment is the tokens that around the focus SIGNAL candidate. We select the window size as the preceding two tokens and the succeeding two tokens of the focus candidate, and the candidate itself.

shown in Table 6-4. For applying these simple temporal expressions, we deal with the temporal expressions as following process to recognize the meaning of the expressions[39].

- Numerical expressions with temporal units: This kind of expressions has two part—"digits" and "temporal unit suffix", ex. 1999 年. The temporal units include "年 (year), 月 (month), 日 (day), 時 (hour), 點 (hour), 分 (minute), 秒 (second)". We recognize the digits as the numbers of the units.
- Numerical expressions without temporal units: this kind of expressions only includes digits, ex. 1974. Because they usually represent the years, we define a simple rule to recognize the meaning: if the digit is large than 1000, the digit mean A.D. years. If the digits is small than 1900 and large than 30, it means the year—19XX A.D. if the digit is small than 30, it means the date in the month.
- Special words: ex. 現在, 當今, 目前; these special words are usually used in news articles and all mean "now". We set the value of these words as the published time of the article.

This definition applies to a handful of numerical expressions. In the testing data, thirty-eight expressions can be recognized in this method. To refer to the related researches to extend the rules for recognizing the meaning of more numerical expressions is an important future direction of our research.

Because the number of time cannot be used as a feature in SVM, we define two features of these simple numerical expressions for the machine learner. First, we compare the expression and the published time to estimate the relation is "before", "after" or "simultaneously". Another feature is comparing the focus expression and the one in the descendant words of the focus expression. The value also includes "before", "after" and "simultaneously". If more than one expressions in the descendant words, we select the most close expression to estimate.

[39] We refer to Li's research (Li et al., 2005 [50]) to deal with the numerical expressions.

	Training data	Testing data
TMP phrases	1037	130
NP-TMP	501	54
Simple numerical expressions	228	38

Table 6-4: The distribution of simple numerical expressions in training and testing data

6.2 Automatic temporal relation identification

After we manually annotate the temporal relation tagged corpus with our proposed criteria, we use SVM (support vector machines) as machine learner to compose a temporal relation identifier. The training data is our annotated corpus in section 5.5.2 and the testing data is another small annotated corpus in section 6.1.1. We perform experiments to investigate the accuracy of automatic annotating the attributes which include temporal properties of events and temporal relations between event pairs. We also investigate the efficient of using preliminary processes and using temporal properties of event in the machine learning based automatic annotation. We compose four temporal properties classifier (E-class, E-dynamic, E-period and E-telicity) and four types of temporal relation identifiers (RLP, RTA, RTP and SUB) which correspond to the attributes of events in section 5.2.

6.2.1 The experiment of temporal property attributes of events

First, we test the performance of our system in the temporal properties annotating task. The distribution of temporal properties in the training data and in the testing data is shown in Table 6-5. We train four machine learning models for analyzing the attributes independently. We test several features and select the useful features for each model. The useful features are shown in Table 6-6. Table 6-7 shows the results of automatic annotating temporal properties. The features are not useful in all training models. The notation "●" means that we use this feature in the model and the notation "−" means un-useful features. The abbreviations "R", "P" and "F" mean "Recall", "Precision" and "F-measure". This automatic annotated data is applied in the automatic temporal relation annotating which is described in next section.

Attribute	value		value		value	
	Train	Test	Train	Test	Train	Test
E-class	actual world		hypothetical world			
	4584	493	2936	239		
E-dynamic	state		dynamic			
	2173	152	5347	580		
E-period	durative		instantaneous		repeat	
	3305	303	3998	414	217	15
E-telicity	telic		non-telic		continue-state	
	3721	384	3695	239	398	109

Table 6-5: The distribution of event properties in training and testing data

Features	E-class	E-dynamic	E-period	E-telicity
Morphological information of focus event	●	●	●	●
2 linearly preceding words	●	●	●	●
2 linearly succeeding words	●	●	●	●
2 linearly preceding events	●	●	●	●
2 linearly succeeding events	●	●	●	●
Head event	●	–	–	–
Head word	●	●	●	–
The signal word of the focus event in the descendant	●	●	●	●
Simple temporal expression in the descendants of the focus event, compare to the published time	●	●	●	–
The children events of the focus event	●	●	●	–
The children words of the focus event	●	●	●	–

Table 6-6: The features for automatically temporal properties annotating

Attribute	value			value			value		
	R	P	F	R	P	F	R	P	F
	Accuracy								
E-class	actual world			hypothetical world					
	0.95	0.94	0.94	0.85	0.81	0.83			
	0.86								
E-dynamic	state			dynamic					
	0.87	0.85	0.86	0.91	0.92	0.91			
	0.85								
E-period	durative			instantaneous			repeat		
	0.76	0.76	0.76	0.84	0.78	0.81	1	0.66	0.80
	0.78								
E-telicity	telic			non-telic			continue-state		
	0.75	0.87	0.81	0.81	0.79	0.80	0.41	0.70	0.52
	0.79								

Table 6-7: The experiment results of automatically temporal properties annotating

6.2.2 The experiment of temporal relation attributes of events

After the temporal properties annotating, we also train four temporal relations annotating models for annotating the relation attributes of events. These four relation-- "RLP", "RTA", "RTP" and "SUB" are abbreviated from the temporal relation attributes—"Rel-linear-preceding (relations of Adjacent event pair)", "Rel-tree-ancestor (relations of Head-modifier event pair)", "Rel-tree-preceding (relations of Sibling event pair)" and "Sub-ord (subordinate)". Because we group some sparse relation types in the testing data (see section 6.1.2), the possible values of attributes in our experiment are summarized as follows:

- RLP: after, before, simultaneous, overlap (includes the values "begun-by", "end-by", "overlap", "overlap-by", "include", "during"), unknown
- RTA: after, before, simultaneous, overlap (includes the values "begun-by", "end-by", "overlap", "overlap-by", "include", "during"), copula-existence, hypothetical, unknown
- RTP: after, before, simultaneous, overlap (includes the values "begun-by", "end-by", "overlap", "overlap-by", "include", "during"), unknown

- SUB: none, hypothetical, explanations, report, introduce, passive, condition

The training data is also the annotated corpus and the testing data is the result of the temporal properties annotating. Table 6-8 shows the distribution of the temporal relation attributes of events in the training / testing data. It should be noted that the number of the attributes of data ignores some negligible instances. Such as, if a verb does not have sibling verbs in the dependency structure, to consider the attribute "RTP (Relation between focus event and its sibling event)" is unnecessary. Therefore the total numbers of the attribute "RTA" and the attribute "RTP" are less than the number of all verbs.

Following the description in section 5.3, the value "hypothetical" of the attribute "E-class" is introduced in the temporal relation type "RTA". If the verb is a hypothetical world event or non-event, the verb is closed into the hypothetical world. The verb in hypothetical world cannot have a RLP relation (Relation of adjacent event pair) between hypothetical and actual world. However, for recognizing the verb that is the root event of the hypothetical world, we annotate the RTA relation (Relation of adjacent event pair) of the root event in hypothetical world as the value "hypothetical". The value "copula-existence" is introduced to annotate the event emphasized by the copula verbs and the existence verbs. If the copula / existence verb governs a verb phrase with several verbs, the root event of the verb phrase has the value "copula-existence".

The features for machine learning are also tuned independently and are shown in Table 6-9. The feature "The SIGNAL word of the event pair in the path to head event" is the meaning of the SIGNAL words in section 6.1.3. The feature "Simple temporal expression in the descendants of the focus event pair, compare to the published time" is the information that we introduce in section 6.1.4. These features are useful for the three relation attributes but are un-useful for the subordinate attribute. "The related position in the dependency structure of focus event pair" is the related relation that we used in section 2.3.2. This is useful for "RLP" but is un-useful in other attributes, because the related position is meaningful only in the adjacent event pairs. The experiment results are shown in Table 6-7. The accuracy of each attribute is the annotating accuracy of the event sequence. We expect that increasing the training data size and investigating more features can improve the accuracy of attributes.

	RLP		RTA		RTP	
	Train	Test	Train	Test	Train	Test
after	2467	270	614	81	1104	121
before	1303	138	1832	180	702	46
simultaneous	1391	82	1026	63	577	20
overlap	240	25	119	26	121	2
hypothetical			1026	122		
copula-existence			417	39		
unknown	2119	217	521	41	107	45
Total	7520	732	5555	552	2611	234

	SUB	
Values	Train	Test
none	3233	301
hypothetical	1542	134
explanations	1523	120
report	566	65
introduce	441	40
Passive	174	35
condition	42	37
Total	7520	732

Table 6-8: The distribution of temporal relations in the testing data

Features	Rel-linear-preceding	Rel-tree-ancestor	Rel-tree-preceding	Sub-ord
Event properties of focus event pairs	●	●	●	●
Morphological information of focus event pairs	●	●	●	●
Distance between focus event pairs	●	●	●	●
Liner preceding event of the focus event pair	●	●	–	●
Liner succeeding event of the focus event	●	●	–	●
Head event of the focus events	–	●	●	●
Head word of the focus events	●	●	●	●
Preceding sibling event of the focus event pair	–	●	–	–
The signal word of the event pair in the path to head event	–	●	–	–
The signal word of the event pair in the descendant	●	●	–	–
Simple temporal expression in the descendant of the focus event pair, compare to the published time	●	●	●	–
Simple temporal expression in the descendant of the focus event pair, compare to each other	●	●	●	–
The words between the focus event pair	●	●	●	●
The events between the focus event pair	●	●	●	●
The related position in the dependency structure of focus event pair	●	–	–	–
The children events of the focus events	●	●	●	●

Table 6-9: The features for automatically temporal relations annotating

	RLP			RTA			RTP		
	R	P	F	R	P	F	R	P	F
after	0.74	0.66	0.70	0.51	0.67	0.58	0.74	0.66	0.70
before	0.53	0.55	0.54	0.81	0.75	0.78	0.49	0.59	0.54
simultaneous	0.42	0.47	0.44	0.48	0.45	0.46	0.42	0.45	0.44
overlap	0.58	0.98	0.73	0.67	0.84	0.74	0.52	0.96	0.67
hypothetical				0.68	0.75	0.71			
copula-existence				0.85	0.76	0.80			
unknown	0.78	0.72	0.75	0.91	0.88	0.89	0.79	0.73	0.76
Accuracy	0.68			0.67			0.71		

	SUB		
Values	R	P	F
none	0.87	0.76	0.81
hypothetical	0.77	0.72	0.74
explanations	0.63	0.57	0.60
report	0.87	0.87	0.87
introduce	0.71	0.79	0.75
passive	0.88	0.81	0.84
condition	0.51	0.31	0.39
Accuracy	0.67		

Table 6-10: The experiment results of automatically temporal relation annotating

6.2.3 The effect of using preliminary processes and temporal properties

The result in Table 6-10 uses the most useful feature into the machine learner. We also investigate the effect of different preliminary processes. Because our system annotates the temporal relation attributes after the preliminary processes and the temporal property annotation, we expect that these processes effect the accuracy of annotating the temporal relation attributes (adjacent, head-modifier and sibling event pairs).

Table 6-11 shows the effect of using different preliminary processes. We divide the experiments as following set up that correspond to the columns in Table 6-10.

- Correct properities: the machine learner is trained with all correct temporal properties and with the result of the signal word identification and the temporal expression identification.

- All automatic identification: the machine learner is trained with the results of all automatic identification processes and with the results of the signal word identification and the temporal expression identification.

- All temporal prosperities w/o E-class: the machine learner is trained with all temporal properties but without the attribute “E-class”. The result of the signal word identification and the temporal expression identification are also used here.
- Only E-class: the machine learner only uses the attribute “E-class” to train the models. The result of the signal word identification and the temporal expression identification are also used here.
- w/o all temporal prosperities: the machine learner does not use any temporal properties. The result of the signal word identification and the temporal expression identification are also used here.
- All automatic identification w/o signal identification: the machine learner does not use the results of the signal word identification. It is trained with the results of all automatic identification processes and the temporal expression recognization.
- All automatic identification w/o simple temporal expression: the machine learner does not use the temporal expression recognization. It is trained with the results of all automatic identification processes and the results of the signal word identification.

Following the results, we observe that using all temporal properties and preliminary processes (the signal word identification and the simple temporal expression recognization) has best proformance. Top line of using the temporal properties is the result in the column “Correct properties”. Comparing the effect of “E-class” and other temporal properties, the “E-class” improves the three temporal relation attributes more than only using other properties. However, the “E-class” cannot improve the sub-ordinate attribute more than using other temporal properties. Certainly, in a head-modifier event pair, whether the events are in the actual world or not do not affect the meaning of the events.

If we do not use the preliminary processes (the signal word identification and the simple temporal expression recognization), the accuracy of annoatating the temporal relation attributes becomes worse. The improvement of using the the signal word is better than using simple temporal expression. It looks like that using simple temporal expression cannot improve the temporal relation annotation. The reason is that the simple temporal expressions are few in our data. In many instances that include the simple temporal expression, only one event related to the temporal expression. Therefore the system cannot acquire useful feature for annotating the temporal relation attributes. For example, our system compares the simple temporal expressions that is the descendants of the focus event pair and uses the result for the temporal relation annotation. If there is only one temporal expression in the focus event pair, the system cannot acquire a comparison[40].

The signal word is useful in the head-modifier relation attribute but it improves the adjacent relation attribute slightly. The reason is that the adjacent relations do not deal with the syntactic structure. In many cases, the signal word between two adjacent events usually do not related to the focus event pair but it is almost related to the head-modifier event pair. This observation shows that the dependency structure is appropriate for describing the temporal relation between events.

Attributes	Accuracy						
The temporal properties of the event							
	Correct properties	All automatic identification	All temporal properties w/o E-class	Only E-class	w/o all temporal properties	All automatic identification w/o signal identification	All automatic identification w/o simple temporal expression
E-class	1	0.86		0.86		0.85	0.86
E-dynamic	1	0.85	0.85			0.85	0.84
E-period	1	0.78	0.77			0.77	0.77
E-telicity	1	0.79	0.79			0.78	0.79
The temporal relation tag of the event							
Adjacent event pairs	0.73	0.68	0.64	0.66	0.62	0.68	0.67
Head-modifier event pairs	0.72	0.67	0.62	0.64	0.60	0.66	0.67
Sibling event pairs	0.73	0.71	0.69	0.69	0.66	0.70	0.70
subordinate	0.69	0.67	0.64	0.62	0.61	0.66	0.67

Table 6-11: The effect of using the temporal properites

[40] Certainly, we can compare any simple temporal expression with the published time. Most of the temporal expressions occur before the published time; therefore we do not compare the simple temporal expressions with the document published time.

6.3 Recall of the proposed method

6.3.1 The recall in all relations and in actual / hypothetical world

Our proposed method only considers the adjacent and the dependency structure of events and created an annotated corpus. After we construct the temporal relation annotating system, we investigate the question again—"how many temporal relations could be identified in the proposed method?" In this section, we investigate the coverage of the temporal relations that are identified by our annotating system and the relations that are annotated manually. That is, to compare the performance between "a trained annotator" and "a trained machine learner".

Table 6-12 describes the results. The column "automatic annotating" is the result of our system. The "manual annotating with proposed method" is using our proposed criteria to annotating the events. The row "true event relations" is the golden data that manually consider all event combinations. The "automatic annotating" and the "manual annotating with proposed method" are almost the same experiments, but one is annotated by the computer and another is by the annotator. The results of our systems include the incorrect results of the dependency structure analyzer and the temporal relation annotating models. Therefore, the coverage of system is worse than "manual annotating with proposed method". In this table, we not only consider the coverage of the temporal relations but also consider the coverage in different world - the actual world and hypothetical worlds. In each block, it includes three areas. The upper area means the total number of relations or the accuracy of identifying the relations in both the actual and hypothetical worlds. The under-left area means the number and accuracy of identifying the relations in the actual world. The under-right area means the number and accuracy of identifying the relations in hypothetical worlds. We divide the relations to different world because the understandable temporal relations are only in the actual world. This can help us to evaluate the practicability of our temporal relation identification system.

First, we evaluate the total accuracy of our system. The aim of our research is this experiment. We use our system to annotate the attributes in a morphological information tagged data. The result shows that our temporal relation identification system can cover

53% (recall) of the all temporal relations. The top line of our system is the result in the column “Manual annotating with proposed method”. 63% (recall) relations can be identified by manual effort with our proposed procedure. The precision in manual effort is 100%. Certainly, our system cannot perfectly identify the temporal relations therefore the precision of our system is 66%. A reason of the worse precision is the multiplication of incorrect temporal relation attributes. We use inference rules to identify long distance temporal relations. However, the inference rules also deduce the incorrect attributes. The output temporal relations therefore include many incorrect relations.

In the viewpoint of actual / hypothetical worlds, the coverage of both actual / hypothetical worlds is similar in manual effort (the top line of our system). However, the recall and the precision in the automatic annotation are both “the hypothetical world is better than the actual world”. The main reason is that the hypothetical worlds have two properties – “local” and “basic”. Many hypothetical worlds are sub-structure in a dependency tree. The influential features for identifying the temporal relations are included in the local structure. We do not need to consider more global features. The “basic” property means that the temporal relations in the hypothetical worlds are general knowledge. They are not affected by the context or syntactic structure. Therefore the system identifies the temporal relation in hypothetical world better than in actual world.

There is no research based on same data set and corpus guideline, therefore we can not compare the result to other research. However, in the shared task—TempEval (see section 2.3), the task “temporal relations between matrix verbs” resembles to the goal of our research. The F-measure in TempEval shared task distribute in 40%~50%. The result of the shared task also shows the difficulty of automatic temporal relation analysis.

			Automatic annotating		Manual annotating with proposed method	
Relations of Adjacent event pair (The attribute Rel-linear-preceding-RLP)	All		711		702	
	actual	hypothetical	467	244	506	196
Relations of Head-modifier event pair (The attribute Rel-tree-ancestor-RTA)	All		526		530	
	actual	hypothetical	378	148	344	186
Relations of Sibling event pair (The attribute Rel-tree-preceding-RTP)	All		221		205	
	actual	hypothetical	154	67	143	62
Total extracted event relations	All		1081		1018	
	actual	hypothetical	729	352	695	323
Extend event relations by using inference rules	All		5155		4166	
	actual	hypothetical	3498	1657	2953	1213
True event relations	All		6646		6646	
	actual	hypothetical	4677	1969	4677	1969
CORRECT relations	All		3522		4166	
	actual	hypothetical	2336	1186	2953	1213
Precision	All		0.66		1	
	actual	hypothetical	0.67	0.71	1	1
Recall	All		0.53		0.63	
	actual	hypothetical	0.50	0.60	0.63	0.62

Table 6-12: The comparison between automatic annotating system and manual annotating

6.3.2 The identificaion efficiency

Our proposed method only covers 63% temporal relations of the all combination of events. An interesting observation (or we can say “a problem”) in our research is that the necessity of the long distance relations. Generally, it is possible that all events in the same article have the temporal relation to each other. Annotators consider the relations of all event combinations with all knowledge that they have. However, readers do not always remember all relations when they are reading an article. Our investigation indicates this observation. Even if our proposed method cannot identify most of the temporal relations (of all event combinations), Using our system can deduce manual efforts.

To investiate the effect of this observation, we evaluate the temporal relation coverage in different method. We require an annotator, who annotated the annotated corpus but did not see the testing data, to identify the temporal relation between events in the testing data (this is the same data set as the data in previous sections) with his thinking. The testing data only includes the morphological information (POS-tags). We do not give the dependency structure and other semantic / syntactic information. We order him to identify the temporal relations "as possible as you can" and we do not require him to consider all combinations of events (this setting is different from section 6.1.2, which considers all combinations with the dependency structure). Then, we compare the result of this manual identification with the temporal relation of all event combinations. The annotator in this evaluation identifies 4191 relations but only 3849 relations are correct (the correct data includes 6646 relations). The recall and precision are 58% and 92%. This result shows that the all event combinations possibly not correspond to the information that human can understand. The annotator only extracts the information that he need in this experiment.

Actually, this experiment is appropricate to the annotating work in TimeBank. They also considered "the possible temporal relation" and did not consider "all event combinations". We can use the annotating data in this experiment as the golden standard for evaluating the coverage of our system. However, the annotating results between different annotators are in consistent and then the inconsistent data cannot be a believable standard. We need to train more annotators if we will use this method to create the golden standard.

We also compare the result of this manual identification with the result in Table 6-12. Because we do not require the annotator to consider all event combinations, the annotator does not focus on long distance relations. Additionally, because our proposed method includes using the inference rules to identify long distance relations, the manual annotating result with our proposed method can identify more temporal relations than this manual identification. Therefore, these manually identified relations are fewer than considering all combinations and manually annotated by our criteria. Certainly, the annotator does not have information of the dependency structure; the manually identified relations are not all correct relations. The total working time of the manually identification is six hours in this experiment, but the coverage of it is only better than our automatic annotating system slightly. To consider the cost, using our system to annotate the temporal relation attributes than refine it is more efficient (the total processing time of our system is 35 seconds).

6.4 Summary

In this chapter, we apply the temporal relation annotated corpus to train the machine learner and evaluate the performance and coverage of our proposed system. First, we investigate four preliminary experiments before we evalute the performance of our system. First, we introduce the testing data and estimate the consistency of our annotating work. Second, we investigate the coverage of our proposed annotating criteria. We compare the annotating result using our criteria and annotating all event pairs to investigate the coverage of our criteria. Third, we describe the experiment of a machine learning based SIGNAL word identification. Final preliminary experiment is simple temporal expression recognitions. We use a simple rule set to recognize the meaning of numerical expressions.

Second, we examine the performance of our temporal relation annotating system. This experiment includes annotating the temporal properties and the temporal relation attributes of events. We use SVM as machine learner to compose a temporal relation identifier. The training data is our annotated corpus in section 5.5.2 and the testing data is another small annotated corpus in section 6.1.1. We perform experiments to investigate the accuracy of automatic annotation the attributes which include temporal properties of events and temporal relations between event pairs.

Finally, we examine the recall of our proposed system. We compare the result of our proposed system and a complete manual work to observe that how many temporal relations can be identified by our system. We investigate the coverage of the temporal relations that are identified by our annotating system and the relations that are annotated manually. That is, to compare the performance between “a trained annotator” and “a trained machine learner”. The accuracies of the annotating experiments are 78%~85% for annotating the temporal property attributes and 68%~71% for annotating the temporal relation attributes. We survey the coverage of our system with a small corpus. The result shows that our proposed system covers about 53% of temporal relations of all possible event pairs.

Chapter 7 Conclusion and Future Direction

7.1 Conclusion

"Temporal information (Time)" has been a subject of study in many disciplines particularly in philosophy, physics, and is an important dimension of natural language processing. The temporal information includes temporal expressions, event and temporal relations. There are many researches dealing with the temporal expressions and event expressions. However, researches on temporal relation identification and the construction of temporal relation annotated corpus are still limited. There is a well-known temporal information annotated guideline for English, TimeML. However, there is no such a research that focus on this in Chinese. Our research is the first work of the temporal relation identification between verbs in Chinese texts. In this research, we propose a machine learning-based temporal relation identification method and construct a automatic identifying system.

Following the observation of our investigation, the distribution of events and temporal expressions is un-balance. The temporal information processing includes two independent tasks: anchoring the temporal expressions on a timeline and ordering the events to temporal order. Our research focus on ordering the events, that is to identify the temporal relations between events. We proposed an annotation guideline for a temporal relation tagged corpus of Chinese and then apply the corpus to construct an automatic temporal relation identifier.

Because identifying the nominal event is difficult, we limit the events to the verbs in articles. The proposed guideline is based on the TimeML language but we also use dependency structure information to acquire more meaningful temporal relations and to reduce manual effort. This proposed method reduces the manual efforts in constructing the annotated corpus. To annotate temporal relations of all combinations of events requires C_2^n manual judges. Our proposed method requires at most 3n manual judges. While the dependency structure based attributes reduce manual annotation costs, the limited relations preserve the majority of the temporal relations.

We use a syntactic parsed corpus—Penn Chinese treebank as the original data for annotating a basic annotated corpus. For using the dependency structure in temporal relation identification, we first construct a dependency analyzer for Chinese and combine it into the temporal relation annotating system. The accuracy of the dependency analyzer is 88% for word dependency analysis and this is better than existed Chinese dependency analyzer.

The process of temporal relation identification includes following steps: to analyze the dependency structure, to analyze the temporal properties of events, to analyze the temporal relation attributes of events and to extend the relation using the inference rule. We create eight machine learning models for each attribute of events. SVM is used as the machine learner in our experiments. The accuracies of the annotating experiments are 78%~85% for annotating the temporal property attributes and 68%~70% for annotating the temporal relation attributes. We survey the coverage of our system with a small corpus. The result shows that our proposed system covers about 52% of temporal relations of all possible event pairs. The average working time required for one article (with 80 events) is about 30 minutes in our annotation work. It is shorter than the annotating work of TimeBank, this needs more than one hour for one article. The effectiveness of our proposed system is described.

7.2 Future directions

7.2.1 The system improvement

For improving the performance of our temporal relation annotating system, there are three directions that we can focus on: to increase the training data size; to add more information of temporal expressions; to add inference rules with syntactic / semantic information and to construct a verb causal relation list of Chinese for applying in machine learning.

First, our machine learner needs more training data. The training data for each model has only 7520 instances. We cannot affirm that the amount of the training data is large enough to make the system practicable. To add more manpower to annotate more training data set is a major future direction of our research. However, the consistency of annotating results is not high between different. Even if we train two annotators at the same time, the consistency of their data is only 81%~89%. This observation may is considered as that our temporal relation types cannot factually describe the situations in Chinese. For example, the attribute distribution in Table 5-7 tends to distribute in several types. Probably, readers do not consider the circumstantial relation types that we defined; we need to consider the definition of types again. However, we need to collect more instances to investigate the distribution for proving this idea.

Second, we need to add more information for the machine learner. For example, we only use the simple numerical expression as a feature. However, many useful temporal expressions cannot be applied, such as the expression of time intervals (ex. 兩小時 (two hours)). We need to refer to other related researches for applying more temporal expressions. Adding the inference rules is another future direction for our research. We do not use the syntactic / semantic information to define the rules. Adding inference rules can supply more long distance relation, which is a deficient of our system, but this direction needs more linguistic investigation in Chinese.

Finally, another deficient of our experiment is that we do not use the semantic information as features for machine learner. The semantic information of the temporal and event expressions is important for recognizing temporal relations between events. As a future research, we would like to introduce the causal relation knowledge of verbs (this is similar to VerbOcean (Chklovski, 2004 [16])). Now, we are constructing Chinese causal relation knowledge of verbs. We select the verb pairs in VerbOcean that have the causal relations (before-after) and then translate these verb pairs to Chinese manually. This work needs a lot of manual effort because we need the verb pairs to correspond to our verb dictionary. We forecast that this causal verb pairs is useful information for our system.

7.2.2 The applications

We expect that our temporal relation identidication system can be applied in many NLP applications, such as machine translation and Q&A system. Following the description in section 1.3.3, for example, to translate the Chinese sentences in section 1.2.1 to English with correct verb tense, our proposed system can identify the temporal relation between events "去/便利/商店/時 (go to the convenience store)" and "幫/我/買/汽水 (to buy a soft drink for me)" / "看到/他/在/買/汽水 (I saw he was buying a soft drink)". And then the tense of the verb"買 (to buy)" can be decided in different context.

In an information retrieval system, the relevancy of a query has a temporal aspect from a user's perspective (Alonso et al., 2007 [94]). The more data sources an information retrieval system acquires, the more important the temporal aspect can be in the retrieval process. Instead of assuming that the user wants relevant search results implicitly sorted by date, it would be interesting to investigate a system that is aware of time for relevancy and shows search results in a temporal context. Following the description in section 1.1.2 and 1.2.2, many events do not have their monopolize implict temporal expression. To require the answer of the query, identifying the temporal relation between events without the temporal expression recognization to acquire the causal relation is an efficient method. Our proposed system can satisify these motivations.

Acknowledge

負笈來日，轉瞬五載，自研究生，碩士，乃至於博士。投身自然語言處理之研究，實爲物理出身之吾所始料未及。此誠爲機緣之巧，蒙交流協會之眷顧，令吾得領受獎學金來日，不致有後顧之憂。然則，未有恩師松本教授之惠予收留，留日之夢仍屬空言。回思至今之留學生活，無師友至親之扶持，絕無今日之學位。恩重無以爲報，僅能於此寥以數言，略爲示己之感恩。

恩師松本教授，不僅予吾求學之機會，更啓迪吾於此領域，令吾得領會研究之樂。領受恩師之教澤已有五載，恩師於學術之餘，更以身教言教給予吾人生徒以學者風範。學者除以其所精之學術爲體，更需以師生協同研究爲用，方可有所大成。顏淵曰：『仰之彌高，鑽之彌堅，瞻之在前，忽焉在後！夫子循循然善誘人：博我以文，約我以禮，欲罷不能，既竭吾才，如有所立卓爾，雖欲從之，末由也已！』吾觀恩師，實有如顏淵觀夫子之慨。恩師稟持『教學相長』之道指導生徒，互以砥礪切磋，以更進學術之道。古人讚韓昌黎爲『匹夫而爲百世師』，非僅爲其人之學有所精，更爲其爲人之所得以令後生所仰。吾恩師風範，實爲吾人生徒仰矚之良師典範，吾得入此門師事恩師，實爲萬幸。

子曰：『小人之德草，君子之德風，風行而草偃』。吾人生徒拜於恩師門下，春風化雨，自必受其感化。諸位師長皆頗有恩師之風，對後進之指導不遺餘力。淺原助理教授以先期學長任爲教員，實際指引討論實驗與研究之細節，賴其指導，研究得以順利進行，遙想吾初入此門，於語言處理一道實爲白丁，全有賴淺原助手不厭其煩巨細靡遺給予指導，令吾免於迷途摸索之弊，實爲大恩。日後獨立進行博士論文之研究，亦有賴淺原助理教授的經驗與共同討論方得啓發新研究，在此特表謝意。副指導教授之乾副教授、石井教授以及諸位同窗，於發表會及日常討論中，對吾研究疏漏不備之處多加指導，令論文之論述更爲完備，感激不盡。負笈於異國，瑣事皆需自理，萬般不便之際幸有諸位同學爲之後援，得無所窒礙，非但爲諸位之宅心仁厚，更爲日本文化之精神體現。留學於此，於學術鑽研之外，更需對此精神文化深刻體認，將其汲取之本國予光大之，方爲留學之道。吾於此尙待學習觀摩，但仍願盡一己之力，取他山之石並紹述我國文化，爲兩國交流盡己微力。

子曰：『遊必有方』，吾雖爲有方，但仍爲遠遊，及長至此，未能多於膝下盡孝，汗顏無已。雙親康健平順，身爲人子，此實爲吾負笈在外之慰藉，生之，長之，教之，親恩似海，無以報之於萬一，唯盼以學而尊親，得以略以盡孝之。

最後，需對留日期間最大之精神支柱，亦即自今將相伴共度一生的內人美玲表予謝意，吾負笈於異鄉，自必於物質生活之外有諸多精神層面之壓力，沒有內人的長期支持並鼓勵，吾絕不可能克服挫敗，勇往直前，順利完成學業。吾所僅有之一切成就與榮耀，皆與內人共有共享。

負笈期間承蒙諸多人仕協助，未能逐一列舉完備，僅以此文，一併致謝於諸位給予吾諸多指導協助之教員與友人，以及內人，恩師，雙親，在此感謝。

民國九十七年歲次戊子 乙卯月乙酉日

References

[1] James F. Allen, 1983. Maintaining Knowledge about Temporal Intervals, Communications of the ACM Vol. 26, No.11.

[2] James F. Allen, R. Gupta, and V. Khandelwal, 2001. Temporal summaries of news topics. In Proceedings of the ACM SIGIR conference on Research and development in information retrieval, Pages 10-18.

[3] Y. Altun, I. Tsochantaridis, and T. Hofmann. 2003. Hidden markov support vector machines. In Proc. of ICML-2003.

[4] Emmon Bach, 1986. the algebra of events, Linguistics and Philosophy 9.

[5] Steven Bethard and James H. Martin. 2007. CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features. ACL 2007 Workshop: SemEval-2007.

[6] Thorsten Brants, 2000. TnT - A Statistical Part-of-Speech Tagger, ANLP 2000.

[7] Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing, EMNLP-CoNLL 2006, pp. 149-164.

[8] Yuen Ren Chao, 1968. A Grammar of Spoken Chinese. Berkeley, CA: University of California Press.

[9] Eugene Charniak, 2001. Immediate-Head Parsing for Language Models NAACL-2001.

[10] Keh-Jiann Chen, Chin-Ching Luo, Zhao-Ming Gao, Ming-Chung Chang, Feng-Yi Chen, Chao-Jan Chen, 1999. The CKIP Tree-bank: Guidelines for Annotation, Presented at ATALA Workshop.

[11] Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto, 2007. NAIST.Japan: Temporal Relation Identification Using Dependency Parsed Tree. ACL 2007 Workshop: SemEval-2007.

[12] Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto, 2007. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure Analysis. TIME 2007.

[13] Yuchang Cheng, 2005. Chinese Deterministic Dependency Analyzer: Examining Effects of Chunking, Root node finder and Global Features, Master's thesis, NAIST.

[14] Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto, 2005. Machine Learning-based Dependency Analyzer for Chinese, Journal of Chinese Language and Computing, Vol.15, No.1, pp.13-24.

[15] Nancy Chinchor. 1997. MUC-7 named entity task definition. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html.

[16] T. Chklovsky and P. Pantel. 2004. Verbocean: Mining the web for fine-grained semantiv verb relations. In Proc. of EMNLP-2004.

[17] Yong Jin Chu and Tseng Hong Liu. 1965. on the shortest arborescence of a directed graph. Science Sinica, 14, pp.1396-1400.

[18] Church K., 1988. A stochastic parts program and noun phrase parser for unrestricted text. Proceedings of the Second Conference on Applied Natural Language Processing

[19] CKIP, 1993. 中文詞類分析. CKIP Technical report no. 93-05.

[20] Michael Collins, 1999. Head-Driven Statistical Models for Natural Language Parsing. PhD Dissertation, University of Pennsylvania

[21] Michael Collins, Brian Roark, 2004, Incremental parsing with the Perceptron algorithm, ACL-2004, pp.112-119.

[22] Michael Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In Proc. of ACL-2002.

[23] Koby Crammer and Yoram Singer. 2003. Ultraconservative Online Algorithms for Multiclass Problems, JMLR.

[24] Bonnie J. Dorr and Mari Broman Olsen, 1997. Deriving Verbal and Compositional Lexical Aspect for NLP Application, ACL 1997.

[25] Jack Edmonds. 1967. Optimum branchings. Journal of Research of the National Bureau of Standards, 71B, pp.233-240.

[26] Jason Eisner, 1996, Three new probabilistic models for dependency parsing: An exploration. Proceedings of COLING 1996, pp.340-345.

[27] L. Ferro, L. Gerber, I. Mani, B. Sundheim and G. Wilson. 2002. TIDES. Instruction Manual for the Annotation of Temporal Expressions. Technical Report Interim Draft for Terqas Workshop, The MITRE Corporation.

[28] Haim Gaifman, 1965. Dependency Systems and Phrase-Structure Systems. Information and Control 8, 304-337

[29] GOH, Chooi-Ling, 2006. Unknown Word Identification for Chinese Morphological Analysis, Doctor's thesis, NAIST.

[30] Caroline Hagège and Xavier Tannier. 2007. XRCE-T: XIP Temporal Module for TempEval campaign. ACL 2007 Workshop: SemEval-2007.

[31] David G. Hayes, 1964. Dependency Theory: A Formalism and Some Observations. Language 40, pp.511-525.

[32] Mark Hepple, Andrea Setzer and Robert Gaizauskas. 2007. USFD: Preliminary Exploration of Features and Classifiers for the TempEval-2007 Task. ACL 2007 Workshop: SemEval-2007.

[33] C.N. Huang, C.F. Yuan and S.M. Pan, 1992. 語料庫，知識獲得和句法分析. Journal of Chinese Information Processing, Vol.6, No.3, pp.1-6

[34] J. Huang, 1982. Logical relations in Chinese and the theory of grammar Doctoral dissertation, Massachusetts Institute of Technology, Cambridge.

[35] Dick Hudson, 2000. Dependency Grammar, ESSLLI 2000.

[36] Hideki Isozaki, Hideto Kazawa and Tsutomu Hirao, 2004. A deterministic word dependency analyzer enhanced with preference learning. COLING-2004, pp. 275-281.

[37] IREX Committee, 1999. Named entity extraction task definitionn, http://nlp.cs.nyu.edu/irex/NE/df990214.txt.

[38] Ray Jackendoff, 1992, Semantic Structure, MIT Press.

[39] Ulrich. H.-G. Kreβel, 1998. Pairwise classification and support vector machines. In Advances in Kernel Methods, pp.255–268. The MIT Press.

[40] Karlsson, F., A. Voutilainen, J. Heikkilä and A. Antilla, 1995. Constraint Grammar: a Language-independent System for Parsing Unrestricted Text.

[41] Pyung Kim and Sung Hyon Myaeng, 2004. Usefulness of temporal information automatically extracted from news articles for topic tracking. Transactions on Asian Language Information Processing (TALIP) Vol. 3, Issue 4, pp. 227-242.

[42] Mayumi Kudo. 2004. The system of tense-aspect and text-the temporal expression in modern Japanese, pp. 26 (In Japanese).

[43] Taku Kudo, Yuji Matsumoto, 2002. Japanese Dependency Analysis using Cascaded Chunking, CONLL-2002.

[44] Taku Kudo, Yuji Matsumoto, 2003. Fast Methods for Kernel-Based Text Analysis, ACL-2003.

[45] Taku Kudo and Yuji Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. In Proc. of EMNLP-2004.

[46] Lai, Bong Yeung Tom, Huang, Changning, 1994. Dependency Grammar and the Parsing of Chinese Sentences. PACLIC 1994

[47] Lin-Shan Lee, Long-Ji Lin, Keh-Jiann Chen, and James Huang, 1991. An Efficient Natural Language Processing System Specially Designed for the Chinese Language. ComputationaI Linguistics, Volume 17, Number 4.

[48] Wenjie Li, Kam-Fai Wong and Chunfa Yuan, 2001. Application and Difficulty of Natural Language Processing in Chinese Temporal Information Extraction, NLPRS 2001.

[49] Wenjie Li, Kam-Fai Wong, Guihong Cao and Chunfa Yuan, 2004. Applying Machine Learning to Chinese Temporal Relation Resolution, ACL 2004.

[50] Mingli Wu, Wenjie Li, Qin Lu and Baoli Li, 2005. CTEMP: A Chinese Temporal Parser for Extracting and Normalizing Temporal Information, IJCNLP 2005.

[51] Charles Li, and Thompson Sandra A., 1981. Mandarin Chinese. University of California Press.

[52] Chih Jen Lin, 2001. A practical guide to support vector classification , http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[53] Jo-wang Lin. 2005. Time in a Language Without Tense: the Case of Chinese, Journal of semantics Vol. 23, pp. 1-53.

[54] McCord M., 1990. A System for Simpler Construction of Practical Natural Language Grammars. Natural Language and Logic. Lecture Notes in Artificial Intelligence 459.

[55] Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee and James Pustejovsky. 2006. Machine learning of temporal relations. In Proc. of ACL-2006.

[56] Inderjeet Mani, James Pustejovsky and Rob Gaizauskas (ed.), 2006. the language of time, Oxford University express.

[57] M. Marcus, B. Santorini and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. 19(2):313–330.

[58] Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In Proc. of EACL-2006.

[59] Ryan McDonald, Fernando Pereira, Kiril Ribarov and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In Proc. of HLP-EMNLP-2005, p.523-530.

[60] Congmin Min, Munirathnam Srikanth and Abraham Fowler. 2007. LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text. ACL 2007 Workshop: SemEval-2007.

[61] Tetsuji Nakagawa. 2007. Multilingual Dependency Parsing Using Global Features, EMNLP-CoNLL 2007, pp. 952-956.

[62] Joakim Nivre and Mario Scholz, 2004. Deterministic Dependency Parsing of English Text. Pages 64-70, COLING-2004.

[63] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing, EMNLP-CoNLL 2007, pp. 915-932.

[64] Kemal Oflazer, 2003. Dependency Parsing with an Extended Finite-state Approach, Computational Linguistics, Volume 29, No.4

[65] Martha Palmer, Fu-Dong Chiou, Nianwen Xue, and Tsan-Kuang Lee, 2005. Chinese Treebank 5.1 http://www.ldc.upenn.edu/, LDC.

[66] Georgiana Puscasu. 2007. WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis. ACL 2007 Workshop: SemEval-2007.

[67] James Pustejovsky, Marc Verhagen, Roser Sauri, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen and Andrea Setzer, 2006. TimeBank 1.2, http://www.ldc.upenn.edu/, LDC.

[68] J. N. Darroch and D. Ratcliff. 1972. Generlized Iterative Scaling for Log-linear Models. In Annals of Mathematical Statistics, volume 43, Issue 5, pp.1470-1480.

[69] J. J. Robinson, 1970. Dependency Structures and Translation Rules. Language 46, pp. 259-285.

[70] Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer and James Pustejovsky, 2005. TimeML Annotation Guidelines, http://www.timeml.org/.

[71] L. Tesniere , 1959. Elements de Syntaxe Structurals. Paris: Klincksieck.

[72] Vladimir N. Vapnik, 1998. Statistical Learning Theory. A Wiley-Interscience Publication.

[73] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple,Graham Katz and James Pustejovsky, 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. ACL 2007 Workshop: SemEval-2007.

[74] Zeno Vendler, 1967. Verbs and Times, Linguistics in Philosophy, Cornell University Press.

[75] S. V. N. Vishwanathan and A. J. Smola. 2002. Fast kernels on strings and trees. In Proc. of NIPS-2002.

[76] Nianwen Xue, Fu-Dong Chiou and Martha Stone Palmer, 2002. Building a Large-Scale Annotated Chinese Corpus. COLING 2002.

[77] Fei Xia, 2000. The Part-Of-Speech Tagging Guidelines for the Penn Chinese Treebank, http://www.cis.upenn.edu/~chinese/ctb.html.

[78] Hiroyasu Yamada and Yuji Matsumoto, 2004. Support Vector Machine を用いた決定性上昇型依存構造解析 情報処理学会論文誌, Vol.45, No.10, pp.2416-2427.

[79] 陳平，1987. 釋漢語中與名詞性成份相關的四組概念，中國語文，Vol. 2 (in Chinese).

[80] 鄧守信，1986. 漢語動詞的時間結構，第一回國際中國語教育研討會論文集 (in Chinese).

[81] 張秀, 1957. 漢語動詞的”體”和”時制”系統，載語法論集第一集 (in Chinese).

[82] 馬慶株, 2000. 漢語動詞的與動詞性結構，北京大學出版 (in Chinese).

[83] David D. Ahn, Joris van Rantwijk, and Maarten de Rijke, 2007. A cascaded machine learning approach to interpreting temporal expressions. HLT-NAACL 2007.

[84] Benjamin Han, Donna Gates and Lavie Levin, 2006. From Language to Time: A Temporal Expression Anchorer. TIME 2006.

[85] Taichi Noro, Takashi Inui, Hiroya Takamura and Manabu Okumura, 2006. Time Period Identification of Events in Text, In ACL 2006, pp. 1153-1160.

[86] Vivian Tsang, Suzanne Stevenson and Paola Merlo, 2002. Crosslinguistic Transfer in Automatic Verb Classification. COLING 2002, pp. 1023-1029.

[87] Xiaodan Zhu, Chunfa Yuan, K. F. Wong and Wenjie Li, 2000. An algorithm for situation classification of Chinese verbs. Tthe second workshop on Chinese language processing, in ACL 2000, pp. 140-145

[88] Inderjeet Mani and George Wilson, 2000. Robust temporal processing of news. In ACL 2000, pp. 67-76.

[89] Inderjeet Mani, Barry Schiffman and Jianping Zhang, 2003. Inferring temporal ordering of events in news. HLT-NAACL 2003, pp. 55-57

[90] Branimir Boguraev and Rie Ando, 2005. TimeML-Compliant Text Analysis for Temporal Reasoning. IJCAI 2005, pp. 997-1003.

[91] Yang Ye, Victoria Li Fossum, and Steven Abney, 2006. Latent features in automatic tense translation between Chinese and English. In Fifth SIGHAN Workshop on Chinese Language Processing, pp. 48-55.

[92] Kadri Hacioglu, Ying Chen and Benjamin Douglas, 2005. Automatic Time Expression Labeling for English and Chinese Text. CICLing 2005, pp. 548-559.

[93] Frank Schilder and Christopher Habel, 2001. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. Proceedings of ACL 2001 workshop on temporal and spatial information processing, pp. 65-72.

[94] Omar Alonso and Michael Gertz and Ricardo Baeza-Yates, 2007. On the value of temporal information in information retrieval. SIGIR Forum, Vol. 42. No.2, pp. 65-72.

[95] Wenjie Li and Kam-Fai Wong, 2002. A word-based approach for modeling and discovering temporal relations embedded in Chinese sentences, ACM Transactions on Asian Language Information Processing (TALIP), Vol. 1, Issue 3, pp.173-206.

List of Publications

Journal Papers

1. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2008. Use of Event Types for Temporal Relation Identification in Chinese Text. *International Journal of Computer Processing of Oriental Languages (IJCPOL),* Vol.21, No.1, to appear.

2. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2008. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure Analysis. *Journal of the Association for Computational Linguistics and Chinese Language Processing (CLCLP),* to appear.

3. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2005. Machine Learning-based Dependency Analyzer for Chinese. *Journal of Chinese Language and Computing,* Vol.15, No.1, pp.13-24.

International Conferences/Workshops (Reviewed)

4. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2008. Use of Event Types for Temporal Relation Identification in Chinese Text. In *Sixth SIGHAN Workshop on Chinese Language Processing, Proceedings of the Workshop (SIGHAN 2008)*, pp.31-38.

5. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2007. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure Analysis. In *14th International Symposium on Temporal Representation and Reasoning (TIME 2007*), pp.59-69.

6. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2005. Chinese Deterministic Dependency Analyzer: Examining Effects of Global Features and Root Node Finder. In *Fourth SIGHAN Workshop on Chinese Language Processing, Proceedings of the Workshop (SIGHAN 2005)*, pp.17-24.

7. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2005. Machine Learning-based Dependency Analyzer for Chinese. In *Proceedings of the International Conference on Chinese Computing (ICCC 2005)*, pp.66-73.

8. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2004. Deterministic dependency structure analyzer for Chinese. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP 2004)*, pp.135-140.

International Conferences/Workshops (Others)

9. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2007. NAIST.Japan: Temporal Relation Identification Using Dependency Parsed Tree. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, pp.245-248.

10. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2006. Multi-lingual Dependency Parsing at NAIST. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL 2006)*, pp.191-195.

Local Conferences/Workshops

11. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2008. Constructing a Temporal Relation Identification System of Chinese based on Dependency Structure Analysis. In *Information Processing Society of Japanese SIG Notes, NL-184*, to appear.

12. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2007. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure Analysis. In *Information Processing Society of Japanese SIG Notes, NL-179*, pp.61-66.

13. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2007. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure. In *21th Annual Conference of the Japanese Society for Artificial Intelligence*, 3I1-5.

14. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2005. Improving a Deterministic Dependency Analyzer for Chinese. In *11th Annual Meeting of the Association for Natural Language Processing*, pp.907-910.
15. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. 2004. Deterministic dependency structure analyzer for Chinese. In *Information Processing Society of Japanese SIG Notes, NL-163*, pp.91-98.

Award

16. Information Processing Society of Japanese SIG Notes NL-179 Best Student Presentation Award. 2007. Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto. Constructing a Temporal Relation Tagged Corpus of Chinese based on Dependency Structure Analysis.