MULTILINGUAL NAMED ENTITY EXTRACTION AND TRANSLATION FROM TEXT AND SPEECH

Fei Huang

December 2005
CMU-LTI-06-001

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

THESIS COMMITTEE
Alexander Waibel (Chair)
Stephan Vogel
Alon Lavie
Tanja Schultz
Kevin Knight (USC/ISI)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Acknowledgements

I want to thank this age, the age of globalization, especially its positive effect on China. When I was 10 years old, I could not imagine I would visit another country tens of thousands of miles away, living there for more than six years for my doctoral study, and having the opportunity of appreciating different cultures. I hope the globalization will continue to shape everyone's life in a positive manner.

I owe a lot to my thesis committee members. I greatly benefited from the scientific vision of my advisor, Alex Waibel. He often pointed out new directions in my research, helped me develop independent research ability and challenged my over-prudence. "Be BOLD!" Yes I will keep this as my lifelong reminder. I also appreciate the constant support from my direct supervisor, mentor and friend, Stephan Vogel. He has always been available for both scientific and causal discussions. I gained a lot from his rich experience in machine translation. Working with him has been fun, especially during the DARPA machine translation evaluations! Alon Lavie and Kevin Knight helped me both scientifically and professionally. Their insightful questions and detailed comments on my dissertation helped improve the quality of this thesis work. I am also grateful to the tremendous help from Tanja Schultz, whose expertise on speech processing and warm-heartedness really impressed me.

In addition, I want to thank many members in the interACT center. Jie Yang introduced me into the Interactive System Labs and helped me adjust to a different culture. Wilson Tam is an enjoyable officemate and sports partner, who helped me with many speech recognition problems. Ying Zhang is a great colleague and fun

pal. I truly enjoyed the interactions with Nguyen Bach, Celine Carraux, Matthias Eck, Sanjika Hewavitharana, Silja Hildebrand, Chiori Hori, Qin Jin, Stan Jou, Mohamed Noamany, Yue Pan, Shirin Saleem, Thomas Schaaf, Ashish Venugopal, Hua Yu and Bing Zhao.

Finally I want to thank my family. I am deeply indebted to my parents, who provide me physical, intellectual and emotional supports all along, and my wife who supports me at the hardest time during the study.

Abstract

Named entities (NE), the noun or noun phrases referring to persons, locations and organizations, are among the most information-bearing linguistic structures. Extracting and translating named entities benefits many natural language processing problems such as cross-lingual information retrieval, cross-lingual question answering and machine translation.

In this thesis work we propose an efficient and effective framework to extract and translate NEs from text and speech. We adopt the hidden Markov model (HMM) as the baseline NE extraction system, and investigate its performance in multiple language pairs with varying amounts of training data. We expand the baseline text NE tagger with a context-based NE extraction model, which aims to detect and correct NE recognition errors from automatic speech recognition hypotheses. We also adapt the broadcast news trained NE tagger for meeting transcripts.

We develop several language-independent features to capture phonetic and semantic similarity measures between source and target NE pairs. We incorporate these features to solve various NE translation problems presented in different language pairs (Chinese to English, Arabic to English and Hindi to English), with varying resources (parallel and non-parallel corpora as well as the World Wide Web) and different input data streams (text and speech).

We also propose a cluster-specific name transliteration framework. By grouping names from similar origins into one cluster and training cluster-specific transliteration and language models, we manage to dramatically reduce the name transliteration error rates.

The proposed NE extraction and translation framework improves NE detection performance, boosts NE translation and transliteration accuracies and helps increase machine translation quality. Overall, it significantly reduces NE information loss caused by machine translation errors and enables efficient information access overcoming language and media barriers.

Contents

List of Figures

LIST OF FIGURES

List of Tables

Chapter 1

Introduction

1.1 Motivation

The amount of electrically accessible information has been increasing dramatically over the last decade. By 15:28:02 June 14, 2005, Google has indexed 8,058,044,651 web pages. Back to year 2001, this number is 2 billion (Sherman (2001)). Lawrence & Giles (1999) estimated that the whole publicly indexable web data had 15 terabyte in 2000, among them 6 terabyte were textual content. According to Lyman *et al.* (2003), new information produced and flowing in 2002 are as much as the following:

Information object	**How many bytes**
Total new information produced in 2002	5 exabytes
New information produced per person	800 megabytes
New information stored in hard disk	92%
Information flow: telephone information	17.3 exabyte
Information flow: TV and radio broadcast information	3,500 terabytes
Information flow: World Wide Web information	170 terabytes
Information flow: instant messaging	274 terabytes
Information flow: e-mail information per year	400,000 terabytes

Table 1.1: How much information exists

Although this large information pool offers human information analysts a great opportunity of discovering valuable information, the overwhelmingly

huge amount is really a burden. It is extremely important to detect and extract desired information in an efficient manner.

On the other hand, this information can be presented in many human languages. In September 2004 Global Reach [1] estimated the number of people online in each language zone, and their result is shown in Figure 1.1. Although English was the most widely used language online, there were still 65% non-English speakers and resources. Nowadays or in the near future, Chinese may replace English as the most often used language on the internet. As a result, when a user fortunately locates web pages or documents containing desirable information, he or she can not understand, utilize and manage them if they are presented in a foreign language. Therefore, it is an essential technique to extract structured key information from unstructured data oceans and translate them into user-understandable languages.

Named entities (NE), the noun or noun phrases referring to persons, locations and organizations, are among the most information-bearing linguistic structures. Extracting and translating named entities benefits many natural language processing tasks. NE recognition is one of the major tasks in information extraction, and NEs are often key queries in information retrieval, correct answers in question answering, and indicative features for summarization. On the other hand, correct NE translations broaden the scope of information access by incorporating facts presented in foreign languages. They bridge related entities from different languages. They are also key structures in multilingual natural language processing such as cross-lingual information retrieval (CLIR) and machine translation (MT). In machine translation, incorrect NE translation not only loses meaningful information from original sentences, but also introduces distorted semantic context which degrades the overall translation quality.

Extracting NEs from well-formed text such as newswire text has been intensively investigated in the past decade. Both NE recognition rules and statistical models have been developed either manually or automatically, and satisfactory performances have been achieved in multiple languages, including Chinese, English, Japanese and Spanish (Chinchor (1998)). However, extracting

[1] http://global-reach.biz/globstats/index.php3

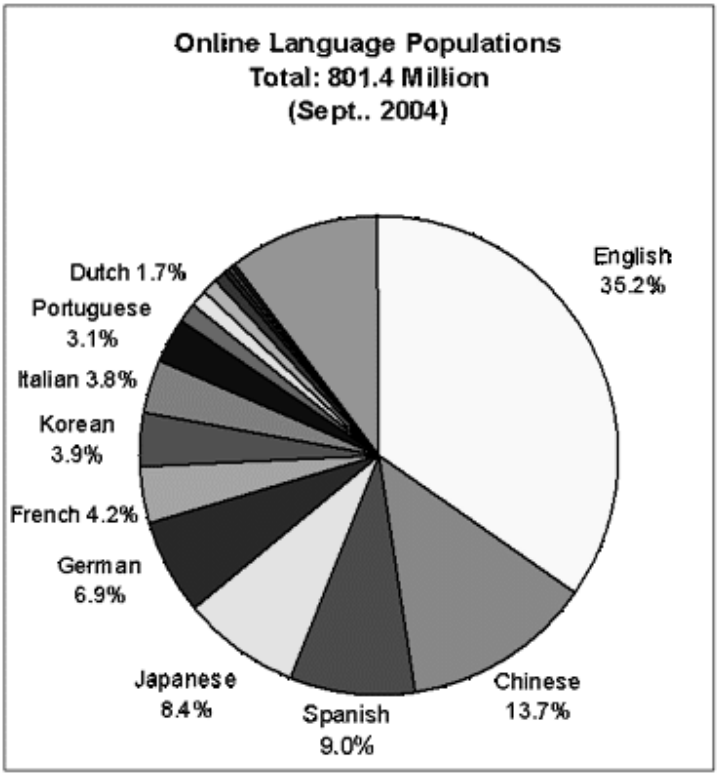


Figure 1.1: Online Language Populations

NEs from ill-formed data, such as speech input, is still a challenging problem. The reasons are:

- Speech transcripts lack format information, such as case and punctuation marks, which can greatly facilitate NE extraction;
- In terms of style and genre, spoken language is quite different from formal written text. Rules and models trained from written language may not fit the speech data well;
- Automatic speech recognition (ASR) errors are particularly difficult for NE extraction and translation from speech, as many ASR errors often occur around NE words.

On the other hand, NE translation is also very challenging. In addition to typical machine translation problems such as word alignment, word reordering and many-to-many translation correspondences [1], special challenges need to be handled for NE translation:

- NE translations involve both phonetic transliteration (translation by pronunciation) and semantic translation (translation by meaning), and sometimes both strategies are used. For example, the English NE "Appalachian Mountain" is translated into Chinese as "阿巴拉契亚 山脉", where "Appalachian" is phonetically transliterated as "阿巴拉契亚 / abalaqiya", and "Mountain" is semantically translated as "山脉 / shanmai";
- Unlike common nouns and verbs, NEs often include many out-of-vocabulary (OOV) words, which are not covered by existing bilingual corpora or translation dictionaries. One has to find solutions to translate these OOV NE words.
- If NEs are automatically extracted from text and speech, speech recognition errors and automatic NE extraction errors further complicate the translation problem.

[1] One source word, phrase or sentence can have more than one correct target translations.

1.2 Thesis Statement

This thesis work provides a language-independent framework that extracts and translates NEs from unstructured text and speech documents in multiple language pairs. NEs are extracted based on a hidden Markov model, and domain portability such as adaptive NE extraction is investigated. A context-based model is proposed to detect and recover NE speech recognition errors. In terms of NE translation, this framework also incorporates several NE translation phonetic and semantic features. These crosslingual similarity features are applied to NE alignment, phonetic projection and translation mining tasks using parallel and monolingual corpora. Additionally, a proposed cluster-specific name transliteration model generates more accurate person name translations based on their origins. This framework significantly improves NE extraction, translation and transliteration accuracies. When integrated into a machine translation system, it also boosts the machine translation quality.

1.3 Thesis Summary

We propose an effective language-independent framework to extract and translate NEs from text and speech. Within this framework, we develop various features and algorithms and apply them to text-based and speech-based NE extraction tasks and NE translation tasks in multiple language pairs. We achieve improved NE detection performance, reduced ASR character error rate and improved NE translation and transliteration accuracies. These techniques reduce the information loss from incorrect NE translation by 50%.

We adopt the hidden Markov model (HMM) as our baseline NE extraction system. With different resources and different problems to solve, we expand the baseline model in the following ways:

- We use bootstrapping technique to train a Chinese NE tagger from imperfectly labeled monolingual data, where NEs are automatically tagged using an existing NE tagger. Given enough noisy data, the bootstrapping

technique is able to correct certain inconsistent NE tagging errors, and a re-trained NE tagger achieves better NE extraction performance.

- Given an NE tagger trained from English broadcast news data, we want to extract NEs from the transcripts of meeting dialogues, the speech from a very different genre. We propose an adaptive NE extraction model to incorporate global and local context information, which significantly improves the NE extraction performance for meeting applications.

- To deal with errors in ASR hypotheses, we develop a context-based NE extraction model which recognizes NEs only from their context words, thus reduces its dependency on the actual misrecognized NE words. This approach, combined with speech recognition confidence measures and information retrieval techniques, reduces ASR errors from 18.2% to 18.0%, improves speech NE extraction accuracy from 69F to 73F and translation accuracy from 66F to 72F.

For NE translation, we develop several language-independent features to capture different similarity measures between source and target NE pairs, including

- Transliteration features representing their phonetic similarity;

- Word translation features characterizing their semantic similarity within NE;

- Context vector features describing the semantic similarities around the NE pair;

We incorporate these features into an NE translation framework to solve various NE translation problems in different language pairs (Chinese-English, Arabic-English and Hindi-English) with varying input data streams (text and speech) and resources (monolingual and bilingual):

- To align NE translation pairs from sentence-aligned bilingual corpora, where NEs have been independently labeled in both languages;

- To discover target NE translations for a given source NE from a sentence-aligned bilingual corpus, where only source language NEs are labeled;
- To search for a target NE translation from monolingual or non-parallel corpora, given a source NE and possibly its context information.

We observe significant improvements in both NE extraction and translation accuracies. When we combine the above NE translation strategies and apply them to machine translation tasks, we also improve the overall text translation quality.

We additionally propose a cluster-specific name transliteration framework. By grouping names from similar origins into one cluster and training cluster-specific transliteration and language models, we manage to dramatically reduce the name transliteration character error rates from 50% to 13%.

Finally we evaluate the effectiveness of the whole NE extraction and translation framework according to the NE information loss reduction. We propose an information-theoretic measure to estimate NE information loss from speech recognition and machine translation. Based on this measure, our NE extraction and translation techniques significantly reduce the NE information loss by 50%.

1.4 Thesis Contribution

This thesis work advances the research on NE extraction and translation in the following ways:

- We design a set of crosslingual, language-independent similarity features which characterize the pronunciation similarity, the semantic similarity and the contextual similarity between NE translations;
- We propose an NE translation framework that integrates the above features to solve various NE translation problems: bilingual NE alignment, NE projection and NE translation mining from non-parallel corpora. We successfully apply the framework in multiple language pairs: Chinese-English, Arabic-English and Hindi-English. We improve both NE trans-

lation accuracy and machine translation quality when integrating the NE translation into a statistical machine translation system.

- We develop a cluster-specific name transliteration framework and substantially improve name transliteration accuracy and reduce character error rate.

- We design an information-theoretic measure to estimate information loss from speech recognition and machine translation. Based on this measure, the proposed NE translation techniques significantly reduce the NE information loss by about 50%.

- We extend the HMM NE tagger with a context-based NE extraction model, aim to detect and correct speech NE recognition errors. This approach, combined with speech recognition confidence measures and information retrieval techniques, improves speech NE extraction and translation accuracy. To the author's knowledge, this is the first attempt towards speech NE translation.

- We adapt a broadcast news trained NE tagger on meeting transcripts, and significantly improve the NE extraction performance.

1.5 Thesis Structure

The rest of this thesis is structured as following:

In Chapter 2 we review the literatures on information extraction, machine translation and especially NE translation.

In Chapter 3 we introduce the basic framework of our NE extraction system, the hidden Markov model. We demonstrate its performances in multiple languages. We also address the model training using imperfectly labeled noisy data.

In Chapter 4 we present a set of crosslingual NE similarity features, including phonetic, semantic and contextual features.

In Chapter 5 we demonstrate how to apply them in several NE translation tasks: NE alignment within an NE-tagged sentence-aligned corpus; NE projection from a resource-rich language to a resource-scarce language; and NE translation mining from monolingual corpora.

In Chapter 6, we focus on NE extraction and translation from speech input. We propose an adaptive NE recognition method to extract NEs from meeting transcripts. We present a context-based NE extraction approach to detect and correct NE ASR errors. This approach, combined with speech recognition confidence measures and information retrieval techniques , demonstrates reduced ASR errors and improved speech NE translation accuracy.

In Chapter 7, we describe the cluster-specific name transliteration framework. We explain how we select name origin distance measures, origin clustering algorithm, name origin classifiers and the phrase-based name transliteration model.

In Chapter 8, we introduce an information-theoretic measure to estimate information loss from speech recognition and machine translation. We apply this metric to estimate the NE information loss from ASR and MT. We also evaluate our proposed techniques in terms of the relative information loss reduction.

Finally we conclude this thesis work with some conclusions and discussions.

Chapter 2

LITERATURE REVIEW

Named entity detection and translation stands between two research areas: information extraction (IE) and machine translation (MT). The IE module enables accurate detection of NEs, and the MT module ensure reliable translation of detected NEs. These two steps can be combined sequentially, where the output of the IE module is the input of the MT module. Alternatively, when NEs detection is less reliable, such as detecting NEs from speech recognition hypothesis, NE detection and translation can be tightly coupled: information from another language can be borrowed via information retrieval (IR) and MT module to help the NE detection.

In this chapter, we will first introduce related research on information extraction, especially on named entity detection, then we will give an overview on machine translation, especially on statistical machine translation. Finally, we will review relevant research on named entity translation.

2.1 Information Extraction

Broadly speaking, Information Extraction (IE) is to identify structured and user-desired information from large volumes of unstructured text. This involves retrieving relevant documents from text collections (domain-specific corpora, general domain corpora or the World Wide Web) and tagging particular relevant words or phrases in text. A narrowed definition of IE, as Grishman

(1997) described, is :

> the identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship.

Unlike information retrieval, IE involves shallow parsing of text, such as part-of-speech tagging and text chunking. Compared with deep understanding of full text where all the information in a text need to be explicitly represented, IE tasks usually have pre-specified semantic categories of output, their relationships and allowable fillers in a relation. Because IE can efficiently handle huge amount of text and achieve reasonably accurate information access, it is particularly appealing to information users overwhelmed by explosive information volumes.

Although people have realized the importance of automatically converting natural language text into structured data since 1950s, only in recent decades rapid progress have been made in information extraction research, particularly thanks to DARPA-funded TIPSTER program and MUC (Message Understand Conferences), and the ACE (Automatic Content Extraction) program under the TIDES (Translingual Information Detection, Extraction and Summarization) project. In the 1998 MUC-7 Information Extraction evaluation campaign, five tasks were designed:

- Named Entity (NE): mark each string that represents a person, organization, or location name, or a date or time stamp, or a currency or percentage figure.
- Template Element (TE): extract basic information related to organization, person and artifact entities.
- Template Relation (TR): extract relational information on employee_of, manufacture_of, and location_of relations.
- Scenario Template (ST): extract pre-specified event information and relate the event information to particular organization, person or artifact entities involved in the event.

- Coreference (CO): capture information on coreferring expressions: all mentions of a given entity, including those tagged in NE, TE tasks.

In 1999, the TIPSTER/MUC program was replaced by the ACE[1] program, which aims to automatic classify, filter and select information based on the content, i.e., the meaning conveyed by the source language data (natural text as well as derived text, such as ASR and OCR output). There are three primary ACE research objectives:

- Entity Detection and Tracking (EDT):
  - Identify five types of identities, Person(PER), Organization(ORG), Location(LOC), Geo-political entity(GPE), Facility(FAC);
  - classify them according to its types and subtypes;
  - detect all mentions of each entity within a document, whether they are named, nominal or pronominal.
- Relation Detection and Characterization (RDC): identify pre-defined relations between entities.
- Event Detection and Characterization (EDC): identify and characterize five types of events in which EDT entities participate.

In both the MUC IE tasks and the ACE IE tasks, named entity recognition is the core task that provides the base of all remaining tasks, and research on NE extraction has been the most intensively investigated.

2.2 Named Entity Recognition

Named entity recognition (NER), also known as NE extraction, NE detection, NE tagging or NE identification, is to recognize structured information, such as proper names (person, location and organization), time (date and time) and numerical values (currency and percentage) from natural language text. It is

[1] http://www.ldc.upenn.edu/Projects/ACE/intro.html

one of the first IE tasks to be researched. Many NER systems based on pattern-matching rules or statistical models achieved satisfactory performances on well-formed text. Based on the 1997 MUC-7/MET-2 evaluation, NE recognition systems have achieved 94% F score on English newswire text and 85%-91% on Chinese text, 87%-93% on Japanese text.

2.2.1 Rule-based Pattern Matching

Earlier NER systems were mainly based on pattern-matching grammars. Appelt *et al.* (1993) proposed the FASTUS system, an information extraction system based on finite state automata. To fill a pre-defined event template, FASTUS first identifies trigger words for each pattern of interest, then recognizes noun phrases, verb phrases and other critical word classes. After that, FASTUS recognizes patterns of events from phrases, identifies event incidents and merges them to form a complete template. It achieves 44% recall and 55% precision on an IE task from 100 texts, the state-of-the-art performance in 1993. In the New York University Proteus IE system (Grishman (1997)), names were identified by a set of patterns (regular expressions) that were represented in terms of part-of-speech, syntactic structures, orthographic features like capitalization and a dictionary of name list. Similar to the FASTUS, shallow syntactic analysis was also applied to extract noun and verb phrases, and for the following scenario pattern matching. Other rule-based IE systems include AutoSlog (Riloff (1996)) and RAPIER (Califf & Mooney (1997)) etc..

2.2.2 Statistical Models

In recent years, when large amount of text data became electronically available, several statistical methods for NE recognition have been developed. Their performance has caught up with and even outperformed those of the above rule-based IE systems.

Bikel *et al.* (1997) proposed a HMM-based NE recognition system, the first high-performance statistical NE recognition system. In this framework, NE classes are represented as hidden states, and words in a sentence are the out-

put from different states. NE recognition problem is cast as a state sequence decoding problem: find the state sequence that maximizes the probability of generating the word sequence in a given sentence. This approach requires NE labeled data to learn state transition probabilities and per state word generation probabilities. Easy to implement and with very good performances (S. Miller & the Annotation Group (1998)), this framework is also the choice for our NE recognition baseline system.

Another HMM NE tagger is proposed by Zhou & Su (2002), where a modified HMM chunk tagger is built to recognize and classify names. Internal (lexical, semantic and gazetteer information) and external (macro context) features are combined. Their system achieves very good performance on MUC-6 and MUC-7 test data.

Sekine *et al.* (1998) applied a decision tree to find and classify names in Japanese texts. The decision tree incorporates POS information, character type (Kanji, Hiragana, Katakana, alphabet, numbers or symbols etc.) and dictionary information to determine the probability that an NE starts or ends at a given position in the text.

Borthwick (1999) proposed a maximum entropy (MaxEnt) framework for NE recognition. NE tagging is considered as a sequence labelling problem, where multiple local and global, internal and external features are developed and combined in the maximum entropy framework to classify each word as one of the following: the beginning of an NE, a word inside an NE, the last word of an NE and the unique word in an NE. These features include:

- binary features, such as ”Is the word a All-cap word?”;
- lexical features, local context words are compared with a vocabulary to record vocabulary indices;
- section features, ”Does the text appear in ”Date”,”Preamble” or ”Text” section in a given article?”;
- dictionary features, ”Does the word exist in a pre-compiled proper name list?”
- external system features, i.e., NE outputs from other systems;

- long range reference resolution features, i.e., partial names referring to the same entity.

Recently Lafferty *et al.* (2001) proposed the Conditional Random Fields (CRF) framework for sequence labelling problem, and McCallum & Li (2003) applied it to the NER task. CRFs are undirected graphical models with efficient procedures for complete, non-greedy finite-state inference and training. Same as the MaxEnt model, it can incorporate a wide array of features flexibly. For a given input sentence, CRF defines the conditional probability of a label sequence (sequence of NE states) based on the total probability over the state sequences. During training parameters can be efficiently learned with quasi-Newton methods. CRFs achieved 84.04% F-score on English text in the CoNLL 2003 NER evaluation.

Transformation-based Learning (Brill (1995)) is a error-driven machine learning technique which applies sequence of transformation rules to maximally decrease the number of errors from the initial classification. It has been successfully applied to POS tagging and NP chunking and word sense disambiguation problems, and they are further applied in the NER problem(Ngai & Florian (2001)).

In addition to above stand alone NE recognition algorithm, Collins (2001) applied boosting and voted perceptron to rerank the NE recognition hypotheses from MaxEnt tagger. Several NE classifiers are also combined based on boosting Carreras *et al.* (2002), stacking and voting (Florian *et al.* (2003) Wu *et al.* (2003)) algorithms to achieve better performance.

2.3 Machine Translation

Research on automatic language translation by a computerized system has been studied since 1950s, with various ups and downs. Machine translation (MT) approaches can be roughly classified into three layers: (1) Interlingua-based MT, (2) transfer-based MT and (3) direct MT, as shown in the pyramid diagram in Figure 2.1(borrowed from Dorr *et al.* (1998)). These layers differ in the depth of linguistic analysis.

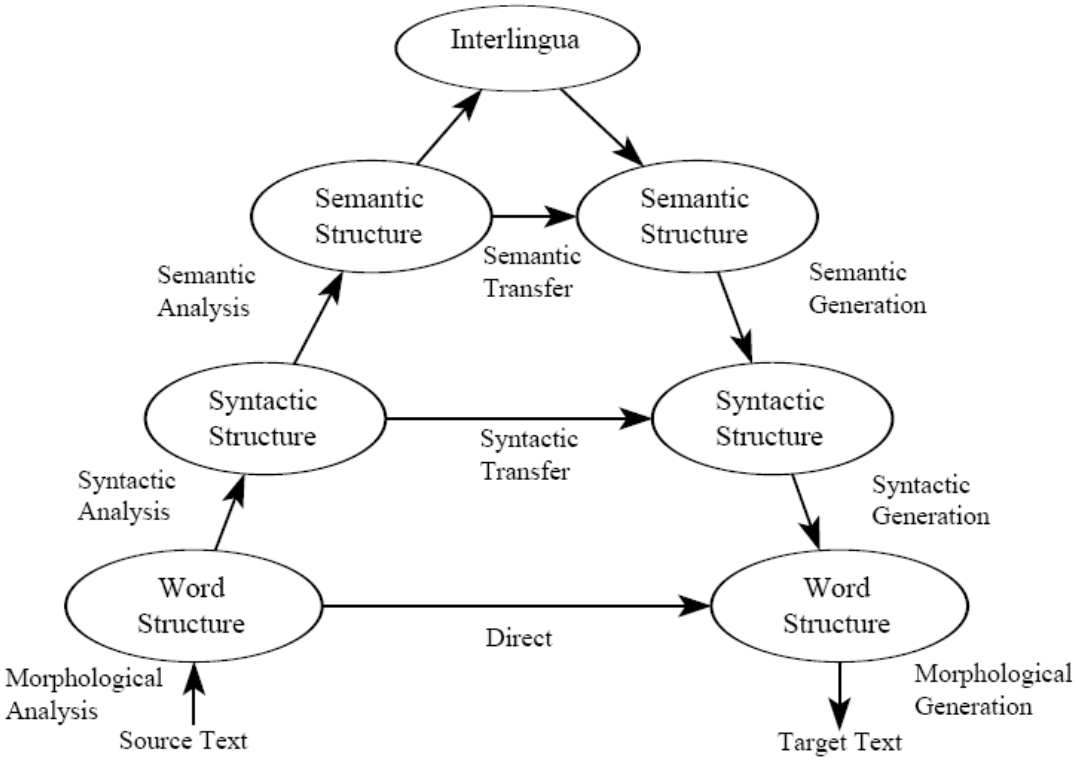


Figure 2.1: Types of MT Systems

2.3.1 Interlingua-based MT

Interlingua-based approaches analyze source language text, produce a language-independent semantic representation of the source sentence (called Interlingua), then generate the target language translation based on the semantic representation. This approach is efficient for translating between multiple language pairs, where Interlingua analysis and language generation are developed for each language only once. When we add in a new language's Interlingua, we naturally acquire the translation between this language and other languages, thus development cost is significantly reduced from $O(n^2)$ to $O(n)$ when there are n languages.

However, Interlingua development and maintenance require much human effort, especially when the application domain is getting broader. Therefore, Interlingua approach is only applied in specific domains. Typical Interlingua-based systems include Uchida (1985), Farwell & Wilks (1990), Mitamura *et al.* (1991) and Waibel *et al.* (1997).

2.3.2 Transfer-based MT

Transfer-based approaches stay between Interlingua and direct MT: syntactic transfer is closer to the direct models while semantic transfer is closer to the Interlingua models. The former applies syntactic analysis on the source language text, converting them into syntactic representations, which are further transformed into target language syntactic representations and target language sentences output. Lavie *et al.* (2003) demonstrated a transfer-based system for Hindi-English translation. Semantic transfer copes with many discourse particle and verb ambiguities that remain after syntactic/semantic analysis.

2.3.3 Direct MT

Direct machine translation models directly map source language sentences into word strings in the target language directly. The advantage and disadvantage are that they usually do not need sophisticated syntactic and semantic analysis, and often ignore meaningful linguistic knowledge. These methods

often require a reasonable amount of sentence-aligned bilingual text translations to train various translation models. Example-based MT (EBMT) and statistical MT (SMT) are typical approaches in this category. Since large amount of parallel corpora are electronically available for some language pairs nowadays, many corpora-based approaches, aka. data-driven MT, are proposed and actively investigated, and they achieved great success.

2.3.3.1 Example-based MT

EBMT, proposed by Makoto Nagao in 1981, is essentially translation by analogy. EBMT systems usually use a parallel corpus as the translation candidate pool, a thesaurus for semantic similarity computation, and a bilingual dictionary for word translation lookup, if needed. Given a source language sentence F, and a sentence-aligned parallel corpus, EBMT systems identify a set of source sentences T that are similar to F from the bilingual corpora, then translate F by selecting and combining best-match segments from the target translations of T. To find similar sentences in the parallel corpus, both "shallow" word-level alignment and "deep" parse tree alignments have been used. The EBMT approach has been combined with other MT techniques, such as rule-based Interlingua and statistical approaches. It can also be used as a component in a large MT system, such as multi-engine MT. An EBMT tutorial can be found at Brown (2002).

Brown (2000) introduced a generalized EBMT system, where text are converted into syntactic templates and strings are matched on the template level. The inexact match allows one word gap in the middle of a match. A bilingual dictionary and heuristic functions are used to identify word alignments within matched text. Appropriate semantic equivalence classes such as numbers, days of the week, city names and syntactic equivalence classes such as masculine nouns and first-person verbs are manually and automatically created and generalized. In addition, members of equivalence class can be equivalence classes, which enable the generation of a paired production-rule grammar. Other extensions including single word equivalence classes, grammar induction and word decomposition further improve the EBMT system's per-

formance.

2.3.3.2 Statistical MT

Statistical MT was first suggested by Warren Weaver in 1949, and was widely recognized thanks to Brown *et al.* (1990). In their SMT framework, the translation is a probabilistic sentence generation process under the source-channel model: to translate a source language (French) sentence f into a target language (English) sentence e, we assume that every English string e' is a possible translation of f with a probability $P(e'|f)$, and the French sentence is initially encoded as an English sentence e^* in the speaker's mind. We want to find the most likely translation, e^*, such that

$$e^* = \arg\max_{e'} P(e'|f). \tag{2.1}$$

with Bayes' rule,

$$P(e|f) = \frac{P(e)P(f|e)}{P(f)}. \tag{2.2}$$

Combining 2.1 and 2.2, we get the Fundamental Equation of Statistical Machine Translation:

$$e^* = \arg\max_{e} P(e)P(f|e). \tag{2.3}$$

$P(e)$ is a target language model, which characterizes the fluency of the generated English sentence, i.e., how likely e is a valid English sentence. $P(e)$ is usually a standard N-gram model, such as trigram language model.

$P(f|e)$ is a translation model, which characterizes the adequacy of e, i.e., how likely e and f carry the same information. This model is usually learned based on

- Word alignment: Brown *et al.* (1993) introduced a series of five statistical translation models featuring lexical translation probabilities, distortion probabilities (position-dependent word alignment) and fertility probabilities (the number of French words that a English word can generate).

Translations process involves generating the number of French words for each English word, generating their positions, and finally generating the actual French word identity. Vogel *et al.* (1996) proposed HMM-based word alignment which considers the alignment position of the previous word when predict the alignment position of the current word.

- Phrase-based alignment: This is currently the most active research topic in SMT. These phrases may not be linguistically well-defined, such as noun phrases, verb phrases, but they are able to capture more contextual information. Comparing with word-based alignment, phrase-based models enable many-to-many word alignment and avoid the difficult NULL word and one-to-many word alignment problems in IBM word alignment models. These phrase translation pairs can be extracted based on inverse transduction grammars (Wu (1997) and Zhao & Vogel (2003)), from initial word alignment paths (Och *et al.* (1999) and Vogel *et al.* (2003)), or estimated with various bilingual word statistics (Venugopal *et al.* (2003) and Zhang *et al.* (2003)). Recently phrase-based translation model have been achieving the best performances in several machine translation evaluations. New models are still proposed and investigated nowadays.

- Structure alignment: More sophisticated linguistic structures such as hierarchical phrases and constituents (noun phrase, verb phrase, etc..) in a parse tree are also proposed. The hierarchical phrases (Chiang (2005)) are phrases containing subphrases, which are context-free grammars learned from bilingual text without syntactic information. As a result, they can represent linguistically meaningful, longer sentence segments. Yamada & Knight (2001) presented a syntax-based statistical model, where a source language parse tree is transformed into a target language parse tree by stochastically applying reordering, insertion and translation at each node. There operations capture the difference between source and target languages, such as word order and case mark. Wu (1997) propose inversion transduction grammars for bilingual parsing of a parallel corpus. The algorithm synchronously constructs parse trees for parallel sentences, and produce word-level alignments within the sentence pair.

Effective and efficient MT evaluation methods also attribute to the recent rapid progress on MT. Traditionally, MT systems are evaluated by subjective judgment, relying on human beings to judge the adequacy and fluency of translation output, which is very expensive, inefficient and inaccurate. Objective evaluation metrics such as WER (word error rate), position-independent WER, sentence error rate do not characterize translation quality well. Papineni *et al.* (2002) proposed an automatic MT evaluation metric, Bleu, which scores a translation output based on its segment similarities to human reference translations. The similarity is measured in terms of the percentage of matched N-grams (N-1, 2,3, 4) between machine and human translation outputs.

2.3.4 Speech Translation

Although so far most MT research focuses on text translation, speech-to-speech translation has been investigated for more than a decade and is becoming a very active research topic. Among earlier speech translation research efforts, C-STAR and Verbmobil are representative projects. C-STAR (Consortium for Speech Translation Advanced Research)[1] was founded by Advanced Telecommunication Research-Interpreting Telecommunications Research Laboratories (ATR-ITL) in 1992, and gave the first public demo of phone translation between English, German, and Japanese. In 1993 the Verbmobil project (Wahlster (2000)) focused on portable systems for face-to-face English business negotiations in German and Japanese. These speech translation systems incorporate three modules: speech recognition module transcribing source language speech into source text, machine translation module translating source text into target text, and speech synthesis module converting target text into target language speech. Earlier speech translation systems used Interlingua for machine translation, thus application domains were often very specific: travel domains such as hotel reservation and flight ticket booking, or business negotiation applications such as the NESPOLE! project (Lavie *et al.* (2001)).

The recent success of SMT also expands the horizon of speech translation research. As large amount of open domain parallel text become available, sta-

[1] http://www.c-star.org/

tistical domain-unlimited machine translation systems can be trained. The statistical framework is able to naturally integrate speech recognition and machine translation modules, and tightly coupling of SR and MT is possible (Ney (1999)).

2.4 Named Entity Translation

Named entity translation and transliteration is a rather new research problem. As the amount of machine-readable text and speech data rapidly increase, it is more important to efficiently access desirable information from the huge data pool, even if it is presented in a foreign language. NE translation is important to many natural language processing tasks such as CLIR, MT and automatic knowledge discovery.

One of the earliest works on NE transliteration is Arbabi *et al.* (1994), where they presented a hybrid algorithm using neural networks and a knowledge-based system to transliterate vowelized Arabic into English.

Knight & Graehl (1997) proposed a generative model for Japanese-English back transliteration, where they presented a probabilistic framework of mapping from English words to English pronunciation, to Japanese pronunciation, to Japanese written format, and finally to the printed Japanese characters sequentially; Stalls & Knight (1998) expanded that model to Arabic-English transliteration, and Al-Onaizan & Knight (2002) combined phonetic-based model with spelling-based model for transliteration, generated NE translation candidates using bilingual dictionary, and ranked transliteration candidates by incorporating monolingual information retrieval results (candidates' occurrence frequency from web search).

Yarowsky & Ngai (2001) proposed the crosslingual induction of NE tagging based on word alignment information within French-English corpora. Given a sentence aligned parallel corpus, they align French and English words using IBM models. They automatically tagged NEs in the English sentences, and the corresponding French NEs can be identified with word alignment information. As a result, they obtained a NE-tagged French corpus, which can be used to train a French NE tagger. The effectiveness of this approach highly depends

on the initial English NE tagging accuracy and the word alignment accuracy.

In the context of spoken document retrieval, Meng *et al.* (2001) developed an English-Chinese NE transliteration technique using pronunciation lexicon and phonetic mapping rules. Given an English name, its English pronunciation phonemes are first generated by pronunciation lexicon lookup and letter-to-phoneme generation rules, then English phonemes are further converted into Chinese pronunciation phonemes (syllable initials and finals) using cross-lingual phonetic mapping. Finally, a Chinese phoneme lattice is constructed from which the most probable Chinese syllable sequence is found.

In Huang & Vogel (2002), we extract NE translations from aligned parallel corpus, where NEs are independently tagged for each language. Then we use a bootstrapping method to correct initial NE tagging errors and improve the NE translation accuracy. Moore (2003) proposed three progressively refined phrase translation models to learn the translations of NE phrases from parallel software manual corpus.

With the success of search engines such as Google, mining NE translations from web corpora becomes a new trend. A large number of web pages contain useful bilingual information, which may not be strictly parallel but contain both source NEs and their translation in another language. These web pages can be found using different information retrieval techniques, and NE translations can be extracted from retrieved web pages with several alignment features. Cheng *et al.* (2004) uses a source NE as the query and search only within target language web pages. Zhang & Vines (2004) search the whole web for all the web pages containing the source NE, then use format features such as parentheses to find their translations. Huang *et al.* (2005a) search for mixed language web pages using the source NE and semantically relevant target words as queries, then apply phonetic, semantic and frequency-distance features for high-recall, high-precision NE translation mining.

Chapter 3

NAMED ENTITY EXTRACTION FROM TEXT

NE extraction from text is a very important research area. It is also the basic technology for NE extraction from speech and NE translation. The text NE extraction quality directly affects the performances of speech NE extraction and NE translation. This problem has been thoroughly investigated, and the state-of-the-art performance is satisfactory.

3.1 HMM-based NE Extraction

One of the state-of-the-art NE extraction models is based on the Hidden Markov Model framework, as described in Bikel *et al.* (1997). In this framework several named entity classes (such as *PERSON*, *LOCATION* and *ORGANIZATION*) as well as one remaining class (*NOT_A_NAME*) are represented by four internal "hidden" states. This is a generative model, assuming that a given sentence is generated according to the following process:

- The current name class N is selected according to the previous word and its name class;
- The first word in a name class is generated according to the current and previous name classes;

- Each subsequent word in this name class is generated from a class dependent bigram model.

In the training procedure we want to estimate the following three probabilities:

1. $p_c(N|w_{-1}, N_{-1})$, the name class transition probability;
2. $p_f(w_1|N, N_{-1})$, the first word generation probability;
3. $p_b(w|w_{-1}, N)$, the class-dependent bigram word generation probabilities.

where N and N_{-1} represent the current and previous name classes respectively, w_1 represents the first word in the current name class, w represents the current word, and w_{-1} represents the previous word.

In the decoding process, the Viterbi decoding algorithm Viterbi (1967) is applied to find the name class sequence which maximizes the probability of generating words in the whole sentence. Suppose the sentence has L words,

$$\begin{aligned} \vec{N}^* &= argmax_{\vec{N}}\ P(\vec{W}, \vec{N}) && (3.1) \\ &= argmax_{\vec{N}}\ p(N_1) \times p(w_1|N_1) \times \\ &\quad \prod_{i=2}^{L} \tilde{P}(w_i, N_i|w_{i-1}, N_{i-1}), && (3.2) \end{aligned}$$

where $\vec{W}$ stands for word sequence $(w_1, w_2, \ldots, w_L)$, $\vec{N}$ denotes name class sequence $(N_1, N_2, \ldots, N_L)$, and $\tilde{P}(w_i, N_i|w_{i-1}, N_{i-1})$ represents the transition probability from w_{i-1} to w_i, assuming the class transition is from N_{i-1} to N_i.

When the transition is between different classes,

$$\begin{aligned} \tilde{P}(w_i, N_i|w_{i-1}, N_{i-1}) &= p(end|w_{i-1}, N_{i-1}) \times \\ p_c(N_i|w_{i-1}, N_{i-1}) &\times p_f(w_i|N_i, N_{i-1}). \end{aligned} \quad (3.3)$$

When the transition is within the same class, i.e., $N_i = N_{i-1}$,

$$\begin{aligned} \tilde{P}(w_i, N_i|w_{i-1}, N_{i-1}) &= \\ p(no\_end|w_{i-1}, N_{i-1}) &\times p_b(w_i|w_{i-1}, N_i). \end{aligned} \quad (3.4)$$

The $p(end|w_{i-1}, N_{i-1})$ and $p(no\_end|w_{i-1}, N_{i-1})$ denote the probability of exiting or remaining in the previous name class given the previous word. Here end and no_end are considered as delimiters generated in each name class, but they are not real words. Thus we do not calculate their word generation probabilities.

To smooth data sparseness problem, we take the following back-off paths for probability estimation.

- $p_c(N|w_{-1}, N_{-1}) \rightarrow p_c(N|N_{-1}) \rightarrow p_c(N) \rightarrow \frac{1}{number\_of\_name\_classes}$
- $p_f(w_1|N, N_{-1}) \rightarrow p_f(w_1|N) \rightarrow p_f(w_1) \rightarrow \frac{P(N)}{Vocabulary\_size}$
- $p_b(w|w_{-1}, N) \rightarrow p_b(w|N) \rightarrow \frac{P(N)}{Vocabulary\_size}$

When a specific model cannot be reliably estimated (e.g., the occurrence frequency of a certain event is too small), we interpolate it with the one-step-further more general model. We manually set the interpolations weights as 0.7 for the specific model and 0.3 for the general model.

As our goal is to extract NEs from multiple spoken languages, we do not exploit any case or punctuation information, which are very helpful features for English NE extraction, as used in Bikel *et al.* (1997).

This framework requires supervised learning for model training. In other words, the three probabilities p_c, p_f and p_b are learned based on frequency counting of specific events from labeled data, which are newswire and broadcast corpus with manually annotated NEs.

$$p_c(N|w_{-1}, N_{-1}) = \frac{C(w_{-1}, N_{-1}, N)}{C(w_{-1}, N_{-1})} \tag{3.5}$$

$$p_f(w_1|N, N_{-1}) = \frac{C(N, N_{-1}, w_1)}{C(N, N_{-1})} \tag{3.6}$$

$$p_b(w|w_{-1}, N) = \frac{C(w, w_{-1}, N)}{C(w_{-1}, N)} \tag{3.7}$$

Language	genre	# of words	# of NEs
Arabic	newswire text	26,325	2,383
	broadcast news	15,872	1,799
	total	42,197	4,182
Chinese	newswire text	36,689	2,410
	broadcast news	29,300	2,140
	total	65,989	4,550
English	newswire text	57,205	4,579
	broadcast news	33,479	2,440
	total	90,684	7,019

Table 3.1: ACE multilingual NE training data

3.2 Multilingual Named Entity Extraction

We apply the HMM NE extraction model to several languages, including Arabic, Chinese and English. For each language we have various amounts of text with manually annotated NEs. They are mainly from the Linguistic Data Consortium (LDC) TIDES Automatic Content Extraction (ACE) 2003 Multilingual Training Data[1]. Table 3.1 shows the sizes and genres of multiple language training data. The training data are from two genres, newswire text and broadcast news. Since the training data from each single genre are rather limited, and considering that these two genres share similar NE occurrence patterns, we combine them together and train a NE extraction system for each language. We apply langauge-specific preprocessing steps:

- For Arabic, we convert UTF-8 encoding text into Darwish, i.e., to romanize the original Arabic script.
- For Chinese, we apply word segmentation procedure with a word list with 43,959 entries.
- For English, we ignore case information and convert all the words into uppercase.

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T09

Test set	# of words	# of NEs	Precision	Recall	F-score
Arabic	3,242	438	86.63	69.03	76.83
Chinese	4,680	425	87.60	69.25	77.35
English	9,532	608	84.73	69.84	76.57

Table 3.2: Multilingual NE extraction performances

We select 90% of the annotated text as the training data, and 10% as the test data. The evaluation metrics for NE extraction include precision, recall and F-score. Precision (P) indicates the percentage of correctly extracted NEs among all the automatically extracted NEs, recall (R) denotes the percentage of correctly extracted NEs compared with all the manually annotated true NEs, and F-score is defined as

$$Fscore = \frac{2PR}{P+R}. \tag{3.8}$$

Table 3.2 shows the NE extraction results for three languages. Although we have different amount of training data for each language, the NE extraction performances on all the three languages are at the same range. We further evaluate the impact of different amount of training data on NE extraction performance. We have a relatively large amount of manually annotated English text from the Hub4 IE NE-tagged corpus. We select part of them (273K words) and all of them (859K words) to train two NE taggers. Because the Hub4 NE tagging guidelines are slightly different from the ACE annotation guidelines, we additionally select one unseen Hub4 NE-tagged document as the test set.

Table 3.3 shows the sizes of different training corpora and their performances. We train three English NE taggers with different training corpora, and evaluate them on the same English test set. We see that with a relatively small amount of training data (90K words with 6.4K NE examples for training), the 76.6% NE extraction performance is reasonably good. Double the NE training samples, the F-score is improved by more than an absolute 10%. Further triple the NE training samples, the F-score is still improved from 87.1% to 92.6%, but the improvement curve is getting diminishing. For Arabic, other than the 42K NE annotated data from ACE, we have additional 223K data from FBIS

Training corpus	# of words	# of NEs	Precision	Recall	F-score
ACE-English	90,684	6,411	84.73	69.84	76.57
Hub4 (part)	273,496	12,183	90.28	84.14	87.10
Hub4 (all)	859,347	34,315	93.23	91.40	92.31

Table 3.3: NE extraction performance vs. various amount of training data: English

Training corpus	# of words	# of NEs	Precision	Recall	F-score
ACE-Arabic	42,197	4,182	86.63	69.03	76.83
FBIS-Arabic	223,577	16,906	92.28	82.52	87.12

Table 3.4: NE extraction performance vs. various amount of training data: Arabic

corpus. We train another Arabic NE tagger with this larger amount of corpus, and observe similar improvement on NE extraction performance, as shown in Table 3.4.

3.3 Learning from Imperfectly Labelled Data

Table 3.3 and 3.4 show that the NE extraction performance highly depends on the amount of training data. For Chinese, we do not have enough manually annotated training data. However, we can use an off-the-shelf Chinese NE tagger, such as IdentiFinder[1], to automatically tag NEs from a Chinese text corpus, thus obtain a large amount of NE annotation corpus with automatic NE tagging errors. Using these imperfectly labeled data we can train our own Chinese NE tagger.

Why bother to train a NE tagger using noisy data? Why not just use the off-the-shelf NE tagger? Could the re-trained NE tagger perform better than the original NE tagger? One motivation is that the performance of our NE tagger can be carefully analyzed and new functions, such as NE tagging confidence measures, top-N decoding hypotheses output, and a context-based NE

[1]IdentiFinder is an HMM-based NE tagger trained with much more training data.

System	Precision	Recall	F-score
Manual	87.60	69.25	77.35
IdentiFinder	79.43	79.06	79.25
Re-trained	87.12	84.11	85.59

Table 3.5: NE extraction performance vs. various amount of training data: Chinese

extraction model (which can handle ASR NE errors for speech NE extraction, see 6.2.1) can be designed and implemented. Moreover, we are interested in the strength from bootstrapping. Running an imperfect NE tagger on large amount of text, we expect that some inconsistent NE tagging errors will be cancelled out, and the overall model reflects correct NE tagging results and unavoidable systematic tagging errors. These systematic errors can be identified according to confidence measures, and the most widely appeared errors can be selected and interactively corrected based on active learning (Tang *et al.* (2002)).

We run the IdentiFinder on a Chinese newswire corpus with 5.5 million words, and retrain our own NE tagger using these automatically labeled data. Table 3.5 shows the NE tagger performances trained from different amount of data: manually annotated 57K words ACE data (**Manual**), the IdentiFinder, trained from several hundred thousands words of manually annotated newswire data, and the NE tagger re-trained from 5.5M noisy newswire data (**Re-trained**), respectively. As expected, the IdentiFinder achieves higher performance than the manually trained model, possibly due to more training data. As for the re-trained tagger, even if it is trained with imperfectly labeled corpus, it is able to recover some inconsistent NE tagging errors and achieve the highest F-score, 85.59%.

3.4 Summary

Since NE extraction from text is the basic technology for both speech NE extraction and NE translation, we implement one of the most widely used NE

tagging model based on hidden Markov model. We evaluate its performances on different languages, including Arabic, Chinese and English. For each language, we also experiment with various amount of training data. Our results are comparable to state-of-the-art results. We develop a bootstrapping technique to train an NE tagger from imperfectly labeled data, and the retrained NE tagger achieves better performances than the original NE tagger.

Chapter 4

CROSSLINGUAL NAMED ENTITY SIMILARITY FEATURES

Different types of NEs are translated in different ways: while most person and location NEs are translated according to their pronunciations, most organization NEs are translated based on their meanings. One should take advantage of these similarity measures to translate different types of NEs. In addition, the context words with which an NE co-occurs indicate the semantic meaning of the NE, thus the similarity between source and target context words reflect the similarity between source and target NEs. In this chapter, we will present different features capturing the phonetic similarity, the semantic similarity and the context similarity between source and target NEs.

4.1 Surface String Transliteration

Transliteration is to translate a source NE into the target language based on their pronunciation similarities. Traditionally transliteration is made on the phoneme level, that is, a source name is first converted into a source language phoneme sequence, which is further translated into a target language phoneme sequence, and finally converted into a target name. In this process, converting source names into phonemes and converting target phonemes into names require name-phoneme mapping dictionaries for both source and target

languages. This is difficult for OOV name translation.

Considering that the written forms of person and location names often resemble their pronunciations, it is possible to discover NE translations through their written forms, i.e., the transliteration on the surface string form. Compared with the traditional phoneme transliteration method, the surface string transliteration does not require a pronunciation lexicon, which is especially an advantage for OOV name translations. For non-Latin languages such as Chinese and Arabic, an indirect surface string transliteration is feasible through a romanization process. This process maps each source language character into a Latin character (letter) or character sequence with similar pronunciations. For example, the Chinese translation of the English name "fitzwater" is "菲 茨 沃 特", whose romanization form, aka *pinyin*, is "fei ci wo te".

The mapping from source language characters into their romanization forms are usually deterministic, while the mapping between the romanization letters and English letters are probabilistic. When a collection of name transliteration pairs are available (e.g., the Chinese-English NE translation dictionary released by LDC[1], in 2003), the letter transliteration probabilities can be learned using an unsupervised learning algorithm, Expectation-Maximization (EM) (Dempster *et al.* (1977)). When such resource is unavailable, we can automatically extract name translation pairs either from a general domain word translation dictionary (e.g., the Chinese-English Translation Lexicon released by LDC in 2002), or from sentence-aligned parallel corpora (such as the Hindi-English parallel corpora described in section 5.2), base on the character transliteration model proposed here. Compared with manually compiled NE translation lists, automatic NE translation extraction can easily acquire large amount of bilingual NE pairs, although the translation accuracy is a little less than perfect.

4.1.1 Character Transliteration Model

The character transliteration model measures the pronunciation similarity between a source name and a target name. With this model we will be able to extract name transliteration pairs from a bilingual dictionary.

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T34

Given a general domain word translation dictionary D, we want to find Chinese-English NE translation pair (f^*_{ne}, e^*_{ne}), which has the highest joint transliteration probability,

$$\begin{aligned}(f^*_{ne}, e^*_{ne}) &= \arg\max_{(f,e)\in D} P_{trl}(f,e) \\ &= \arg\max_{(f,e)\in D} P_m(f)P_t(e|f) \end{aligned} \tag{4.1}$$

where f is the Chinese character sequence and e is the English word string. $P_m(f)$ is the probability of generating the character sequence of the Chinese NE, and $P_t(e|f)$ is the probability of transliterating the Chinese name into an English one. While $P_m(f)$ can be computed directly from a character-based language model trained with Chinese NEs,

$$P_m(f) = p(f_1)p(f_2|f_1)\prod_{i=1}^{m} p(f_i|f_{i-1}, f_{i-2}), \tag{4.2}$$

the transliteration model is computed in the following way. Suppose f has m characters. For $i = 1, 2, \ldots, m$, a Chinese character f_i is transliterated into an English letter string e_i through a pinyin syllable y_i . The generation process can be depicted as:

$$f_i \in f \rightarrow y_i \rightarrow e_i \in e$$

Note that English substrings are strictly monotone, i.e., there is no letter overlapping between e_i and e_{i-1}. The subscript i indicates that the substring is transliterated from f_i, and it is not necessarily the ith word/letter in e.

Let us assume each Chinese character is independently transliterated into an English letter string through its pinyin syllable. Considering that the mappings from Chinese characters to their pinyin syllables are mostly deterministic, i.e., $p(y_i|f_i) \approx 1$, then

$$\begin{aligned}P_t(e|f) &= \prod_{i=1}^{m} p(e_i|f_i) \\ &= \prod_{i=1}^{m} p(e_i|y_i)p(y_i|f_i). \end{aligned} \tag{4.3}$$

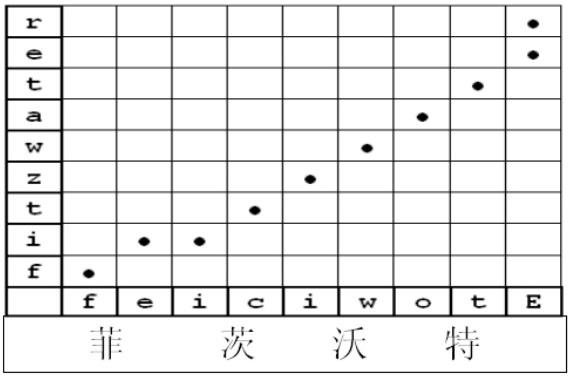


Figure 4.1: Surface string transliteration example

Suppose y_i is composed of m_i letters. For $j = 1, 2, \ldots, m_i$, the letter $y_{i,j}$ is aligned to the letter $e_{i,k}$, where the alignment is represented as $k = a_j$. Note that when two letters are aligned to one letter, usually that involves letter insertion and deletion. With the independence assumption,

$$p(e_i|y_i) = \prod_{j=1}^{m_i} p(e_{i,k}|y_{i,j}). \tag{4.4}$$

Thus the transliteration probability between a source name and a target name, $P_t(e|f)$, can be computed as

$$P_t(e|f) = \prod_{i=1}^{m} \left[p(y_i|f_i) \prod_{j=1}^{m_i} p(e_{i,k}|y_{i,j}, k = a_j \in A) \right] \tag{4.5}$$

where A is the character alignment path identified based on dynamic programming. Figure 4.1 shows an example of Chinese-English name transliteration. Given the letter alignment path (the sequence of black dots), the name transliteration probability is the product of letter transliteration probabilities over all the aligned letter pairs. We compute the joint transliteration probability for each word translation pair in the C-E translation dictionary, rank them and select top N entries as name transliteration pairs.

4.1.2 Parameter Estimation Using EM

Dynamic programming has been successfully applied in searching for the "optimal" alignment path between two strings, where "optimal" means the minimum accumulated editing cost between the aligned word/letter pairs. Usually the alignment cost is defined as 0 if the aligned character pairs are the same, or 1 if there exists insertion, deletion or substitution errors.

However, such a spelling-based binary cost function is not appropriate for pronunciation-based transliteration, where the phonetic similarity is more important than the orthographic similarity. Ideally the alignment cost between letters with similar pronunciations (e.g., "*c*" and "*k*" or "*p*" and "*b*") should be smaller. Considering that the letter transliteration probability reflects the alignment cost, we define the alignment cost between a English letter $e_{i,k}$ and a pinyin character $y_{i,j}$ as:

$$D(e_{i,k}, y_{i,j}) = -\log\, p(e_{i,k}, y_{i,j}). \tag{4.6}$$

This cost function is defined as the minus logarithm of the letter transliteration probabilities. Naturally, the transliteration cost between a source and a target NEs is defined as:

$$C_{translit} = \sum_{(k=a_j)\in A^*} D(e_{i,k}, y_{i,j}) \tag{4.7}$$

$$= -\sum_{(k=a_j)\in A^*} \log\, p(e_{i,k}, y_{i,j}). \tag{4.8}$$

A^* is the optimal letter alignment path, and the character transliteration probability is calculated from the character alignment frequency,

$$p(e_{i,k}|y_{i,j}) = \frac{C(e_{i,k}, y_{i,j})}{\sum_{e'} C(e', y_{i,j})} \tag{4.9}$$

where $C(e_{i,k}, y_{i,j})$ is the frequency that $e_{i,k}$ and $y_{i,j}$ is aligned.

The alignment frequency is collected from character alignment paths over all name transliteration pairs, and each path is identified using dynamic pro-

gramming under a given character alignment cost function. This can be solved with the EM algorithm, as following:

1. Initialization: set all character alignment costs to be binary.

$$D(e,y) = \begin{cases} 0 & \text{if } e = y; \\ 1 & \text{otherwise.} \end{cases}$$

2. E-step: Identify character alignment paths based on the current alignment cost function $D(e,y)$.

3. M-step: Estimate character alignment frequencies, update the character transliteration probabilities and alignment cost functions as in Formula 4.9.

4. Repeat step 2 and 3.

In practice, we first extract name transliteration pairs from a bilingual word translation dictionary using the binary cost function. Based on this initial imperfect NE list, the letter transliteration model and the character language model are trained, and used for the NE joint probability estimation (see formula 4.1, 4.2, and 4.5). In the following iterations, the letter transliteration probabilities and alignment cost functions are updated, NE transliteration pairs are re-selected according to the updated joint probabilities, and the translation and language models are re-trained using the cleaner NE pairs.

4.1.3 Experiment Results

The bilingual dictionary is the Chinese-English dictionary version 3.1 released by the LDC[1]. This dictionary contains 81,945 translation entries. Initially we apply the standard string editing distance with the binary cost function and select top-ranked 3,000 NE translation pairs for transliteration model training. Under the string editing distance function, Chinese names are always

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L27

ranked highly because the romanizations of Chinese characters are often correct translations. Selecting too few NE translation pairs, foreign name translations are not included and their transliteration patterns are not effectively represented. On the other hand, extracting too many NEs also increases more non-NE noises.

From this name transliteration list, the letter transliteration model and the Chinese character language model are trained. In each of the following iterations, we expand the amount of training data by adding additional 500 NE transliteration pairs. These name pairs are extracted from the bilingual dictionary according to their joint transliteration probabilities, which are calculated based on Formula 4.1 using the current transliteration and language models. Given more name transliteration pairs, these models are more accurately trained in each iteration. By estimating the precision of these extracted name pairs, we can evaluate each model's quality. This process continues until adding more NE pairs does not improve the name extraction accuracy any more, which happens at the 6th iteration where totally 5,500 ~6,000 NE transliterations are included.

We randomly select 10% of extracted translation pairs and evaluate the precision that they are actually true names. Figure 4.2 shows the precision curve after the 0 (baseline), 2nd, 4th and 6th iterations. "0/1 baseline" represents the result when using the binary cost function. "Ite N" illustrates the result after the Nth iteration. One can see that for well-trained models (those after the 4th iteration) the NE translation extraction precisions remain high for up to top 5000 entries. The NE precisions are consistently increased after each iteration. It is noticed that the most significant accuracy degradation happens at the 6000th name translation pairs. This indicates that most NE pairs in the dictionary have already been included, and adding more non-NE entries will "pollute" the transliteration and langauge models, thus the performance drops.

4.2 Word-to-Word Translation Model

The surface string transliteration model captures the pronunciation similarity between source and target NE pairs, which is an important feature for person

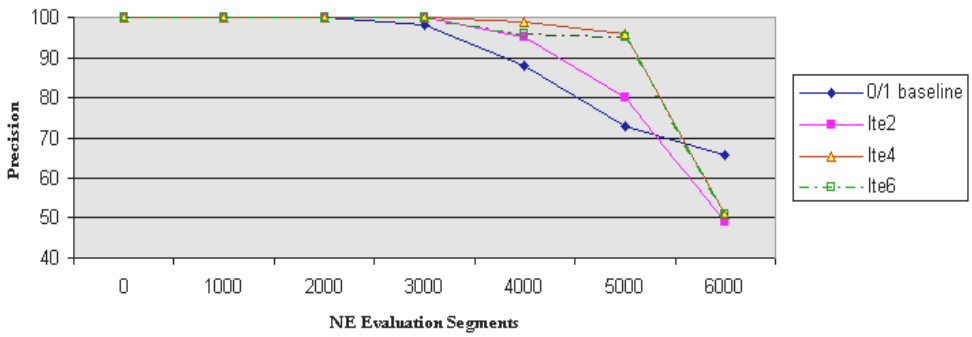


Figure 4.2: NE translation precisions with iteratively trained transliteration models.

and location NE translations. However, most organization NEs are translated based on their semantic meanings, which is most effectively characterized by a word–to-word translation model. This feature models the likelihood that the source and the target NEs are semantically equivalent, by calculating the translation probability between each source word and each target word. The word translation probabilities can be estimated either from a parallel corpus using various alignment models, such as IBM translation models (Brown *et al.* (1993)) and the HMM alignment model (Vogel *et al.* (1996)), or from a pre-compiled bilingual dictionary like the Chinese-English translation dictionary.

Assume an English NE e_{ne} has L English words, $e_1, e_2, \ldots, e_L$, and a Chinese NE f_{ne} has J Chinese words, $f_1, f_2, \ldots, f_J$. Suppose the word translation probability $p(f|e)$ is already acquired via IBM alignment based on a parallel corpus. The semantic translation probability of the source NE given the target NE is computed using the IBM model-1, as:

$$P_{trans}(f_{ne}|e_{ne}) = \frac{1}{L^J} \prod_{j=1}^{J} \sum_{l=1}^{L} p(f_j|e_l). \tag{4.10}$$

This model allows one source word f_j to be aligned to any target word e_l, while one target word can only be aligned to one source word. Considering that the semantic equivalence between the source and target NE translations should

be symmetric, we estimate both $P(f_{ne}|e_{ne})$ and $P(e_{ne}|f_{ne})$, and define the NE translation cost as:

$$\begin{aligned} C_{trans}(e_{ne}, f_{ne}) &\equiv C_{trans}(e_{ne}|f_{ne}) + C_{trans}(f_{ne}|e_{ne}) \\ &= -[\log P_{trans}(e_{ne}|f_{ne}) + \log P_{trans}(f_{ne}|e_{ne})] \end{aligned} \tag{4.11}$$

That is, the translation cost of a given NE pair (e_{ne}, f_{ne}) is the sum of the minus logarithm of the bidirectional conditional translation probabilities.

4.3 Context Vector Semantic Similarity

The surface string transliteration model is effective to find NE translation pairs with similar pronunciations and spellings, but it is less effective at identifying NE pairs with dissimilar pronunciations or discriminating different target NEs with similar pronunciations. On the other hand, NEs often occur within certain semantically related contexts, such as the title of a person, the neighbor area of a location. It is possible to combine the contextual semantic similarity with the phonetic similarity to improve the NE translation accuracy. For example, the following sentence pair talks about the same person: Although

Chn: 荷兰/dutch 驻/at 华/china 大使/ambassador 郝/hao 德/de 扬/yang 先生/Mr. 参观/visit 武汉/wuhan 。
Eng: ...among those present at today's signing ceremony were d . j . van houten , dutch ambassador to china...

the ambassador's name ("d. j. van houten") pronounces differently from its Chinese translations ("郝/hao 德/de 扬/yang"), the common context within which they both occur (although in different languages), "dutch ambassador to china", indicates the strong association between the source NE and the target NE. In this section, we want to measure the semantic similarity between the contexts of a source NE and a target NE.

Usually different context words have different correlation weights with regard to an NE, and the weights depend on the word types and distances to the NE. These weights reflect their discriminative power in predicting an NE's

meaning. The context word identities and their correlation weights can be represented by a context vector, and the semantic similarity between the source and target NE contexts can be computed in terms of their context vector semantic similarity. In the following, we will describe how to construct a context vector for a given NE, and how to calculate the crosslingual semantic similarity between a Chinese context vector and an English one.

4.3.1 Context Vector Selection

A context vector represents the words within a certain context of a given NE, and each word is associated with a weight reflecting its semantic significance to the NE. We select the most relevant words to construct a context vector, where the relevancy is characterized according to the word's Part-of-Speech (POS) and distance to the NE. We use Phi-square coefficients to represent the word identity weights, and these weights are atomic elements for the estimation of different POS tag weights and distance weights. The POS tag weights indicate the types of words that should be included in the context vector, and the distance weights indicate the optimal window length of the context vector.

As a variant of the χ^2, ϕ^2 is a measure of the correlation between two categorical variables. Its value ranges from 0 (no correlation between the two variables) to 1 (perfect correlation between them). In our situation, we want to measure the correlation between an NE and its context word. The NE-word semantic correlation coefficient can be defined as:

$$\phi(n,w) = \frac{o_{11}o_{22} - o_{12}o_{21}}{\sqrt{(o_{11}+o_{12})(o_{11}+o_{21})(o_{21}+o_{22})(o_{12}+o_{22})}}, \tag{4.12}$$

where n, w are the NE and its context word respectively. $o_{11}, o_{22}, o_{21}, o_{12}$ are the frequencies that they co-occur, neither occur, one occur and the other does not occur. The higher the coefficient is, the more likely that the NE and the context word are semantically correlated.

We estimate the NE-word correlation coefficients from an English newswire corpus. It is composed of 37M words from 188,755 documents. 380,641 unique English NEs are automatically tagged using our NE tagger, and the coefficients

are calculated for each (NE, word) pair as in Formula 4.12. Here the word is within [-20, 20] window around the NE. Table 4.1 shows the top 20 words having the highest coefficients with regard to the NE "Ehud Barak", the former Israel Prime Minister. Obviously words with high coefficients are mostly topic relevant words. This indicates that the ϕ^2 NE-word correlation coefficient is an effective measure to topical relevance. We estimate the semantic significance of a POS tag, like the noun, by summing the (NE, word) correlation coefficients over all (NE, word) pairs, and we factor in the probability that each word's POS matches the current POS tag. Then we normalize it over all POS tags.

Suppose an NE n has a context word w, whose POS tag is t. Under the empirical (NE, word) pair distribution $f(n, w)$, the weight of the POS tag t is defined as:

$$W(t) = \frac{\sum_{(n,w)} C(n,w)\phi(n,w)p(t|w)}{\sum_{(n',w')} C(n',w')\phi(n',w')} \tag{4.13}$$

where $C(n, w)$ is the frequency that (n, w) co-occur, $p(t|w)$ is the probability that word w has POS t. Figure 4.3 illustrates the normalized weights of different POS tags. One can observe that high weight POS tags are often content words (e.g., NNS (plural noun), VBG (verb gerund), NNP (proper noun), etc. [1]). This is in accordance with what we normally expect. One may consider using more general tag set such as "noun", "verb", "adjective" and "adverb" or even just "content words" vs. "functional words" to select context. However, such general POS tag set does not differentiate the most important content words (such as proper nouns) from less important content words (such as comparative adverb). We select the top 14 POS tags whose weights are larger than 0.03 as context vector weights. Only the context words with these POS tags are selected as context vector (CV) words.

Similar to the POS tag weights, the distance weights represent the semantic significance of CV words at different positions. Starting from a 20 words long window ranging from -10 (left 10 CV words) to 10 (right 10 CV words), the weight corresponding to words at location l can be estimated from the

[1] We use the Penn Treebank POS tag set as our POS tagger is trained from Treebank data.

Word	Correlation Coefficients
prime	0.0319
israeli	0.0222
minister	0.0194
caretaker	0.0160
yasser	0.0145
arafat	0.0115
leader	0.0069
palestinian	0.0063
outgoing	0.0060
al-shara	0.0047
clinton	0.0046
bill	0.0034
yatom	0.0032
david	0.0030
summit	0.0030
ariel	0.0029
camp	0.0028
likud	0.0028
sharon	0.0028
cabinet	0.0027

Table 4.1: Context words with high correlation coefficients for the NE ″Ehud Barak″

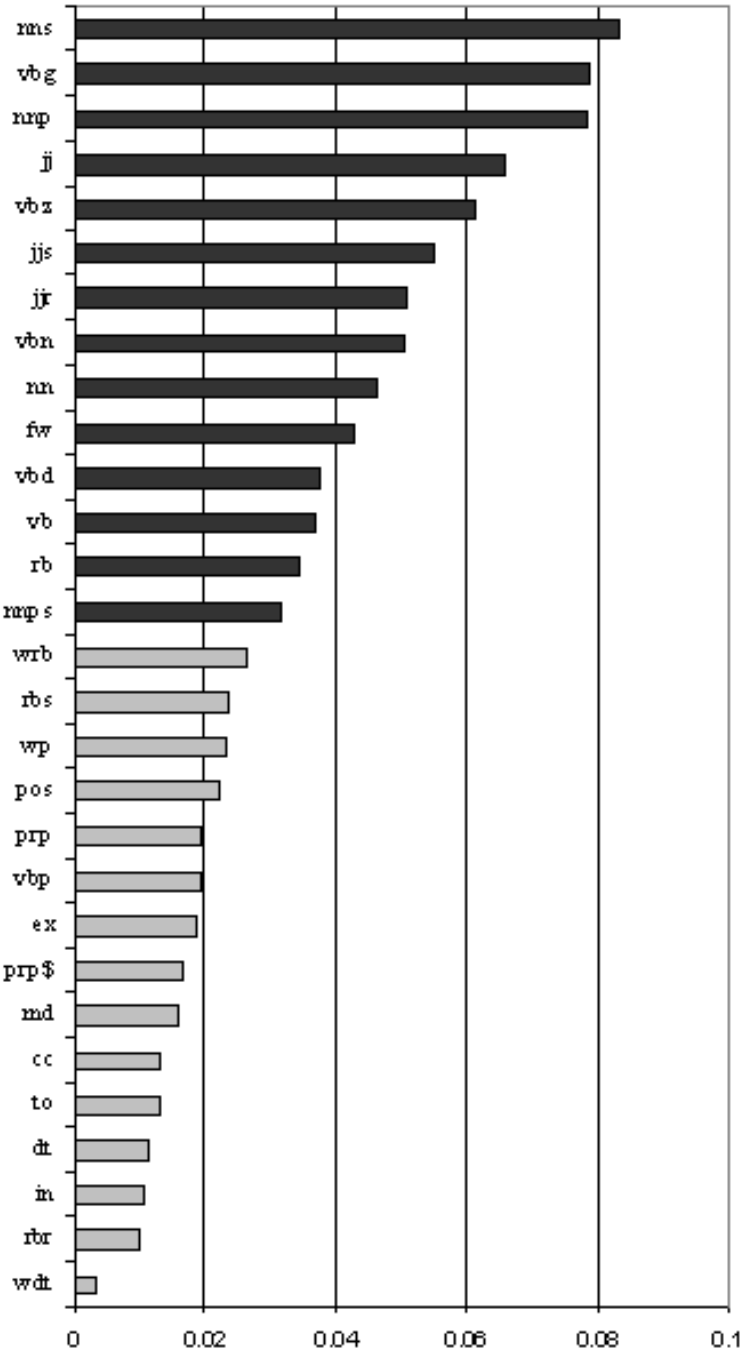


Figure 4.3: Normalized word POS weights

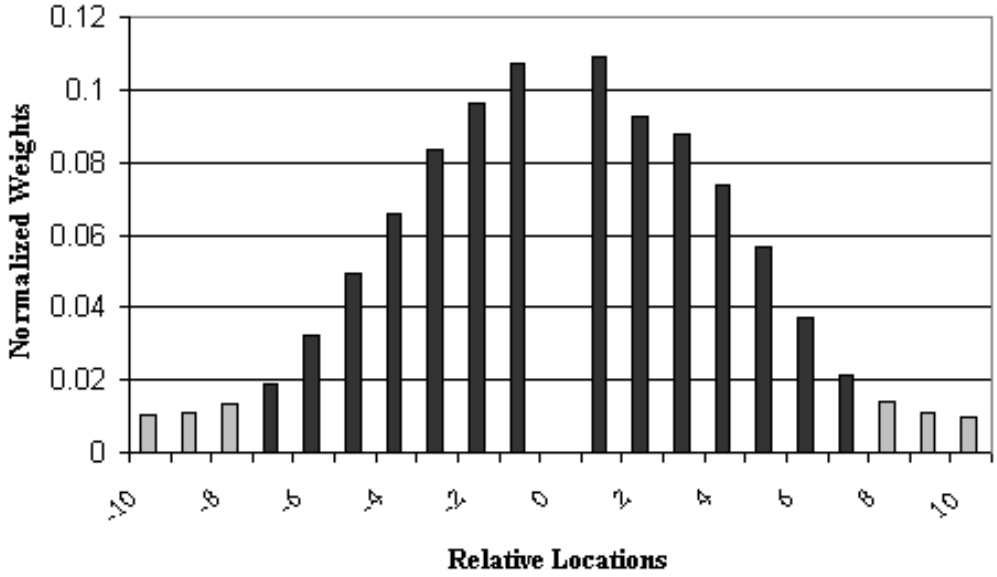


Figure 4.4: Normalized word location weights

(NE,word) coefficients:

$$W(l) = \frac{\sum_{(n,w)} C(n,w,l)\phi(n,w)}{\sum_{l'}\sum_{(n',w')} C(n',w',l')\phi(n',w')}, \tag{4.14}$$

where l is the location index, $l \in [-10,10], l \neq 0$. $C(n,w,l)$ is the frequency that word w occurs at the location l in the context vector of n. Figure 4.4 illustrates the normalized location weights, which looks like a Gaussian distribution: words within short distances have higher correlation coefficients to NEs. Notice that about 95% of weights are distributed within the [-7,7] window, so we select the context window length to be 14.

To summarize, the context vector of an NE is constructed from its left and right 7 content words, where "content words" are those whose POS tags are among the top 14 Content POS tag Set (CPS), as shown in Table 4.2. The complete Penn Treebank Tag Set are listed at Appendix A. The context vector is composed of both context word identities and their semantic weights:

$$V = \{(w, W(t,l)) | l \in [-7,7], l \neq 0, t \in CPS\}, \tag{4.15}$$

POS Tag	Meaning	Weights
NNS	common noun, plural	0.116
VBG	verb / gerund	0.110
NNP	proper noun	0.109
JJ	adjective	0.091
VBZ	verb/present, 3rd singular	0.085
JJS	adj. / superlative	0.076
JJR	adj. / comparative	0.071
VBN	verb/past particle	0.070
NN	common noun	0.064
FW	foreign word	0.059
VBD	verb/past	0.052
VB	verb, base form	0.051
RB	adverb	0.043

Table 4.2: English Context Vector POS Set and Weights

where $W(t,l) = W(t)W(l)$ is the product of their POS and location weights.

We apply similar processing on Chinese NE-tagged corpus. As a result, 9 out of 33 Chinese POS tags (different from the English POS tag set) are selected as a context word POS set, as shown in Table 4.3, and similarly we select the context window in the range of [-7, 7].

4.3.2 Semantic Similarity between Context Vectors

Given a bilingual NE pair (n_e, n_f) with their context vectors (v_e, v_f), the semantic similarity between the two vectors can be defined as the "mutual translation probability", which is the product of two conditional semantic translation probabilities,

$$S(v_e, v_f) = P(v_f|v_e)P(v_e|v_f) \tag{4.16}$$

where $P(v_e|v_f)$ is the probability that the source vector is "semantically translated" into the target vector. It is computed using a modified IBM translation

POS Tag	Meaning	Weights
NN	common noun, (except proper noun and temporal noun)	0.4248
VV	verb (excepting predicative adjective, ”be” and ”have”)	0.2445
NR	proper noun	0.1312
AD	adverb	0.0782
CD	cardinal number	0.0452
JJ	adjective	0.0332
M	measure word	0.0247
VA	verb (predicative adjective)	0.0183
NT	temporal noun	0.0139

Table 4.3: Chinese Context Vector POS Set and Weights

model-1 (Brown *et al.* (1993)),

$$P(v_e|v_f) = \frac{1}{I^J}\prod_{j=1}^{J}\left[W(t_j,l_j)\sum_{i=1}^{I}p(e_j|f_i)\right], \tag{4.17}$$

where I is the length of the source context vector and J is the length of the target context vector. $p(e|f)$ is word translation probabilities. $W(t,l)$ is the CV weights of the target word. Incorporating W ensures that important context words, such as the title of a person, should be translated correctly. $P(v_f|v_e)$ is estimated in the similar way.

The above three features represent various phonetic and semantic similarities between source and target NEs. In the next chapter, we will demonstrate how we apply these features to find NE translations in different scenarios.

4.4 Summary

In this chapter we propose a surface string transliteration model capturing the phonetic similarity between a source NE and a target NE. We also adopt an IBM word-to-word translation model measuring the semantic similarity between the source and target NEs. Finally we develop a context vector model characterizing the semantic similarity between the contexts of source and tar-

get NEs. We demonstrate how to select context words based on their POS and distance information, and how to calculate each context word's weight, which reflects the semantic significance of that word.

4. CROSSLINGUAL NAMED ENTITY SIMILARITY FEATURES

Chapter 5

NAMED ENTITY TRANSLATION FROM TEXT

In this section we will demonstrate how to apply various NE translation similarity features, introduced in Chapter 4, to translate NEs from text input stream. In particular, we will tackle NE translations in the following three scenarios:

- Given a sentence aligned parallel corpus, where NEs are automatically tagged in both languages independently, how to align these bilingual NEs? Additionally, is it possible to improve the NE extraction accuracy for each language given accurate NE alignments with another language?

- Given a sentence aligned parallel corpus as above, but NEs are automatically tagged only in one language (usually in English), how to find their translations in the other language? In other words, we attempt to project NE tagging across languages.

- Given a source NE and its context, such as the sentence or the document where this NE occurs, how to find the translation for this NE? This is a very typical problem in machine translation, where the source NEs are surrounded with contextual information, such as neighbor phrases, sentences or documents.

5.1 Named Entity Alignment from Parallel Corpus

We want to align NEs from a sentence aligned parallel corpus, for example, a corpus containing Chinese and English sentence pairs. NEs are automatically tagged for each language. We incorporate the transliteration and word-to-word translation features to measure the phonetic and semantic similarities between a source NE and a target one. We define the NE alignment cost as the sum of the two features, and NE pairs with minimum alignment cost are considered as correct translations. The aligned NE pairs can be used to construct a probabilistic NE translation dictionary, which is integrated into a statistical machine translation system to improve its translation quality.

One problem is that the bilingual sentences in the parallel corpus may not be strictly aligned. Since these sentences are automatically mined and aligned from comparable corpora, many sentence pairs may talk about the same events or topics, but may not be exact translations of each other. Some NEs occurring in a Chinese sentence may disappear in the English sentence. For example, an automatically aligned Chinese-English sentence pairs is : Here a Chinese NE,

Chn: 据/According to 马 其 顿/ Macedonia 新闻/news 社/agency 报道/report, 马 其 顿/macedonia 参加/join 该/the 部队/force 的 人数/people 约/about 为/is 150 人/troops .
Eng: Macedonia will send about 150 troops to join the force , the news agency said .

"马 其 顿/Macedonia 新闻/news 社/agency" is just simply translated as "the news agency", a non-NE English phrase. Thus it is impossible to align these unparallel NEs and non-NE phrases.

Even if the sentence pair is a strict translation and all the NEs in the source language are translated in the target languages, there exists automatic NE tagging errors: some NEs are either untagged (missing), partially tagged or over-tagged (false positive) with other words. These NE tagging errors also cause NE translation problems. For example, an automatically tagged Chinese sentence

> 记者 PER{王} 能 标 ORG{赞比亚 司法部} 长 PER{马兰} 博 12 日 在 此间 会见 LOC{中国} 监察 代表团

should be tagged as

> 记者 PER{王 能 标} LOC{赞比亚} 司法部 长 PER{马兰 博} 12 日 在 此间 会见 ORG{中国 监察 代表团}

To recover from these partial NE tagging errors, we apply an variable-length sliding window around the original tagged NEs. The window initially matches an automatically tagged NE, but both ends of the window are allowed to expand and shrink within a certain range. As a result, a set of candidate NEs are generated, e.g., @PER{王} , @PER{王 能} and @PER{王 能 标}, which is the correctly tagged NE. To measure the likelihood that generated NEs are true NEs, we use type-specific NE language model probabilities.

5.1.1 Type-specific NE Language Models

We assume that a bilingual NE translation pair (e_{ne}, f_{ne}) should have the same NE types, i.e., person names should be translated as person names, and location names should be translated as location names. Therefore we train word-based language models for each NE type (person, location and organization) in each language. The training data is collected from person/location/organization NEs in manually or automatically annotated corpora, where the NE_START and NE_END symbols are added as boundaries for each NE. We calculate the probability that both e_{ne} and f_{ne} are generated from the same NE type language models, and select the NE type that maximizes such probabilities (corresponding to the minimization of the minus logarithm probabilities). If an NE is misrecognized (false positive or partial matching with incorrect NE boundaries), the probability should be small under correct NE type language models.

We define the overall NE language model cost as

$$C_{lm}(f_{ne}, e_{ne}) = \min_{N} \left[-\log \tilde{P}(f_{ne}|N) - \log \tilde{P}(e_{ne}|N) \right]. \tag{5.1}$$

5.1.2 Multiple Feature Cost Minimization with Competitive Linking

Given the NE-tagged sentence pair, we apply the variable-length sliding window around the originally tagged NEs. We consider all the extracted source and target NEs as possible NE translation candidates. For each bilingual NE pair, we combine several feature functions via linear combination. The linear combination framework includes a phonetic feature, the *transliteration cost* $C_{translit}$, a semantic feature, the *translation cost* C_{trans} and a NE language model feature, the *LM cost* C_{lm}. For a NE pair (e_{ne}, f_{ne}), the multiple feature alignment cost is their linear combination:

$$C_{mfa}(f_{ne}, e_{ne}) = \lambda_1 C_{translit}(f_{ne}, e_{ne}) + \lambda_2 C_{trans}(f_{ne}, e_{ne}) + \lambda_3 C_{lm}(f_{ne}, e_{ne}), \quad (5.2)$$

where $C_{translit}$, C_{trans}, and C_{lm} are defined in formulae 4.7, 4.11 and 5.1, respectively. λ_1, λ_2 and λ_3 are their interpolation weights. We consider transliteration, translation and LM costs equally important. We selected some correctly aligned NE pairs and examined the value range of different alignment costs. We set the interpolation weights such that the weighted alignment costs are in the same range. Another scheme is to set the interpolation weights according to the type of aligned NE pairs. As most person names are phonetically transliterated, one may set the transliteration weight with higher value if the aligned NE pairs are person names. Similarly, one may set the translation weight with higher value when aligning organization names, as most organization names are often semantically translated. Since location names can be either phonetically or semantically translated, the three weights can be set equally.

Given a bilingual sentence pair containing multiple NEs in each language, we want to find the optimal NE alignment path A^* such that the sum of the NE pair alignment costs along A^* is minimum.

Mathematically, let $F = (f_{ne_1}, f_{ne_2}, \ldots, f_{ne_n})$ denote the set of n source NEs in a given source sentence, and let $E = (e_{ne_1}, e_{ne_2}, \ldots, e_{ne_m})$ denote the set of m target NEs in a given target sentence. The optimal NE alignment path A^*

satisfies

$$\begin{aligned} A^* &= argmin_A S(A) \\ &= argmin_A \sum_{(j=a_i)\in A} C_{mfa}(f_{ne_j}, e_{ne_i}). \end{aligned} \tag{5.3}$$

where sentence alignment cost $S(A)$ is the sum of NE alignment costs along an alignment path A. We adopt an algorithm similar to the competitive linking algorithm (Melamed (2000)) to find A^* :

1. Initialization: the source sentence F has n tagged NEs, and the target sentence E has m tagged NEs. T is an empty set to store aligned NE pairs;
2. Let N store all possible source and target NE combination pairs, with $m \times n$ entries in total. Compute the multiple feature alignment cost for each pair with Formula 5.2;
3. Sort NE pairs in N according to their alignment costs, and the pair with minimum alignment cost is on the top;
4. Move the topmost pair (f_{ne}, e_{ne}) from N to T ;
5. Remove all $(f_{ne}, \bullet)$ and $(\bullet, e_{ne})$ pairs from N to avoid alignment confliction. In other words, if a source NE is aligned with a target NE, it can not be aligned with any other target NEs. Vice versa;
6. Repeat Step 3, 4 and 5 until N is empty or the alignment cost is higher than a threshold. T stores all aligned NE pairs, and implicitly represents the optimal alignment path A^*.

Because bilingual sentence pairs may not be exact translation of each other, it is very likely that NEs appeared in one language do not have their translations in the other language. Therefore we did not penalize unaligned NEs, such as adding NE null alignment costs.

Note that this algorithm is a greedy search approximation. At each step it chooses locally an optimal alignment pair, the NE translation pair with minimum alignment cost, among all the unaligned pairs. It cannot guarantee the

global optimality. But empirically it often finds the alignment path with minimum or close to minimum sentence alignment cost.

We apply NE alignment over all the sentence pairs in a bilingual corpus. For each source NE, we store all aligned target NEs, together with their alignment frequencies in the whole corpus. The NE translation probability is calculated according to the relative alignment frequencies:

$$P_{align}(e_{ne}|f_{ne}) = \frac{C(f_{ne}, e_{ne})}{\sum_{e'_{ne}} C(f_{ne}, e'_{ne})}. \tag{5.4}$$

We construct an NE translation dictionary that includes source NEs, their possible target translations and the translation probabilities, $P_{align}(e_{ne}|f_{ne})$. Since the alignment is bi-directional, the above formula can also be used to estimate $P_{align}(f_{ne}|e_{ne})$.

5.1.3 Improving Named Entity Alignment

We evaluate the NE alignment performance on a set of Chinese-English sentence pairs. We randomly select 100 Chinese-English sentence pairs with 4950 Chinese words and 5646 English words. After manually annotating and aligning, this test set yields 357 NE translation pairs. These manually aligned named entities are used as the gold standard to evaluate the performance of automatic NE alignments. The evaluation metrics for NE alignment include precision, recall and F-score. Similar to the evaluation metric for NE extraction, precision measures the percentage of correct NE alignments over all the automatic NE alignments, recall measures the percentage of correct NE alignments over all the manual NE alignments.

Table 5.1 shows the precision, recall and F-score of NE alignment when we use different similarity feature functions. Using the word-to-word translation cost alone achieves 74.5% F-score. The reason is, this model effectively captures the semantic similarities between bilingual frequently occurring NEs, thus a majority of NEs can be reliably translated. The variable-length NE alignment window and NE LM costs are able to correct some automatic NE tagging errors, thus improve the NE alignment F-score by 3%. On top of that the

	Precision	Recall	F-score
C_{trans}	66.1%	85.5%	74.5%
$C_{trans} + C_{lm}$	69.7%	87.7%	77.7%
$C_{trans} + C_{lm} + C_{translit}$	73.8%	90.5%	81.3%
$ManualAnnotation$	91.3%	96.1%	93.7%

Table 5.1: Precision, recall and F-score of NE alignment using different similarity features

transliteration model effectively captures person name alignment, and additional 2.8% F-score improvement is achieved.

The last row shows the NE alignment accuracy when all the three feature functions are applied on manually annotated test data, where there is no NE tagging error. The significant improvement in F-score , from 81.3% to 93.7%, indicates that initial automatic NE tagging errors (83F for Chinese NE tagging and 87F for English NE tagging) remain the major cause of alignment errors. Figure 5.1 also illustrates some NE alignment examples from Chinese-English sentence pairs, where (*) indicates incorrect NE alignments.

5.1.4 Improving Machine Translation Quality

We construct an NE translation dictionary based on NE alignment frequencies collected from a Chinese-English sentence-aligned parallel corpus. This newswire corpus contains about 152K sentence pairs, with 6M English words and 5.5M Chinese words, from Xinhua News Agency, Penn Treebank data and subsampled FBIS data. After automatically NE tagging, 480K Chinese NEs and 340K English NEs are labeled. It is rather surprising to find so many extra Chinese NEs (30% of total Chinese NEs are not translated). Further analysis shows that this is mainly due to the inexact translation between Chinese and English sentences. Additionally, automatic NE tagging errors are also part of the reason.

We apply multi-feature NE alignment on each sentence pair, then collect NE alignment frequencies over all the sentence pairs. The more often two NEs are aligned, the more likely they are correct translations. So we add the

Example1:

对 LOC_C1(德国) 来 说 , 重要的 LOC_C2(亚洲) 市场 是 LOC_C3(日本) , LOC_C4(中国) LOC_C5(韩国) , LOC_C6(台湾) , LOC_C7(香港) 以及 LOC_C8(东盟) 等 国家 和 地区 , 其中 尤 以 向 LOC_C9(东盟) 国家 的 出口 增长 最 快 .
It noted that LOC_E3(Japan) , LOC_E4(China) , LOC_E5(South Korea) , LOC_E6a(China)'s LOC_E6b(Taiwan) , LOC_E7(Hong Kong) and member states of the ORG_E8a(Association of Southeast Asian Nations) (ORG_E8b(ASEAN)) are all major markets for German products , adding that German exports to ORG_E9(ASEAN) countries increased faster than in other markets . In 1994 , for example , German exports to LOC_E10(Malaysia) increased by 41 percent and to LOC_E11(Thailand) by 33.5 percent.

With translation model

LOC_C7(香港)	-----	LOC_E7(Hong Kong)
LOC_C5(韩国)	-----	LOC_E5(South Korea)
LOC_C6(台湾)	-----	LOC_E6b(Taiwan)
LOC_C3(日本)	-----	LOC_E3(Japan)
LOC_C4(中国)	-----	LOC_E4(China)
LOC_C9(东盟)	-----	ORG_E9(ASEAN)
LOC_C8(东盟)	-----	ORG_E8a(Association of Southeast Asian Nations)
(*)LOC_C2(亚洲)	-----	LOC_E6a(China)
(*)LOC_C1(德国)	----	LOC_E11(Thailand)

Example2:

ORG_C1(中 葡 联合 联络 小组) 第十 五次 全体会议 将于 今年 11 月10 日 至 13 日 在 LOC_C2(北京) 举行 。
the 15th plenary session of the sino-portuguese joint liaison group is scheduled to be held between november 10 and 13 this year in LOC_E2(beijing).

With translation model

LOC_C2(北京)	-----	LOC_E2(beijing)

With translation and language models

LOC_C2(北京)	-----	LOC_E2(beijing)
ORG_C1(中 葡 联合 联络 小组)	-----	ORG_(sino-portuguese joint liaison group)

Example3:

ORG_C1(外交部) 发言人 PER_C2(章) 启 月 今天 在 此间 宣布
ORG_E1(foreign ministry) spokeswoman PER_E2(zhang qiyue) announced here today

With translation and language models

ORG_C1(外交部)	-----	ORG_E1(foreign ministry)
PER_C2(章)	-----	PER_E2(zhang qiyue)

With translation, language and transliteration models

ORG_C1(外交部)	-----	ORG_E1(foreign ministry)
PER_C2(章 启 月)	-----	PER_E2(zhang qiyue)

Figure 5.1: Selected parallel sentences and extracted NE translations with different feature combination

logarithm of the alignment frequency on the multi-feature alignment cost, and construct the NE translation dictionary based on the updated NE alignment costs. After removing unreliable NE translation pairs whose alignment costs below a threshold (-20 in our experiment), we get 300K NE translation pairs. Removing duplicate entries, we have 71,848 unique NE translation pairs, with 25,721 unique Chinese NEs and 32,857 unique English NEs. On average, each Chinese NE has three candidate translations. The NE translation dictionary achieves 83% translation accuracy on automatic NE tagging sentence pairs, and 88% translation accuracy on sentences with manual NE annotations.

We integrate this dictionary into a statistical machine translation (SMT) system and evaluate it on the Chinese-English newswire translation task. The SMT system is based on weighted finite state transducers (Vogel *et al.* (2003)), where each transducer is a collection of bilingual translation pairs of words, phrases or NEs. In our experiment, three transducers are used in the translation system:

- A word-based transducer (LDC), which is essentially the LDC Chinese-English bilingual dictionary. Since this dictionary is manually compiled, it has very high accuracy.
- Phrase-to-phrase transducers (HMM), where the phrase pairs are extracted from the HMM Viterbi alignment path from each sentence pair in the same bilingual corpus.
- A NE transducer based on the NE translation dictionary.

The evaluation data is the newswire test data used in TIDES 2001 machine translation evaluation. It contains 993 Chinese sentences, 24,821 words. Automatic NE tagging yields 2,379 NEs with 3,597 words. Evaluation metrics are fully automatic, including Bleu (Papineni *et al.* (2002)) and NIST (Doddington (2002)) scores. These scores measure the precision or the information gains of matched N-grams between reference translations and the machine translation hypothesis. Table 5.2 shows the improvement on translation qualities with and without the NE transducer under various baseline systems. We find that the NE translation dictionary showed improvements in both cases. When we

	NIST	Bleu
LDC	6.01±0.1	22.04±0.5
LDC+NE	6.47±0.1	23.56±0.5
LDC+HMM	7.50±0.1	29.07±0.6
LDC+HMM+NE	7.57±0.1	29.22±0.6

Table 5.2: Improved translation quality by adding NE translations

only use the LDC dictionary in the baseline MT system, adding the NE transducer significantly increases the Bleu score by 2 points, which corresponds to 0.46 increase of NIST score. When the HMM-based phrase transducer is further added into the baseline system, the improvement from NE transducer is quite small, with 0.15 Bleu points and 0.07 on NIST score (not statistically significant). This is because both the phrase transducer and the NE transducer are trained from the same bilingual corpus, thus most of the information carried by the NE transducer has already been included in the phrase transducer. More improvement to MT quality is introduced by NE translation mining technique, which can access monolingual corpus to translate OOV NEs. Details are given in section 5.3.

5.2 Named Entity Translation Projection Across Language

Given a sentence-aligned parallel corpus where source and target languages have different amount of resources (annotated data, NLP tools etc.), for NEs in the resource-rich language we want to identify their translations in the resource-poor language. For example, in the Surprise Language Exercise (Oard (2003)), the Hindi to English (H-E) translation system was needed in a short period of time (a month). Although many NLP resources and tools, including high-performance NE taggers, are available for English, there are very limited resources and tools for Hindi. As a result, without a Hindi-English translation lexicon and a Hindi NE tagger, direct Hindi-English NE alignment is not possible. However, given automatically tagged English NEs, we can find their

Hindi translations based on their phonetic similarity.

5.2.1 Extracting NE Translation Pairs with Limited Resources

Yarowsky & Ngai (2001) proposed a method that projects NEs from English to another language using standard word alignment models (Brown *et al.* (1993)). However, the performance of this approach highly relies on the quality of word alignment. With limited amount of parallel corpus[1], the word alignment quality is not satisfactory. As some NEs may be incorrectly segmented as sequence of characters in a certain language, it is more difficult to align several NE characters in one language with one NE word in another language, due to word segmentation errors and alignment model limitations. Additionally, for machine translation purposes, this approach gives no additional benefits over phrase-to-phrase translation model, where the phrase pairs are also extracted based on word alignments from the bilingual corpus (Vogel *et al.* (2003)).

Considering that person and location names are often phonetically translated and their written forms resemble their pronunciations, it is possible to discover NE translation pairs through their written forms, i.e., through surface string transliteration. Compared with traditional phoneme transliteration methods, a surface string transliteration model does not require a pronunciation lexicon, which is an advantage especially for less frequently occurring names. As introduced in section 4.1, the surface string model measures phonetic similarity according to probabilistic string editing distances. Similar to the Chinese-English name transliteration, a romanization process is also required for Hindi-English transliteration. For example, a Hindi word is romanized as "kalakattaa", which is the translation of "Calcutta".

We automatically learn the transliteration model between Romanized Hindi and English letters, and apply this model to extract Hindi-English NE pairs from the sentence aligned parallel corpora. Given a sentence pair, for each tagged English person and location name we search for its Hindi correspondence in the Hindi sentence. The Hindi candidate NEs are composed of N consecutive words (with varying N), and we try to find the candidate NE with the

[1]The Hindi-English parallel corpus has several hundred thousands words.

minimum transliteration cost. Finally we construct an Hindi-English NE translation dictionary from the bilingual corpus. The Hindi-English transliteration model can be learned either directly from the parallel corpus, or adapted from an already learned Chinese-English transliteration model (see section 4.1). Because of the noise in the Hindi-English parallel corpus and the high quality Chinese-English alignment model baseline, the adapted model outperforms the directly learned model. Detailed experiment results are shown in section 5.2.3.

5.2.2 Adapting A Transliteration Model for Hindi NE Translation

Because of the difference in language pairs and encoding schemes, the following problems must be tackled before applying the Chinese-English transliteration model to Hindi-English NE translation:

- The Hindi sentences are encoded as Devanagari characters. A romanization tool based on code table lookup is applied to convert Devanagari characters into Roman letters.
- The transliteration model is originally trained on Chinese-English NE pairs. Because of different Hindi-English letter alignment patterns, model adaptation is required. In practice, the Chinese-English transliteration model is first applied to compute the Hindi-English transliteration cost, resulting in a list of Hindi-English NE pairs with minimum alignment cost. From those imperfect NE pairs, the Hindi-English transliteration model is re-trained and applied in the next round of NE pair extraction. After each iteration, the transliteration model are updated.
- Given that Hindi NE tagger is not available, it is impossible to extract Hindi-English NE pairs by "monolingual NE detection followed by bilingual NE alignment". On the other hand, Hindi NEs can be detected by projecting English NEs into Hindi, according to their phonetic similarity or transliteration cost, where the English NEs can be automatically detected using any existing NE tagger.

The following steps describe the procedure of Hindi-English NE translation extraction from aligned sentence pairs:

1. Convert UTF-8 encoded Hindi Devanagari characters into Roman letters;
2. Initialize the Hindi-English transliteration model using a Chinese-English transliteration model;
3. Automatically extract NEs from English sentences.
4. For each extracted English NE, we select the Romanized Hindi word or word sequences with minimum transliteration cost by applying a variable-length window sliding from the beginning to the end of the corresponding Hindi sentence.
5. Collect Hindi-English NE transliteration pairs over all the sentence pairs, and sort them according to their transliteration cost weighted by alignment frequencies. Pairs with high transliteration cost (less than -7) are removed;
6. Retrain the current Hindi-English string transliteration model with selected Hindi-English name transliteration pairs, and collect letter alignment frequencies. Based on that we update letter transliteration probabilities and transliteration model;
7. Repeat step 4 to step 6 until convergence or over-fitting are observed (see below);
8. Map the Romanized Hindi words back to their corresponding Devanagari Hindi characters.

Figure 5.2 illustrates how the transliteration model is initially trained from Chinese-English model, then adapted for Hindi-English NE translation extraction. Notice that this approach searches for an acoustically similar Hindi name for a given English person and location name, and the corpora to be searched are not necessarily strictly parallel. It can be applied to comparable and monolingual corpora containing the Hindi name as well.

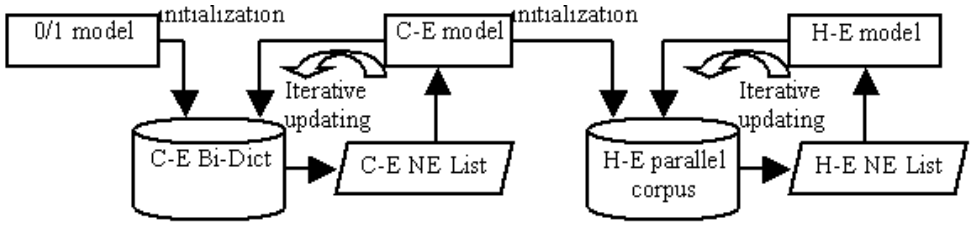


Figure 5.2: Iterative training and adaptation of the transliteration models.

5.2.3 Experiment Results

In our experiments, the Hindi-English parallel corpus is from the India Today news agency, with 10,096 sentence pairs, 223K Hindi words and 215K English words. Automatic NE tagging on the English side yields 2,451 English location NEs and 1,614 English person NEs, which in total generates 1,172 unique names.

We study several transliteration models to extract Hindi-English NE translation pairs from the above corpus. These models are 0/1 binary cost model (the standard string editing distance model), the original Chinese-English transliteration model and the adapted Hindi-English transliteration models (after the 1st and 2nd model update iterations). For each model, we select 220 bilingual NE pairs as the evaluation set, which rank as 1st-20th, 80th-99th, 180th-199th, ... 980th-999th in the sorted NE translation list. As in the sorted list, top ranked NE pairs always have higher translation accuracies than bottom ranked NE pairs, the selected NE pairs present a complete evaluation set for the transliteration model. We ask a native Hindi speaker to check the translation accuracy for selected NE pairs. Table 5.3 shows the NE translation accuracy using different transliteration models. One can see that the Chinese-English transliteration model outperforms the generic string alignment model by 7%, because the former model is able to capture Hindi-English pronunciation similarity to some degree. When we adapt it to the right Hindi-English transliteration model, we gain additional 4.6% improvement on NE translation accuracy. Further adaptation within the Hindi-English language pair still yields small but noticeable improvement. The Hindi-English transliteration model can be learned in

Alignment models	Precision
0/1 binary	79.1%
C-E	86.3%
H-E (1st iteration)	90.9%
H-E (2nd iteration)	91.8%

Table 5.3: H-E NE pairs translation accuracies using different alignment models

Iterations	Precision
0 binary	79.1%
1	85.9%
2	86.8%
3	88.2%
4	87.2%
5	86.8%

Table 5.4: Iterative NE translation accuracies starting with binary cost alignment model

two ways: adapted from the Chinese-English transliteration model or directly adapted from the binary cost string alignment model. In the second case the Hindi-English model is initialized with the binary cost, then is re-trained iteratively using extracted NE translation pairs. Table 5.4 shows the NE translation accuracies after each iteration. We notice a significant increase of transliteration accuracy after the first iteration, then a small but noticeable improvement after the second and third iterations, then slightly decreasing accuracy in subsequent iterations. The performance drop is possibly due to model overfitting. However, the best result achieved this way (88.2% in iteration 3) is not as good as the best result (91.8%) when we use the Chinese-English transliteration model for initialization. The reason is that the Chinese-English model already captures letter pronunciation similarities to some extent, thus it will provide more reliable baseline NE pairs for further re-training.

We also show some extracted Hindi-English NE pairs examples in Figure 5.3, together with their modified transliteration costs (minus logarithm of the alignment frequencies, thus the alignment cost can be negative). Note that the

Devanagari Hindi NE	Romanized Hindi NE	English NE	Alignment Cost
पाकस्तान	paakistaana	pakistan	-4.013
मुशर्रफ	musharrapha	musharraf	-0.125
कलकत्ता	kalakattaa	calcutta	-0.088
मार्गरेट अल्वा	maargareta alvaa	margaret alva	0.619
गंजम जलि	ga~jama jile	ganjam district	3.205
और मुंबई	aura mu~baii	sullied mumbai (*)	3.253

Figure 5.3: Extracted Hindi-English NE pairs

lower the weighted cost, the more accurate the transliteration. One can find similar spelling patterns between aligned Romanized Hindi NEs and English NEs, for both correct and incorrect (marked with "*") NE translation pairs. Since for each extracted English NE the proposed approach always searches for the best matching Hindi NE, its recall rate depends mostly on that of the English NE extraction. We also share the bilingual NE list within the TIDES Surprise Language Exercise community.

5.3 Search for Named Entity Translation

NE alignment from sentence-aligned parallel corpora usually achieves high translation accuracy. However, because of the limited amount of bilingual corpora, rarely occurring NEs may not be covered. Translating them correctly is more difficult. For example, when translating an ambassador's name, 郝 德 扬, in the following Chinese sentence:

Chn: 荷兰 驻 华 大使 郝 德 扬 先生 访问 武汉

Ref: netherlands' ambassador to china, van houten， visited wuhan

Hyp: netherlands ambassador hao germany hurls visited wuhan

Because the correct translation "van houten" was not included in the translation lexicon and parallel corpus, the ambassador's name is first wrongly segmented as a sequence of single characters, which then are inappropriately translated based on the semantic meanings of each single character, "郝/hao

德/germany 扬/hurls". On the other hand, if these rarely occurring NEs do appear in the parallel corpora, their translations may already be captured by various phrase translation tables. As a result, the overall MT translation performance is not improved much. This is especially true when NE translation dictionaries is trained from the same parallel corpus where other phrase translation tables are learned.

However, correct translations of these rarely occurring NEs may exist in a much larger target language monolingual corpus, which is much easier to obtain compared with bilingual corpora. It would be desirable to make use of relevant monolingual information to augment the limited coverage of the bilingual corpus.

To extract relevant monolingual information, we developed an approach combining information retrieval, NE extraction and machine translation techniques. Given a source (Chinese) NE together with the context it occurs (e.g., the document or sentence containing the NE), we want to find the target (English) documents containing the NE translation. After automatically tagging all the NEs in the retrieved English text, we can compare the source NE with each extracted English NE based on their phonetic and semantic similarity measures. Finally we can choose the best-matched English NE as the translation. Assuming that the documents containing the same NE share common topics (even if the texts are in different languages), we want to retrieve topic relevant English documents from a monolingual corpus using the translated Chinese contexts as the query. Figure 5.3 illustrates the overall architecture. We first automatically extract NEs in the source language (Chinese) document, for which we want to find the translations. The Chinese document is automatically translated into the target language (English) using our existing MT system, then we search an English monolingual corpus using the MT hypothesis as the query. Topic-relevant English documents are retrieved, and English NEs are automatically extracted and compared with the Chinese NEs. The best-matched pairs (in terms of phonetic and context semantic similarities) are considered as correct translations.

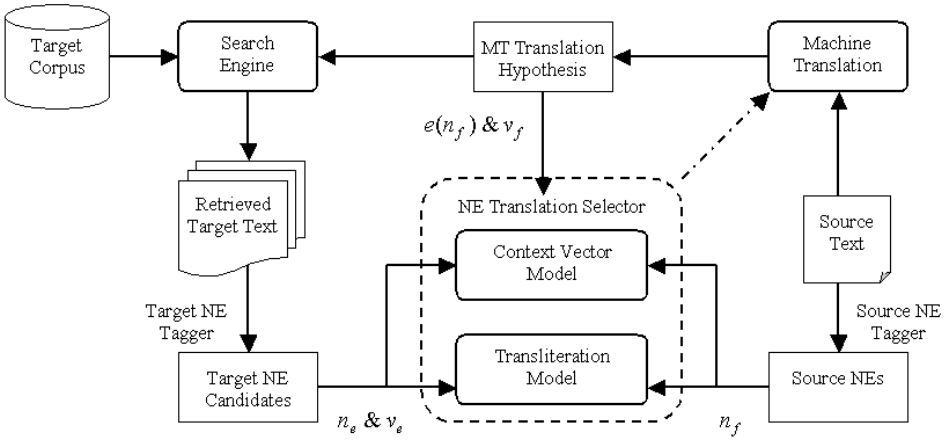


Figure 5.4: Overall architecture of NE translation

5.3.1 Query Generation

Given the source document, we can select the source language contexts in several ways: a few key phrases around the NE, the sentence containing the NE, or the whole document. Short queries usually include less irrelevant information, thus retrieve less irrelevant target documents. However it is crucial to correctly identify and translate these short contexts: key phrases such as content words and NEs. If the queries are not carefully selected or correctly translated, important topical information may be missing and retrieved target documents may not contain correct NE translations. On the other hand, correct contextual information is more likely to be included in longer queries, such as the translation of a sentence or the whole document. Due to the high risk of missing correct NE translations with short queries, we prefer to choose longer contexts, the whole document translation, as the query. In our current implementation, we use a statistical machine translation system to translate the Chinese document into English, filter out the unreliable NE translation from the hypothesis, and then select content words and feed them into any search engine, such as Google [1] or the Lemur Toolkit (Ogilvie & Callan (2001)).

[1] http://www.google.com

5.3.2 Corpus Indexing and Search Engine

Most commercial search engines can access and collect huge information from the World Wide Web, which is very helpful for rare NE translations (Zhang *et al.* (2005) and Huang *et al.* (2005b)). However for our research purposes we prefer a more flexible corpus indexing strategy that allows both sentence based and document based indexing such that we can select the most effective corpus indexing unit. Additionally, we are able to analyze the NE translation coverage of different corpora. So we start with building our own search engine using Lemur[1], a toolkit for language modeling and information retrieval.

The indexed corpus is composed of 963,478 English newswire documents from 10 years Xinhua news articles (1992-2002), which corresponds to over 7.3 million sentences and 200 million words. The indexing just follows standard procedures except stemming and stop word removal. The retrieval model is the widely used TF-IDF model. Given an English query (MT hypothesis of the source context), the search engine returns a ranked list of relevant English documents with relevance scores. We apply an English NE tagger on the retrieved documents, finding candidate English NEs and compare them with the source NE based on phonetic and context vector semantic similarities.

The toolkit allows us to select the appropriate indexing granularity: sentences, paragraphs or documents. We experiment with both sentence and document based query generation and corpus indexing. We want to select the appropriate indexing units such that the retrieved texts have the highest coverage of NE translations. From a set of Chinese newswire documents we selected 114 Chinese NEs, manually translated them and verified that the English translation is correct. As described above, we select appropriate context words from MT hypothesis as the query, search the English corpus and use the top 100 sentences or documents. We evaluated the NE translation coverage by counting how many correct NE translations can be found in the retrieved texts. It turned out that the document-based query/indexing covered 65.8% correct NE translations, while the sentence-based query/indexing has the coverage of 54.4%. One reason is that the returned top 100 documents contain several hundreds

[1] http://www.cs.cmu.edu/~lemur

sentences since each document contains multiple sentences, thus the more text the higher NE translation coverage. Another reason is, the sentence-level topic information is rather limited. If its translation hypothesis is not reliable, the generated query as well as the retrieved text may become irrelevant. In the following experiments we select document based query and indexing.

Another factor related to NE translation coverage is the corpus's time span. As the corpus is synchronous with source documents containing the source NEs, the coverage is higher. Since the above English corpus includes news articles up to year 2002, it covers 65-70% NE translations for 2001 and 2002 MT evaluation test set, but less coverage on 2003 and 2004 test set. We observe 20% coverage decrease per year. It is desirable to search the most updated data directly from the World Wide Web.

5.3.3 Combining Similarity Features for NE Translation

Given a Chinese NE n_f, we construct its context vector v_f using the approach described in section 4.3.1. Similarly, for each tagged NE n_e in the retrieved English text we construct an English context vector v_e. Their overall similarity score is defined as:

$$D(n_f, n_e) = \lambda_t C_{trl}(n_e, n_f) + \lambda_s S(v_f, v_e), \tag{5.5}$$

where C_{trl} is the NE pair's transliteration cost and S is the context vector semantic similarity. Because many rarely occurring NEs are person and location names, whose translation are mainly phonetic transliteration, we set the transliteration weight λ_t to be 0.75 and the context translation weight λ_s to be 0.25.

We select the most similar English NE (the one with minimum D), and consider it the correct translation of the source NE. In addition, we feed it back to the machine translation system such that the source NE can be correctly translated and the overall MT quality can be improved. Notice the dashed line connecting the NE Translation Selector and the Machine Translation module in Figure 5.4. Notice that better MT hypotheses (especially the correctly trans-

	Token(325)	Type(149)
	Correct (Percentage)	Correct (Percentage)
Bilingual Data	91 (28%)	41 (28%)
Translit	187 (57%)	71 (50%)
Translit+SCV	204 (68%)	83 (60%)

Table 5.5: OOV NE translation precision using bilingual and monolingual corpora

lated NEs) will help to formulate more appropriate queries, which will retrieve documents that are more relevant and improve NE translation quality again. This procedure can be an iterative process.

5.3.4 Evaluation

To evaluate the effectiveness of the proposed NE translation strategy, we test it on the Chinese-English machine translation task. The test data set is the NIST 2002 Machine Translation Evaluation test data. The test data is composed of 100 Chinese documents, 878 sentences, and 25,430 words. 2469 NEs are automatically tagged, among which PERSON, LOCATION and ORGANIZATION names roughly account for 20%, 60% and 20% respectively. Since most ORGANIZATION NEs are semantically translated word-by-word, given that we already have good word and phrase translation components in the baseline system, we will focus on PERSON and LOCATION NE translations, as they are often transliterated. We evaluate both NE translation accuracy as well as the overall machine translation quality, before and after the NE translations are incorporated into the baseline MT system.

5.3.4.1 Improving NE translation accuracy

Among 1,986 tagged PERSON and LOCATION NEs, 354 NEs are not covered by the 50K entries LDC bilingual dictionary, and we refer them as OOV NEs. After removing incorrectly tagged NEs, there are 325 correct Chinese NEs left which correspond to 149 unique NEs. Table 5.5 shows the type and token NE

	Total NEs(1986)	Total Words (3057)
	Correct (Percentage)	Correct (Percentage)
LDC	1331 (67%)	1331 (43%)
LDC+ALNE	1608 (81%)	1944 (64%)
LDC+ALNE+OLNE	1771 (89%)	2462 (80%)

Table 5.6: All NE word translation accuracy using aligned and retrieved NE pairs

translation precisions using bilingual and monolingual corpora. The bilingual data are 6 million words Chinese-English sentence-aligned bilingual corpus. We tag NEs in both language and apply the NE alignment approach (see section 5.1) to construct a 39K entry NE translation dictionary. This bilingual data still covers a relatively small amount of source NEs, and it achieves relatively low NE translation accuracy (28%).

We observe much higher translation accuracy when using much more monolingual information with different similarity models. If we only apply the transliteration model ("Translit"), the accuracy is improved from 28% to 50% for NE types, and from 28% to 57% for NE tokens. Additionally adding the context vector semantic features ("+SCV") further improves the accuracies by 10% (for type) and 11% (for token). Further error analysis indicates that 50% of errors are due to the limited coverage of the retrieved documents, i.e., correct NE translations are either not included in or not retrieved from the English corpus. If all the correct NE translations can be retrieved (we manually add the correct NE translations into the retrieved documents) , the translation accuracy is expected to be about 78%. One could add more data from the web to increase the corpus coverage. Table 5.6 shows the translation accuracy of all the LOCATION and PERSON NEs as well as their words in the whole test data. Because many rarely occuring NEs are composed of more than one word and hard to translte, the word translation accuracy is lower than the NE translation accuracy. Here we use three different resources:

- LDC: the 50K entry Chinese-English word translation dictionary from LDC;

	NIST	Bleu
Baseline	6.51±0.1	24.1±0.6
Baseline + ALNE	6.55±0.1	24.4±0.6
Baseline+ALNE+OLNE	6.82±0.1	25.2±0.6

Table 5.7: Improving small-track Chinese-English MT quality

	NIST	Bleu
Baseline	7.82±0.1	29.45±0.6
Baseline + ALNE	7.87±0.1	29.62±0.7
Baseline+ALNE+OLNE	7.98±0.1	30.09±0.7

Table 5.8: Improving large-track Chinese-English MT quality

- ALNE: an additional 39K NE translation dictionary aligned from 6 million words parallel corpus;
- OLNE: retrieved NE translations from relevant English documents.

We notice that the NE translation accuracy and the word translation accuracy can be improved by 14% and 21% when adding the aligned NE pairs, and retrieved NE translations further improve the accuracies by 8% and 16%. Overall, the proposed NE translation framework significantly improves the translation accuracy by an absolute value of 22%-37%.

5.3.4.2 Improving Machine Translation Quality

We integrate the retrieved NE translation pairs into our machine translation system, and test it in different translation tasks: the small data track and the large data track. They differ in the amount of bilingual resources that is allowed to use: for small data track we can only use 100K words for MT system training, while for large data track there is constraints on the amount of bilingual data. We use 6M words bilingual corpus for the large track MT system training. Again, the translation quality is evaluated in terms of NIST and Bleu scores. Table 5.7 and 5.8 show the translation quality when the MT system uses different NE translation strategies: the baseline system uses the LDC

	Small track		Large track	
	NIST	Bleu	NIST	Bleu
Baseline	5.62±0.2	21.6±1.0	6.85±0.2	27.3±1.4
Baseline + Translit	6.27±0.2	24.1±1.0	7.20±0.2	29.2±1.6
Baseline + Translit + SCV	6.36±0.2	24.5±1.1	7.28±0.2	29.5±1.5

Table 5.9: Improving C-E MT quality on selected 164 NE sentences

bilingual dictionary as well as several phrase transducers trained from some bilingual corpus, but no specific NE translation dictionary is used. "ALNE" is the NE alignment dictionary learned from the same bilingual corpus. It improves NIST and Bleu scores in both small and large data tracks, even though with a relatively small margin. This is because the NE dictionary and other transducers are learned from the same bilingual resource, and many NE translation information are already captured by other phrase transducers. "OLNE" refers to extracting NE translations from retrieved English documents. Since this approach is able to access and utilize information from much larger monolingual corpora, it can translate NEs which are not covered by the bilingual corpus, and bring the most improvement to machine translation qualities. Tables 5.7 and 5.8 show improved MT quality over all the test sentences, although these improvements are mainly from more accurate translations of rarely occurring NEs (frequent NEs have been reliably translated using the LDC dictionary). To accurately measure the effect of these infrequent NE translations, from the whole test set (878 Chinese sentences) we select 164 sentences containing the 325 infrequent NEs in Table 5.5. We translate this subset using the baseline MT system with and without the OLNE approach. The result is shown in Table 5.9. As we can see, this subset is more difficult to translate (lower Bleu/NIST scores compared with the whole test set, using the same baseline MT system) since it contains these rarely occurring NEs. However, the OLNE approach is very effective to translate these infrequent NEs and bring in significant improvement of MT qualities on this subset sentences. We apply similar NE translation techniques on the Arabic-English translation task. Similarly, we align the NE translation pairs from an Arabic-English parallel

	NIST	Bleu
Baseline	9.03±0.3	47.12±2.2
Baseline + ALNE	9.06±0.3	47.31±2.3
Baseline + OLNE	9.11±0.3	48.10±2.3
Baseline+ALNE+OLNE	9.17±0.3	48.71±2.3

Table 5.10: Improving Arabic-English MT quality

corpus. The parallel corpus includes 350K sentence pairs, 6 million Arabic words and 6.5 million English words. Initially 151K NE pairs are extracted from the corpus, which result in 46K unique Arabic-English NE pairs after removing duplicates. Top 20K NE pairs are selected as reliable translations, and are integrated into the existing machine translation system as the NE translation dictionary (ALNE). Additionally, we searched a monolingual English corpus for the translations of some rarely occurring source NEs in the test data, using the approach similar to the one in Figure 5.4. The result NE translation pairs are represented as OLNE. Table 5.10 shows the improvement on Arabic-English machine translation quality evaluated by the NIST and Bleu scores. The test data is 203 Arabic sentences from 25 documents. The baseline system combines several phrase transducers (the ISA transducer, the BiBr transducer and the HMM transducer) (Vogel *et al.* (2003)). We observed that the aligned NE translation pairs increase the NIST score by 0.03 over the baseline, while the retrieved NE translations (OLNE) additionally increase the NIST score by 0.08. The combined NE translation techniques increase both the NIST score (by 0.21) and the Bleu score (by 1.59). However, these improvements seem not statistically significant.

5.4 Summary

In this chapter we demonstrate three applications of text NE translation. We combine phonetic and semantic similarity features as well as NE language models to align source and target NEs from sentence-aligned Chinese-English corpus. In the parallel corpus NEs are automatically tagged for each language.

We observe improved NE alignment quality with different combination of alignment features. In the second task we work with a sentence-aligned Hindi-English corpus, where only English NEs are automatically tagged. We project English NEs into Hindi based on their pronunciation similarity, and achieve over 90% NE translation accuracy. Finally we attempt to translate rarely occurring NEs by searching for topic-relevant NEs from monolingual corpus. We query a pre-indexed English corpus with the MT hypothesis of the NE's context, extract candidate NEs from retrieved English documents and select the best match based on their phonetic and context vector semantic similarities. We achieve significant improvement on NE translation accuracy (28% to 68%). When we add the translated NEs into the MT system, we also improve the MT quality.

Chapter 6

SPEECH NAMED ENTITY EXTRACTION AND TRANSLATION

Past research mainly focused on NE extraction from well-formatted written text such as newspaper articles. These are grammatical sentences and carry useful information for NE extraction, such as punctuation, alphanumerical and word case information. Spoken languages such as broadcast news, presentations, meetings and casual conversations have rather different characteristics, which make NE extraction more difficult (Zechner (2001)). Extracting and translating NEs from speech present new challenges:

- Spoken language contains many ungrammatical segments and disfluencies, such as filled pause, repetition, repair and false start;
- Spoken language appears in very different genres and styles, while the styles of written texts are relatively consistent;
- Spoken language relies on automatic speech recognizers to transcribe them into text, and the ASR hypotheses contain errors from speech recognizers.

Most recognizer output is just a sequence of word tokens, without punctuation and case information to help NE extraction. Recently research on rich for-

mat transcription of speech has been conducted, where the speech transcripts include punctuations, cases, sentence boundaries and even disfluencies annotations. Even though, because of all the above mismatches, rules and statistical models learned from well-formed written text may not be suitable for speech NE extraction, and new techniques should be developed to handle these new challenges. Once NEs are extracted from the speech transcripts, we can similarly apply the text-based NE translation to translate them.

In this chapter we will focus on two speech NE extraction problems:

- NE extraction from manual transcripts of meetings, where we have a baseline NE tagger trained from broadcast news speech. and we want to adapt it for meeting applications.

- NE extraction from broadcast news ASR hypothesis, where we aim to develop an approach to detect and recover NE speech recognition errors, and translate them from Chinese into English.

6.1 Named Entity Extraction from Manual Transcript

In this task we are working on manual transcripts of meeting dialogues, and any off-the-shelf NE tagger can be directly applied to the error-free transcripts. When we apply a commercial NE tagger, IdentiFinder™, to broadcast news manual transcripts, we get an F-score of 91%, just a slight drop from 93% (F-score on newspaper articles) (Robinson *et al.* (1999)). This little degradation is because the good match between NE tagger training text and the test transcripts: both data are in the newswire domain (newspaper articles vs. broadcast news); and both data are with similar styles: compared with conversational speech, broadcast news data are more formal, more fluent and more similar to the written text. However, if we apply the same NE tagger to more informal speech (e.g., group meeting dialogues), because there are much more disfluencies and diverse topics in the meeting dialogues, the NE tagging model does not capture these new characteristics. As a result the NE extraction performance drops significantly with the F-score around 40-60%.

To deal with these mismatches between the NE tagging model and the test data, we propose an adaptive method for named entity extraction for meeting understanding. The baseline NE tagger is a statistical model trained from broadcast news data. It additionally makes use of global NE cache information and name list information from meeting profiles to adapt the baseline model to each specific meeting, and improves the NE extraction performance. This combination of unsupervised (NE cache model) and supervised (name list) information sources significantly outperforms the baseline model by 26% F-score. The performance is also comparable to that of a statistical model trained from a small amount of manually annotated meeting transcripts.

The statistical baseline model is the HMM model introduced in section 3. The model parameters are three kind of probabilities: NE class transition probability, first word generation probability and class-dependent bigram probability. The adaptation model uses the NE cache information as well as meeting profile information to re-estimate the class transition probabilities and word generation probabilities, and improves the performance of named entity extraction.

6.1.1 Global Context Adaptation Model: NE Cache Model

We assume that each meeting has certain coherent topics. A name (or more generally speaking, a word) can have more than one NE class (e.g., "Washington" could be a person name or a location name). Nevertheless, within a meeting, all the occurrences of this name tend to have a consistent NE class, and this type is related to the topic of this meeting. However, if we apply the baseline broadcast news NE tagger on meeting transcripts, due to various speech disfluencies and context mismatches, different instances of the same name can be tagged with different name types. To alleviate this problem, we estimate the average probability that a name has a certain NE class within this meeting context. Such average probability is supposed to be internally consistent. If we can reliably estimate such probabilities, we can correct some initial NE tagging errors. In practice, we use the baseline model to identify ambiguous NE words, whose several instances have different NE classes after the first-

pass NE tagging. We build a cache model to store the probabilities that each instance belongs to a certain NE class with different contexts, estimate their global name class probability, select the most likely NE class and then relabel these instances accordingly.

Formally, given a word w, the most likely name class over the whole meeting should satisfies

$$\begin{aligned} \hat{N(w)} &= \arg\max_N \ P(N|w) \\ &= \arg\max_N \ \prod_i P(N_i|w_i). \end{aligned} \tag{6.1}$$

$P(N|w)$ is the global name class probability for word w. Under the independence assumption, this is the product of all the local name class probabilities, $P(N_i|w_i)$. The local probability is computed as the linear interpolation of two kind of probabilities: *forward* name class probability $P(N_i|w_i, w_{i-1})$ and *backward* name class probability $P(N_i|w_i, w_{i+1})$. Intuitively, this model estimates the local name class probability based on its past and future contexts.

To calculate the forward probability,

$$P(N_i|w_i, w_{i-1}) = \frac{P(w_i, N_i|w_{i-1})}{P(w_i|w_{i-1})}, \tag{6.2}$$

where

$$\begin{aligned} P(w_i, N_i|w_{i-1}) &= \frac{P(w_i, N_i, w_{i-1})}{P(w_{i-1})} \\ &= \frac{\sum_{N'_{i-1}} P(w_i, N_i, w_{i-1}, N'_{i-1})}{P(w_{i-1})} \\ &= \frac{\sum_{N'_{i-1}} P(w_i, N_i|w_{i-1}, N'_{i-1})P(w_{i-1}, N'_{i-1})}{P(w_{i-1})} \\ &= \sum_{N'_{i-1}} \tilde{p}(w_i, N_i|w_{i-1}, N'_{i-1})p'(N'_{i-1}|w_{i-1}) \end{aligned} \tag{6.3}$$

and

$$P(w_i|w_{i-1}) = \sum_{N'_i} P(w_i, N'_i|w_{i-1}). \tag{6.4}$$

For backward probability,

$$\begin{aligned} P(N_i|w_i, w_{i+1}) &= \frac{P(w_{i+1}, w_i, N_i)}{P(w_{i+1}, w_i)} \\ &= \frac{P(w_{i+1}|w_i, N_i)P(w_i, N_i)}{P(w_{i+1}, w_i)} \\ &= \frac{[\sum_{N'_{i+1}} \tilde{p}(w_{i+1}, N'_{i+1}|w_i, N_i)]p'(N_i|w_i)}{P(w_{i+1}|w_i)} \end{aligned} \tag{6.5}$$

In the above formulae, $\tilde{p}$ is the word and class transition probability, which is calculated using class transition probabilities p_c, first word generation probabilities p_f and class-dependent bigram word generation probabilities p_b (see 3.3 and 3.4). $p'(N|w)$ is a *prior* probability that word w has name class N, regardless of its context. It is computed from the general domain broadcast news training data.

Thus, the local name class probability for word w at position i is the interpolation of the forward and backward probabilities, and w's global name class probability is the average probability over all the occurrences in the whole meeting. We select the most likely name class for w as the final NE type.

In summary, the whole NE tagging procedure is the following:

- Apply the baseline NE tagger on the test data;
- Identify ambiguous words, which have :
  - different NE class labels over the whole meeting after the first pass decoding;
  - low class assignment confidence, which is defined as the ratio between the top 2 class-dependent word generation probabilities;
- Re-estimate the global name class probability for ambiguous words ;

- Select the NE class which has the highest global name class probability;
- Relabel all the occurrences of the ambiguous word with the most likely class label.

6.1.2 Supervised Adaptation: Learning from Meeting Profile

The effectiveness of NE cache model depends on the quality of the first pass decoding. For a given word, after the first pass decoding if most of its occurrences are correctly tagged, the remaining few tagging errors can be corrected according to the global name class probability. However, if the word is difficult to tag (such as an OOV word), the baseline NE tagger will make lots of tagging errors. In this case, the global name class probability will be dominated by incorrect NE classes, and the cache model can even discard the few initially correct NE tags. On the other hand, meeting profiles usually contain the attendants' names, the topics to be discussed, or even a concise summary of the meeting. When such information is taken into account in the form of probabilistic name lists (e.g., person, location and organization name lists), the adaptation model has more accurate estimation of the context-independent prior name class probability $p'(N|w)$. As a result, the NE tagging performance will be improved.

In our current implementation, we only make use of meeting participants' names from meeting profiles, and we assign them to the *PERSON* name class with probability 0.9. The remaining probability mass, 0.1, is equally distributed among the rest name classes. These probabilities, $p'(N|w)$, are used in the computation of word generation probabilities.

6.1.3 Experiment Results

To evaluate the performance of the baseline model and the adaptation approach, we conduct several experiments. We train our baseline model using Hub4 NE-IE training data (52 broadcast news transcripts, about 260K words), and test it on one manual transcript of broadcast news episode (2318 words, 106 named entities). The result is the **Baseline**. We also run the IdentiFinder™(which

	BN	MT1	MT2
IdF	87.91	27.14	47.03
Baseline	88.35	37.93	60.37

Table 6.1: Baseline model on BN and MT data

	MT1	MT2
BL	37.93	60.37
BL+MP	50.07	65.65
BL+MP+CM	66.67	68.33

Table 6.2: Adaptation on baseline model for MT data I

is retrained with the same broadcast news training data) on the same test data, and obtain the **IdF** result. Finally, we run both NE taggers on two meeting manual transcripts, **MT1** (10554 words,137 named entities) and **MT2** (11146 words, 240 named entities). Table 1 summarizes the F-scores of these experiments.

As shown in Table 6.1, both the IdentiFinder and our NE tagger work reasonably well on broadcast news data. However, their performances drop considerably on the two meeting transcripts. This clearly shows the severe effect of model mismatch. Furthermore, we observe that their performances vary significantly from meeting to meeting. This is mainly due to the nature of different meetings.

Table 6.2 demonstrates the experiment result when different adaptation strategies are applied. **BL**, **MP** and **CM** represent the baseline model, the meeting profile model and the NE cache model respectively. When the meeting profile information is integrated into the baseline model, we observe improved NE extraction performance, especially for person names. Specifically, in **MT1** the meeting profile covers 45 instances of the 137 named entity instances, improving F-score by 32%, and in **MT2**, the meeting profile covers 24 of the 240 named entity instances, improving F-score by 8.7%, respectively. When cache model adaptation is further applied on **BL+MP**, most of the local NE tagging errors are corrected as long as the correct name classes have the highest global name class probabilities after the baseline tagging. This leads to additional improvement. Figure 6.1 illustrates the F-scores of different systems

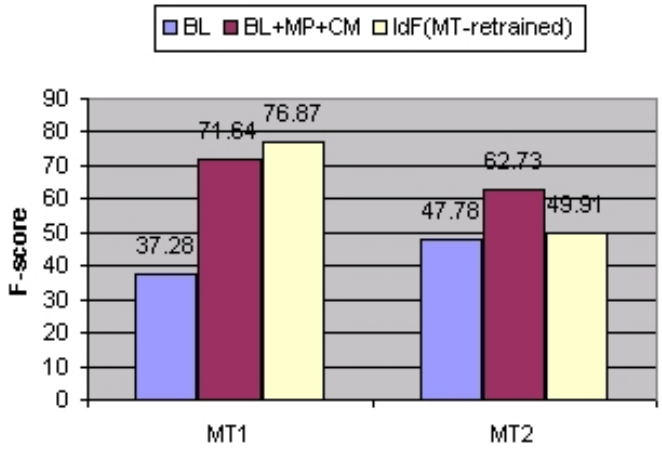


Figure 6.1: F-score comparison on ENAMEX class.

	BL	BL+MP+CM	Improvement	IdF(retrained)
MT1	37.93	66.67	75.77%	67.90
MT2	60.37	68.33	13.18%	61.11
MT3	47.76	54.99	15.13%	56.99
MT4	53.61	59.49	10.96%	63.87
MT5	53.87	58.23	8.09%	69.69
MT6	38.98	52.18	33.86%	66.10
MT7	60.33	61.13	1.32%	58.27
MT8	27.57	58.60	112.55%	68.32
Avg.	**47.55**	**59.95**	**26.07**%	**64.03**

Table 6.3: NE tagging adaptation for various meeting transcripts

on ENAMEX, which contains three name classes: LOCATION, ORGANIZATION and PERSON.

More experiment results are presented in Table 6.3. We find that the NE cache model plus meeting profile information is very effective in **MT1**, **MT6** and **MT8**, and less effective in **MT7**. In general, the proposed adaptive NE tagging approach increases the named entity extraction accuracy by an average of 26.07% over the baseline system.

In Table 6.3, we compare the proposed adaptive NE extraction model with the IdentiFinder system (re-trained with a small amount of meeting man-

ual transcripts (denoted as **IdF(retrained)**). Among all 8 meeting transcripts, which share similar topics and genres, 6 are used for training, and the remaining 2 are used as the test data. We make 4-fold cross validation experiments. In each "fold", the training set contains roughly the same number of words (about 90K), and covers most of the meeting participants' names in the test set, with the instance coverage from 58% to 100%. Trained with such domain-specific data, the NE tagger demonstrates much better performance than the baseline system which is trained from broadcast news data. We also find that the retrained NE tagger in general performs better than the NE adaptation model, which does not use the domain-specific meeting transcripts. However, in some applications it is possible that the adaptation model is comparable (as in **MT1**, **MT3**, **MT4**), even outperforms the retrained model(as in **MT2**, **MT7**).

We select some segments from the test data, and present them in Figure 6.2. We find that:

- The baseline NE tagger trained from broadcast news data can detect typical named entities, such as "*john*", from informal or ungrammatical context, but it also misrecognizes non-names which are often labeled as named entities in the broadcast news domain, e.g., "*channel one*";

- Additional information from meeting profiles, although quite limited in amount, can be very helpful because of its high relevance to the meeting;

- Even with the combination of the meeting profile information with the baseline probabilistic model, some names appearing in the meeting profile are still misrecognized, especially when the local context is highly informal, e.g.,"*...susi three nils four...*". Cache models can recover these local tagging errors.

- The cache model adaptation works best when correct name classes are assigned higher global name class probabilities. Otherwise — especially for OOV words— it isn't helpful, and even detrimental, as in the case of "*bbn*".

Example 1

BL: uh <b_enamex TYPE="PERSON">john <e_enamex> is on <b_enamex TYPE= "ORGANIZATION"> **channel one** <e_enamex> uh ***ty*** number two ***susi*** three ***nils*** four and myself five

BL+MP: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on <b_enamex TYPE= "ORGANIZATION"> **channel one** <e_enamex> uh <b_enamex TYPE="PERSON"> ***ty*** <e_enamex> number two ***susi*** three ***nils*** four and myself five

BL+MP+CM: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on **CHANNEL ONE** uh <b_enamex TYPE="PERSON"> ***TY*** <e_enamex> number two <b_enamex TYPE="PERSON"> ***SUSI*** <e_enamex> three <b_enamex TYPE="PERSON"> ***NILS*** <e_enamex> four and myself five

Reference: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on channel one uh <b_enamex TYPE="PERSON"> ty <e_enamex> number two <b_enamex TYPE="PERSON"> susi <e_enamex> three <b_enamex TYPE="PERSON"> nils <e_enamex> four and myself five

Example 2

BL/BL+MP: is *bbn* a name
<b_enamex TYPE="PERSON"> ***bbn*** <e_enamex> 'S the name of a company
yeah then it 'S just *bbn* without spaces

BL+MP+CM: is *bbn* a name
BBN 'S the name of a company
yeah then it 'S just *bbn* without spaces

Reference: is <b_enamex TYPE="ORGANIZATION"> bbn <e_namex> a name
<b_enamex TYPE="ORGANIZATION ">bbn <e_enamex> 'S the name of a company
yeah then it 'S just <b_enamex TYPE="ORGANIZATION "> bbn <e_enamex> without spaces

Figure 6.2: Some examples from test data .

6.1.4 Discussion: Information Retrieval Based on Meeting Profiles

Meeting profiles are usually short, and their limited topic-specific information does not offer big benefits. Nevertheless, some topic-related information in meeting profiles, e.g., scheduled topics, meeting summary, could be used as queries to retrieve topic relevant documents. Since most of the retrieved documents are written text, the baseline model obtains more accurate NE tagging results. Such additionally extracted named entities, together with their annotation confidence, can be integrated into the model for adaptation.

6.2 Named Entity Extraction from ASR Hypothesis

Although NE extraction from well-formatted text input has been intensively investigated and achieved satisfactory performance, NE extraction from speech remains under-explored. Kubala *et al.* (1998) and Miller *et al.* (2000) applied text-based NE tagger on the first best hypothesis of English broadcast news speech recognition systems, and they noticed that on average 1% WER costs 0.7 points of NE extraction F-score. Palmer *et al.* (2000) extracted NEs directly from word recognition lattices. Zhai *et al.* (2004) applied weighted voting on N-best hypotheses for Chinese broadcast news speech NE extraction. They also observed that NE extraction from Chinese speech seems more difficult than from English speech, which was also in line with our findings, as we observed on average 1.05 points of F-score drop for 1% WER.

Traditional NE recognition systems (Bikel *et al.* (1997), Grishman & Sundheim (1995) and Appelt *et al.* (1993)) developed statistical models or pattern-matching rules that explicitly model the distribution or patterns of actual NE words. However, when extracting NEs from speech, less frequently occurring NEs, especially those containing OOV words, are often misrecognized. If we apply the standard text-based NE extraction models on speech recognition hypothesis, these recognition errors also lead to NE extraction errors, such as insertion, deletion and substitution errors, as shown in the following examples (where we present each Chinese word, its pronunciation (pinyin) and its

meaning (in English):

Insertion

Ref: 这架/*zhejia/this* 小型/xiaoxing/small 客机/keji/airplane

Hyp:@*LOC*{浙江/*zhejiang/zhejiang*} 小型/xiaoxing/small 客机/keji/airplane

Deletion

Ref: 外交部长/waijiaobuzhang/foreign minister @*PER*{钱其琛/*qian qichen/qian qichen*} 指出/zhichu/pointed out

Hyp: 外交部长/waijiaobuzhang/foreign minister 前一阵/*qiznyizhen/a while ago* 指出/zhichu/pointed out

Substitution

Ref:@*PER*{阿拉法特/*alafate/Arafat*} 下令/xialing/order 成立/chengli/to set up

Hyp: @*LOC*{拉巴特/*labate/Rabat*} 下令/xialing/order 成立/chengli/to set up

Rather than tagging the erroneous ASR hypothesis using the text-based NE tagger, we distinguish reliably recognized words from unreliably recognized ones in the hypothesis using ASR confidence scores. We apply the standard text-based NE tagging only on the reliable hypothesis. For unreliably recognized words, we rely on their correctly recognized context information to predict candidate NE types and lengths. For example, if "foreign minister" is followed by a unsure word X, which is followed by "pointed out", then most likely X is a person name. We propose a context-based NE tagging model that is able to discover the partial information of a candidate NE within the HMM NE extraction framework. Querying with the reliable context information we retrieve topic-relevant documents from a large monolingual corpora, and extract topic relevant NEs from retrieved documents. We compare them with the candidate NEs, and select the best-fit NE or keep the original hypothesis based on a phonetic score, a semantic score and a language model score. To translate these NEs, we similarly retrieve target language relevant documents, extract target NEs and compare them with the source NE.

6.2.1 Context-based Named Entity Extraction

The context-based NE extraction attempts to identify partial information about NEs when their actual word identities are missing. In other words, we aim to detect the NE classes and positions from their contexts, as shown in the following sentence pair (one with the actual NE words and one with a replacement token:

> 外交部长/foreign minister @PER{钱其琛/qian qichen} 指出/pointed out
>
> 外交部长/foreign minister @PER{U_3} 指出/pointed out

Here U_3 represents a universal word token with 3 characters. We encode the word length (number of Chinese characters) in order to make use of duration information from a speech recognizer.

Similar to the HMM NE extraction framework, the context-based model estimates three generative probabilities (see Section 3) for a typical word. It additionally models these parameters for U_n, a universal word token with n characters but without the word identity. In particular, we estimate the following generative probabilities for U_n (n=1,2,3...7):

1. $p_c(N|U_n, N_{-1})$, the name class transition probability given U_n as the last word in the previous class;
2. $p_f(U_n|N, N_{-1})$, the first word generation probability when w_1 is U_n;
3. $p_b(w|w_{-1}, N)$, the class-dependent bigram word generation probabilities. We have to consider different scenarios based on whether w and/or w_{-1} are U_n or not.

Regarding to parameter estimation, we use the same training data for the text-based NE tagger, 5.6M Chinese words corpus where NEs are automatically tagged. Each word with n characters is considered as a word with identity w_n , and a universal word token without identity, U_n. We estimate the three generative probabilities for w_n based on frequency counting. We estimate the name class transition and word generation probabilities for U_n based on the frequency counting of w_n.

The NE class transition probability, the probability of generating the current NE class N when the previous word is a U_n and the previous NE class is N_{-1}, is:

$$\begin{aligned} p_c(N|U_n, N_{-1}) &= \frac{C(U_n, N_{-1}, N)}{C(U_n, N_{-1})} \\ &= \frac{\sum_{w_n} C(w_n, N_{-1}, N)}{\sum_{w_n} C(w_n, N_{-1})} \end{aligned} \tag{6.6}$$

where $C(w_n, N_{-1})$ is the frequency that the previous word w (length=n) and the previous NE class N_{-1} co-occur. And $C(w_n, N_{-1}, N)$ is the frequency that the current NE class N follows N_{-1} and w_n.

Similarly, the U_n's first word generation probability is estimated as:

$$\begin{aligned} p_f(U_n|N, N_{-1}) &= \frac{C(fw = U_n, N, N_{-1})}{C(N, N_{-1})} \\ &= \frac{\sum_{w_n} C(fw = w_n, N, N_{-1})}{\sum_{w'_n} C(fw = w'_n, N, N_{-1})} \end{aligned} \tag{6.7}$$

where $fw = U_n$ means the first word in the new NE class N can be converted into U_n (a word with n characters).

As to the word generation probability, we have to distinguish cases according to where the U_n appears:

$$\begin{aligned} p_b(w|U_n, N) &= \frac{C(U_n, w|N)}{C(U_n|N)} \\ &= \frac{\sum_{w_n} C(w_n, w|N)}{\sum_{w_n} C(w_n|N)} \end{aligned} \tag{6.8}$$

$$\begin{aligned} p_b(U_n|w, N) &= \frac{C(w, U_n|N)}{C(w|N)} \\ &= \frac{\sum_{w_n} C(w_n, w|N)}{C(w|N)} \end{aligned} \tag{6.9}$$

$$p_b(U_m|U_n, N) = \frac{C(U_n, U_m|N)}{C(U_n|N)} = \frac{\sum_{w_n}\sum_{w_m} C(w_n, w_m|N)}{\sum_{w_n} C(w_n|N)} \quad (6.10)$$

These probabilities can be learned from a Chinese corpus with NE annotations. Other than the word duration feature, we do not model other features such as case (not applicable for Chinese), digits or punctuation marks (not available from ASR output).

During decoding, given a sequence of words with their confidence scores, we first select unreliably recognized words, then map them into appropriate U_n tokens. With the context-based NE tagging model (Formulae 6.6 to 6.10, we use a similar Viterbi decoding algorithm to find the most likely NE class sequences. One can imagine the importance of word identity information for NE recognition. NE extraction without such information cannot achieve as good performance as the system with such information. However this approach demonstrates the contribution from NE context words and NE word length information, which are particularly useful for detecting and recovering NE speech recognition errors. Besides the typical one-best hypothesis, we can also generate N-best hypotheses, as presented in the experiment section. This will help increase the recall rate for the following candidate NE selection.

6.2.2 NE Extraction from ASR Hypothesis

We use speech recognition confidence scores to identify unreliably recognized words. Suppose an acoustic signal X is recognized as a word W, the posterior probability $P(W|X)$ is a good confidence measure. With Bayes rule,

$$P(W|X) = \frac{P(W)P(X|W)}{\sum_{W'} P(W')P(X|W')} \quad (6.11)$$

where theoretically W' is all the words in the vocabulary. Practically it is often the top N candidate words recognized from X, which can be generated from a word recognition lattice. We compute the confidence score for each word hypothesis, then convert words with low confidence scores into appro-

priate U_n's. We apply the context-based NE extraction model on unreliably recognized words, and the standard text-based NE extraction model on other words.

The following example shows that a person name (卡特曼/kataman) is misrecognized as "他/he 爆满/full" with low confidence scores (the number in the parentheses). Converting them into a universal word token U3, we are able to find the correct NE class and position in the converted hypothesis.

Reference words

美国/US 副/deputy 助理/assistant 国务卿/secretary of state 卡特曼/kataman 说/said

Hypothesis with confidence scores

美国/US(0.99) 副/deputy(0.99) 助理/assistant(0.99) 国务卿 /secretary of state(0.95) 他/ta/he(0.17) 爆满/baoman/full(0.08) 说/said (0.85)

Converting unreliable words into universal tokens

@LOC{美国}/US 副/deputy 助理/assistant 国务卿/secretary of state U3 说/said

Recognizing NE class and positions from converted hypo

@LOC{美国}副 助理 国务卿 @PER{U_3} 说

Note that speech recognizer may produce duration errors or word segmentation errors, where a three-character word is recognized as a four character word, or a two character word followed by a one character word. To deal with this problem we will allow the character length n to vary within a certain range, or we can select N-best NE recognition hypothesis to include correct NE positions.

6.2.3 Candidate NE Selection and Ranking

Given a hypothesized partial NE (with NE type and position information) as well as its context words, we want to find topic-relevant documents contain-

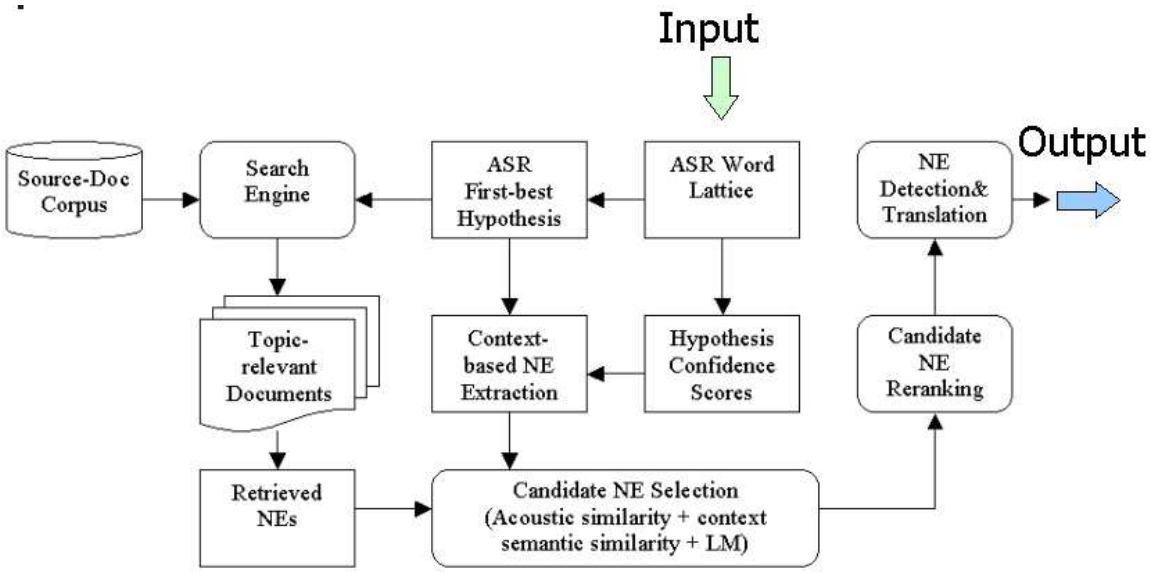


Figure 6.3: Overall architecture for ASR NE error detection and correction

ing the correct NE. Base on reliably recognized context words, we query a pre-indexed Chinese corpus and retrieve topic-relevant documents. NEs in the returned documents are automatically extracted and compared with the hypothesized ASR NE based on their phonetic and semantic similarities, and the best-fit retrieved NE is considered as the the correct NE. Figure 6.3 shows the overall architecture. The indexed corpus is composed of 63,092 Chinese documents from the Xinhua News Agency. The corpus has over 444K sentences and 22M words. We select top 100 retrieved documents and automatically tag NEs in these documents. We compare each extracted NE with the hypothesized NE words in the ASR hypothesis. The best-matched retrieved NE is selected to replace the universal word token, if the matching cost is below a predefined threshold. Otherwise we keep the original words.

Suppose U_n is the hypothesized NE extracted from the ASR hypothesis using the context-based model, and its corresponding word(s) hypothesis is f'. For each retrieved NE f_{ne_i}, we compare it with f' based on their phonetic and semantic similarities. We convert f_{ne_i} and f' into pinyin, and calculate their phonetic similarity according to the string alignment distance, where letters are aligned based on their pronunciation similarity, as described in 4.1. To cal-

culate the semantic similarity, we construct context vectors for both f_{ne_i} and f'. The vectors include the most informative context words around f_{ne_i} and f', as well as their weights. The context words are often NEs, nouns and verbs, while the weights are calculated based on the word's POS and distance to f and f_{ne_i}, as described in 4.3.1. The semantic similarities between two context vectors can be calculated in terms of WordNet, latent semantic analysis, or syntactic structures. In the current implementation, we consider the percentage of common words in both context vectors.

We first filter out unlikely NEs from the retrieved documents by comparing their phonetic and semantic similarities with the candidate NE f'. As a result, we only keep a small number of the most likely NEs. We fit them within f's left and right contexts, and use a language model to score each candidate hypothesis. Overall we combine the phonetic similarity, the semantic similarity and the language model score to rank all candidate NEs. The combination is a linear interpolation model. The candidate NE with the minimum overall cost is selected if the cost is above a certain threshold. This best-fit retrieved NE replaces the recognized word(s) f' in the ASR hypothesis, and is marked with the appropriate NE class from the relevant documents.

6.2.4 Named Entity Translation from Speech

Once NEs are identified from the recognition hypothesis, which may be either directly extracted from reliably recognized words or recovered from unreliably recognized words, translating them is straightforward. We just apply the text-based NE translation techniques (see section 5) to translate these extracted NEs. We either look up their translations from a pre-constructed bilingual NE dictionary, which is especially applicable to those frequently occurring NEs, or we can search for the correct NE translation from topic-relevant target documents, the same way as we search for the candidate source NEs.

However, the monolingual corpus to be searched must be in the target language (English in our experiment), and the query (context words) should be in English. To generate an English query, we use a statistical machine translation system to translate the source context into English. The phonetic similarity be-

tween a source NE f_{ne} and a target NE e_{ne} can be similarly calculated based on the surface string transliteration model, and the semantic similarity between the bilingual context vectors can be calculated with IBM models.

The tight coupling of NE speech recognition and translation is possible with the context-based NE extraction model. With the NE class and position information as well as the reliable context words, we can skip the source NE recognition process and find the translations of these hypothesized NEs directly from the target topic-relevant documents. For example, in the above example, the original transcript "美国 副 助理 国务卿 卡塔曼" is misrecognized as "美国 副 助理 国务卿 他 爆满", where only the person's name "卡塔曼/kataman" was misrecognized as "他 爆满/ta baoman". When we detect the type of the candidate NE is a person name, and we know the context words in English are "US Deputy Assistant Secretary", we can directly search the English corpus. From the retrieved relevant documents, we find the NE "Kartman" which is phonetically similar to "他 爆满/ta baoman" and semantically relevant to "US Deputy Assistant Secretary". Even if we do not recover the source NE "卡塔曼", we still translate it correctly. This is appealing to translate some English NEs whose Chinese translations are not included in the Chinese corpus.

6.2.5 Experiments

We first compare the context-based NE extraction model with the standard HMM NE extraction where NE word identity information is used. Secondly, we compare the NE extraction performance on ASR hypothesis, when applying the standard HMM NE extraction versus the context-based NE extraction with information retrieval techniques. Finally, we evaluate the improvement on NE translation quality.

Our test data is the Chinese broadcast news transcription used in the 1997 Hub4 Mandarin Chinese Evaluation. This test data includes 114 segments from 42 episodes, with 9176 words in total. For evaluation, we only consider three NE types: person names, location names and organization names. Their statistics are shown in Table 6.4.

	PER	LOC	ORG
number of NEs	211	478	209
number of Words	511	552	487

Table 6.4: Test data NE word distribution

6.2.5.1 Context-based NE Extraction from Manual Transcripts

We train both the standard HMM NE tagger and the context-based NE extraction model with the same data, 5.6M words of Chinese newswire text. This is the imperfectly labeled data whose NEs are not manually annotated, but automatically tagged using IdentiFinder (as described in section 3.3).

We first evaluate their performances on the manual transcript of the test data. As Table 6.5 shows, the standard HMM NE tagger obtains comparable performance as the IdentiFinder ™. To evaluate the context-based NE extraction model, we replace all the NE words in the test manual transcripts with appropriate universal word tokens Un. The results are shown in Table 6.5. As expected, recognizing NEs without word identity information significantly decreases the performance, reducing the F-score from 85.6 to 24.8. This indicates that the most important information for NE extraction is the actual NE word identity. Without this information and only using the context information, we can identify 14.8% NEs (the recall rate) in the top 1 hypothesis, even though the precision is reasonably high. That means if a word is recognized as an NE based on its context words, it is very likely a true NE. If we select the best decoded sentence from the top 100 hypotheses, the F-score is significantly increased from 24.8 to 50.3. Detailed analysis shows this is mainly due to the increase on recall rate (from 14.8% to 35.1%), while precision rate is also increased from 74.9% to 88.8%, which is even higher than that of the standard HMM NE tagger. So to increase the recall rate, we select candidate NEs from top 50 NE extraction hypotheses.

	Precision	Recall	F-score
IdentiFinder	86.63	84.41	85.50
Retrained NE Tagger	87.12	84.11	85.59
Context-Top1	74.91	14.85	24.78
Context-Top10	85.88	26.87	40.94
Context-Top50	88.12	32.78	47.78
Context-Top100	88.83	35.12	50.33

Table 6.5: NE extraction result on the manual transcript, using standard model and context-based model

	CER	Precision	Recall	F-score
Standard NE tagger on manual transcript	0	87.12	84.11	85.59
Stand NE tagger on ASR hypothesis	18.2	74.39	59.73	66.25
Context-based model on ASR hypothesis	18.0	76.96	67.19	71.74

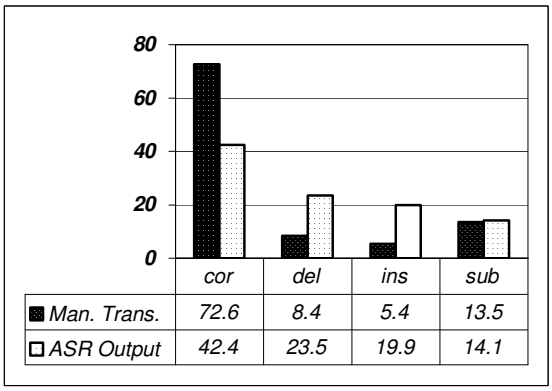
Table 6.6: NE exaction evaluation from speech input

6.2.5.2 NE Extraction from ASR Hypothesis

We manually annotated NEs from the manual speech transcripts. This is the gold standard file for NE extraction evaluation. We evaluate the NE extraction results under three scenarios:

- Apply the standard HMM NE tagger on the manual transcripts. This is similarly to the text-based NE tagging;
- Apply the standard HMM NE tagger on the ASR hypothesis, as done by most speech NE extraction systems;
- Apply the context-based NE extraction model to detect and recover speech NE recognition errors, as proposed in this section.

For each scenario, we evaluate both the speech recognition character error rate (CER) and the NE extraction precision, recall and F-score. The results are shown in Table 6.6. We find that the newswire-trained NE tagger works reasonably well on broadcast news manual transcripts, although we observe 2% F-score degradation, which are mainly due to genre differences. Running



	cor	del	ins	sub
Man. Trans.	72.6	8.4	5.4	13.5
ASR Output	42.4	23.5	19.9	14.1

Figure 6.4: Distribution of NE extraction errors in reference and ASR hypothesis

the same tagger directly on the speech recognition hypothesis, we get a much lower NE extraction quality with the F-score of 66.25. Detailed analysis in Figure 6.4 showed that with misrecognized NE words, both deletion and false insertion NE errors increased by 15%. When we apply the context-based NE extraction model and combine it with information retrieval approach to detect and correct NE recognition errors, we observe slightly reduction on ASR character error rate (18.2% to 18.0%). The NE extraction performance is improved prominently. The F-score is increased from 66.25 to 71.74. Detailed analysis indicates that this is mainly because of the significant increase of recall rate (from 59 to 67), thanks to the context-based NE extraction from top 50 NE decoding hypotheses.

6.2.5.3 Speech NE Translation from Reference and ASR Hypothesis

We translate both manually annotated as well as automatically extracted NEs using both bilingual and monolingual resources, as described in Chapter 5. The bilingual resource is a NE translation dictionary with 71K entries, automatically aligned from a 6M word sentence-aligned parallel corpus. The monolingual resource is a 200M word English corpus containing 10 years news articles from Xinhua New Agency.

We evaluate the NE translation quality by means of precision, recall and F-

		Precision	Recall	F-score
Correct	Type	72.18	72.18	72.18
	Token	83.07	83.07	83.07
Acceptable	Type	75.70	75.70	75.70
	Token	85.74	85.74	85.74

Table 6.7: NE translation performance on manually transcribed and manually annotated NEs

score. Precision (P) is calculated as the percentage of correctly translated NEs among the total number of translated NEs, while recall (R) is calculated as the percentage of correctly translated NEs among the total number of correct NEs in the manual annotation. F-score is defined as $2PR/(P+R)$. Due to errors from NE extraction and translation, we classified NE translation results into three categories:

- **Correct**, neither NE extractions nor translations have any errors;
- **Acceptable**, there are minor errors in either NE extraction or translation, but the results are acceptable. For example, two NEs "邓亚萍" and "杨影" are recognized as one NE "邓 亚萍 杨 影" and translated as "deng yaping yang ying";
- **Wrong**, there are significant errors in either NE detection or translation.

We evaluate NE translation performance in four scenarios. In addition to the three scenarios in NE extraction evaluation, we also measure the NE translation in an ideal scenario: translating manually annotated NEs from manual speech transcripts. In this case there are no errors from speech recognition and NE extraction. This illustrates the oracle performance of the NE translation module, as shown in Table 6.7. We evaluate the NE translation quality in both "Correct" and "Acceptable" categories. Type refers to the total number of unique NEs, while token refers to the total number of NEs. Since the NEs to be translated are the same as the correct NEs in the manual annotation, precision equals recall and F-score.

In the second scenario, we apply the standard NE tagger to the manual transcript, and then translate those automatically detected NEs. In this case,

		Precision	Recall	F-score
Correct	Type	57.19	58.80	57.98
	Token	75.66	73.05	74.33
Acceptable	Type	63.01	64.78	63.87
	Token	79.93	77.17	78.52

Table 6.8: NE translation performance on manually transcribed and automatically extracted NEs

		Precision	Recall	F-score
Correct	Type	44.09	48.59	45.96
	Token	70.28	56.34	62.54
Acceptable	Type	54.00	59.51	56.62
	Token	77.63	62.25	69.10

Table 6.9: NE translation performance on ASR hypothesis and automatically extracted NEs

the only errors are from automatic NE extraction. We get the NE extraction F-score of 85.59. The NE translation results are presented in Table 6.8. Notice that 15% NE extraction errors lead to an additional 7-10% NE translation errors.

In the third scenario, we apply the standard NE tagger directly on the ASR hypothesis, and get 66% NE extraction F-score. 18% character error rate leads to 19.3% increase on NE extraction errors. Translating these detected NEs, we get even lower NE translation quality. As shown in Table 6.9, both speech recognition errors and NE extraction errors lead to an additional 9% drop on NE token translation and 12% drop on NE type translation.

Finally, we evaluate the proposed context-based NE extraction and error correction method on the ASR hypothesis. Extracting NEs on the corrected hypothesis improves the NE extraction F-score from 66.25 to 71.73. Table 6.10 shows the overall translation quality on the corrected hypothesis. We observe an absolute 4-6% improvement on NE translation F-score over the straightforward ASR NE translation approach (shown in Table 6.9).

Figure 6.5 shows the F-score of overall NE extraction, type and token translations with degraded speech input. From left to right they are: manually transcribed and annotated NEs, manually transcribed but automatically extracted NEs, automatically recognized and extracted NEs, and the context-based NE

		Precision	Recall	F-score
Correct	Type	53.42	57.75	55.50
	Token	76.91	64.92	70.41
Acceptable	Type	59.93	64.79	62.26
	Token	80.47	67.93	73.67

Table 6.10: NE translation performance on improved ASR hypothesis and automatically extracted NEs

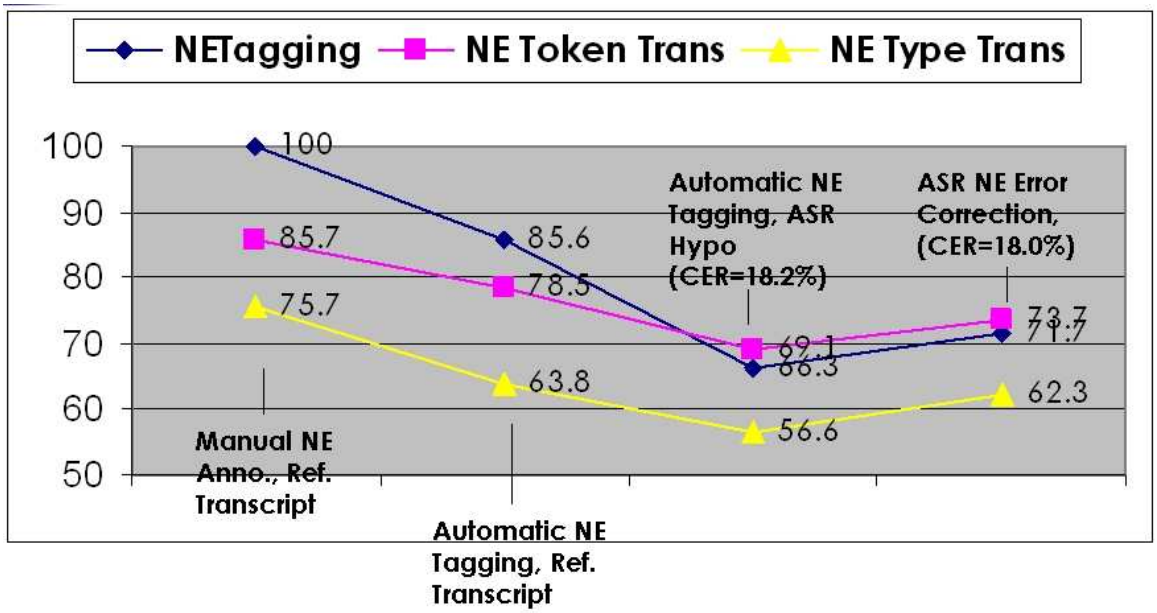


Figure 6.5: NE extraction and translation quality with degraded speech input

extraction and error correction. Obviously, with higher character recognition error rate and lower NE extraction F-score, NE translation quality decreases. However, compared with the degradation on speech recognition and NE extraction, the NE translation quality decreases much slower.

Some speech recognition, NE extraction and translation examples are shown as follows. To help non-Chinese speakers understand the pronunciation similarity between the reference sentence and the ASR hypothesis, we attach pinyin after each misrecognized Chinese word.

Manual transcription, NE annotation and translation:

任命 @PER{列显伦/liexianlun} @PER{沈澄/shencheng} @PER{包致金/baozhijin} 为/wei 终审 法院 常设 法官

appoint @PER{Henry Litton} @PER{Charles Ching} @PER{Kemal Bokhary} as Court of Final Appeal permanent judges

ASR Hypothesis, automatic NE tagging and translation:

任命 电线/dianxian 论/lun 县城/xiancheng 包/bao 之/zhi 心/xin 为/wei 终审 法院 常设 法官

appointed wire by the county package of mind for final trials court permanent judges

Corrected NE detection on ASR hypothesis and automatic translation:

任命 PER@{列/lie 显伦/xianlun} @PER{沈/shen 澄/cheng} @PER{包/bao 致/zhi 金/jin 生/sheng} 终审 法院 常设 法官

appointed @PER{patrick chan} @PER {charles ching} @PER{bokhary} for final trials court permanent judges

The first two sentences are manually transcribed speech and manually annotated and translated NEs. The middle two sentences are the ASR hypothesis and automatic NE extraction and translation results. Due to speech recognition errors on these person names, almost all the words are misrecognized and inappropriately translated according to their individual semantic meaning, which are neither coherent nor related to the source NEs. In the last two sentences, one may notice that three person names have been recovered from the initial recognition errors (although one additional character is incorrectly added into one person name). The last line shows the translations of these newly detected NEs, where one NE is correctly translated and another is partially correctly translated.

6.3 Summary

In this chapter we focus on NE extraction from speech input. We develop an adaptive NE extraction strategy, applying the NE tagger trained with broad-

cast news data on manual transcripts of meetings. We update the NE tagging model using information from both NE cache models and meeting profiles. We improve the NE extraction F-score from 47% to 60%. This result is even comparable to an NE tagger trained with a small amount of domain-specific NE annotated data. We also develop a context-based NE extraction model. This model identifies partial NE information, such as NE type and location, based on their left and right context as well as NE word duration. Combined with the information retrieval and candidate NE re-ranking techniques developed in Chapter 5, this model is able to detect and correct NE speech recognition errors. We also compare NE extraction and translation performances with different input, from manual broadcast speech transcripts to ASR hypothesis, and observe significant improvement using the proposed approach.

6. SPEECH NAMED ENTITY EXTRACTION AND TRANSLATION

Chapter 7

NAME TRANSLITERATION

Previous NE translation approaches, either NE alignment from sentence aligned parallel corpus, or searching for NE translations from target language monolingual corpus, rely on the existence of correct NE translations in target sentences or retrieved target language documents. With this precondition we apply several similarity feature functions to select the correct translation. When correct NE translations do not exist in the target text, we have to generate the name translation via transliteration, i.e., to translate based on phonetic approximation.

Given a source NE, machine transliteration generates a phonetically similar equivalent in the target language. The transliteration patterns are highly dependent on the name's origin, i.e., the country or the language family this name is from. For example, when transliterating names from Chinese into English, as shown in the following example, the same Chinese character "金" is transliterated into different English letters, according to the origin of each person.

金人庆 — *Jin* Renqing (China)

金大中 — *Kim* Dae-jung (Korea)

马丁 路德 金 — Martin Luther *King* (USA)

金丸信 — *Kane*maru Shin (Japan)

何塞 华金 布伦纳 — Jose Joa*quin* Brunner (Chile)

Several approaches have been proposed for name transliteration. Knight & Graehl (1997) proposed a generative transliteration model to transliterate foreign names in Japanese back to English using finite state transducers. Stalls & Knight (1998) expanded that model to Arabic-English transliteration. Meng *et al.* (2001) developed an English-Chinese NE transliteration technique using pronunciation lexicon and phonetic mapping rules. Virga & Khudanpur (2003) applied statistical machine translation models to "translate" English names into Chinese characters for Mandarin spoken document retrieval. All these approaches exploit a general model for NE transliteration, i.e., source names from *different* origins or language families are transliterated into the target language with the *same* rules or probability distributions, which fails to capture their different transliteration patterns.

Ideally, to explicitly model these transliteration differences we should construct a transliteration model and a language model for each origin. However, some origins have not enough NE pairs for reliable model training. We propose a cluster-specific NE transliteration framework. Considering that several origins from close language families may share the same pattern of transliteration, we group these origins into one cluster, and build cluster-specific transliteration and language models.

Starting from a list of bilingual NE translation pairs with origin labeled, we group closely related origins into clusters according to their language and transliteration model perplexities. For each cluster we train language and transliteration models from merged NE translation pairs. Given a source name, we first select appropriate models by classifying it into the most likely cluster, then we transliterate the source name with the corresponding models under the statistical machine translation framework. This cluster-specific transliteration framework dramatically improves the transliteration performance over the general transliteration model. Further more, we propose a phrase-based transliteration model, which effectively combines context information for name transliteration and achieves significant improvements over the traditional character based transliteration model.

7.1 Name Origin Clustering

Provided with a list of bilingual name translation pairs, whose origins are already labeled, we want to find the origin clusters where closely related origins (countries sharing similar languages or cultural heritages) are grouped together and less related origins are apart. We consider the following factors for clustering:

- Define a similarity measure between clusters;
- Select a clustering algorithm: hierarchical clustering vs. flat clustering. If hierarchical, bottom-up or top-down clustering;
- Define the clustering termination condition: how many clusters should be optimally generated?

Assuming a generative process in creating these name translation pairs from cluster-specific models, we define the similarity measure between two clusters as their LM and TM perplexities, i.e., the likelihood of generating one cluster's name pairs using the other cluster's character LMs and TM. We choose bottom-up hierarchical clustering, starting with each origin as a separate cluster. Finally, we select the desirable number of clusters based on source and target LM perplexities.

7.1.1 Cluster Similarity Measure Definition

Let $S_i = (F_i, E_i)$ denote a set of name translation pairs from origin i, from which origin i's model θ_i is trained:

$$\theta_i = (P_{c(i)}, P_{e(i)}, P_{t(i)})$$

where
$P_{c(i)}$: N-gram source character LM trained from F_i;
$P_{e(i)}$: N-gram target character LM trained from E_i;
$P_{t(i)}$: IBM-1 character translation models trained from S_i, including $P_{t(i)}(E|F)$,

the probability of generating a target letter given a source character, and symmetrically $P_{t(i)}(F|E)$ (Brown *et al.* (1990)).

The distance between origin i and origin j can be symmetrically defined as:

$$d(i,j) = -\frac{1}{|S_i|}\log P(S_i|\theta_j) - \frac{1}{|S_j|}\log P(S_j|\theta_i) \tag{7.1}$$

Assuming name pairs are generated independently,

$$P(S_i|\theta_j) \propto \sum_{t}^{|S_i|} \log\left[P_{c(j)}(F_i^t)P_{t(j)}(E_i^t|F_i^t) + P_{e(j)}(E_i^t)P_{t(j)}(F_i^t|E_i^t)\right] \tag{7.2}$$

$P(S_j|\theta_i)$ is defined in a similar way.

To ensure each origin has enough name pairs for reliable model training, we select M origins from a list of name translation pairs such that each origin has at least c pairs. Name pairs from the remaining origins are treated as unlabeled data for model re-training (see Section 7.2.2). We calculate the pairwise distances among these origins, and cluster them based on group-average agglomerative clustering (Manning & Schutze (1999)), where the distance between clusters C_i and C_j is the average distance over all member origin pairs, defined as:

$$D(C_i, C_j) = \frac{\sum_{i\in C_i}\sum_{j\in C_j} d(i,j)}{|C_i| \times |C_j|} \tag{7.3}$$

7.1.2 Clustering Scheme

The group-average agglomerative clustering algorithm implements bottom-up hierarchical clustering, as follows:

- Initialization:
  - Initialize current cluster number: $m = M$;
  - Specify desirable number of clusters: n
  - For $i = 1, \ldots, M, C_i = i$, i.e., each origin is a separate cluster.

- Repeat:
  while $m > n$
  - $\forall(i, j) \in [1, m]$, calculate $D(C_i, C_j)$;
  - if$(i', j') = \arg\min_{(i,j)} D(C_i, C_j)$, $C_{i'} = C_{i'} \bigcup C_{j'}, C_{j'} = \emptyset, m - -$.

The bottom-up clustering algorithm can generate M different cluster partitions, ranging from the initial M individual origin clusters to the final single general cluster. As a result, the order of origin merge represents a clustering tree, where the most similar origins are merged in the early stage and are closer to leaves in the tree. If we associate each node in the tree with its merging order, every ordered node represents a clustering configuration, which indicates the current clusters at that point.

7.1.3 Optimal Cluster Configuration Selection

To select the optimal number of clusters from the clustering tree, we calculate the probabilities of generating a held-out name pair list L from different cluster configurations, and select the one with the minimum perplexity.

Formally, the optimal cluster configuration

$$\omega^* = \arg\max_{\omega \in \Omega} P(L|\Theta_\omega); \tag{7.4}$$

where
Ω: The set of M clustering configurations in the tree, $\Omega = \omega_1, \omega_2, \ldots, \omega_M$;
Θ_ω: The set of cluster-specific LMs under configuration ω. $\Theta_\omega = \{\theta_j | j \in \omega\}$;
$P(L|\Theta_\omega)$: The likelihood of generating held-out name pairs from Θ_ω. It is the product of generating each name pair from its most likely name origin cluster:

$$\begin{aligned} P(L|\Theta_\omega) &= \prod_{t=1}^{|L|} max_{j \in \omega} P(F^t, E^t|\theta_j) P(\theta_j) \\ &= \prod_{t=1}^{|L|} max_{j \in \omega} P_{c(j)}(F^t) P_{e(j)}(E^t) P(\theta_j). \end{aligned} \tag{7.5}$$

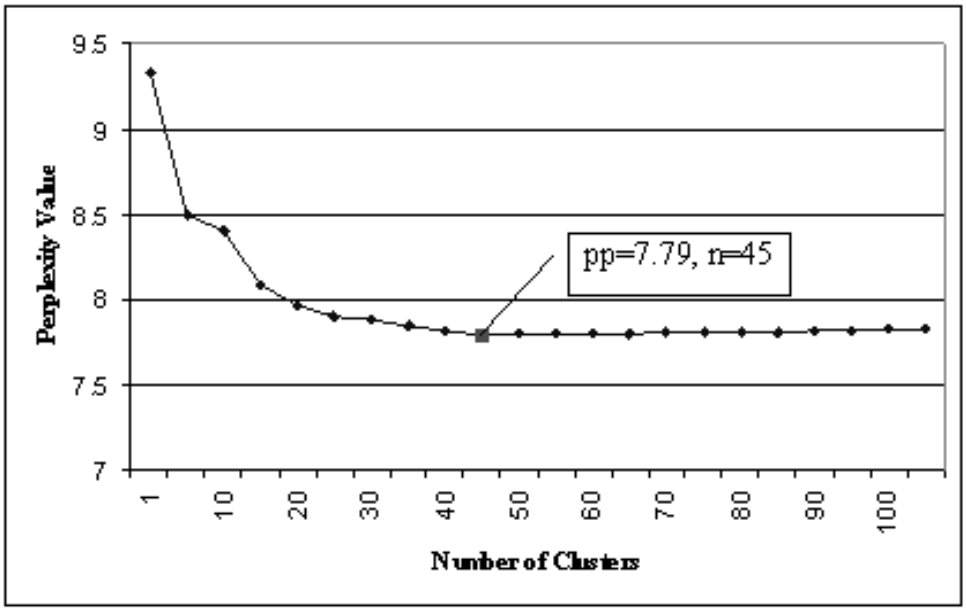


Figure 7.1: Perplexity value of LMs with different number of clusters

The language model perplexity is defined as:

$$pp(L, \Theta_\omega) = 2^{-\frac{1}{|L|} log P(L|\theta_\omega)} \quad (7.6)$$

$$= P(L|\Theta_\omega)^{-\frac{1}{|L|}} \quad (7.7)$$

We cluster 56K Chinese-English name pairs from M = 112 origins (we set $c = 50$ in our experiments) into different numbers of clusters. We evaluate the perplexities of different cluster configurations with regard to the held-out 3K name pairs from 112 origins. The held-out data have the same origin distribution as the training data. Figure 7.1 shows the perplexities curve. As one can see, it reaches its minimum when $n = 45$. This indicates that the optimal cluster number is 45. Table 7.1 lists some typical origin clusters. It can be easily seen that countries are often grouped together according to their language families. These countries are either geographically adjacent or historically affiliated. For example, while the Kazakh language belongs to the Central Turkic language family, many Kazakh names in our training data have sub-strings like "-yev", "-chenko" and "-vich", thus Kazakhstan is clustered into the Rus-

sian group. In the English group, the Netherlands (Dutch) seems an abnormity. Actually it is first merged with South Africa, which was colonized by the English and Dutch in the seventeenth century, then further clustered into this English-speaking group. Additionally, some origins cannot be merged with any other clusters because they have unique names and translation patterns, e.g., China and Japan, and they are kept as single origin clusters.

7.2 Name Origin Classification

After similar name origins are grouped into clusters, we can train an origin classifier to classify source names or name translation pairs into their most likely cluster. Identifying the source name's origin enables appropriate cluster-specific modeling for name transliteration, as presented in the next section. Also, identifying a name pair's origin helps combine more unlabeled training data for each cluster, which has the potential to train better name classification and transliteration models.

7.2.1 Identify Origin Cluster with Source Names

Given a source name, we want to find the most likely cluster it is from. We use the source character language model as the classifier, and assign the name to the cluster with the highest LM probability. Assuming a source name is composed of a sequence of source characters: $F = f_1, f_2, \ldots, f_l$. We want to find the cluster j^* such that

$$\begin{aligned} j^* &= \arg\max_j P(\theta_j|F) \\ &= \arg\max_j P(\theta_j)P(F|\theta_j) \\ &= \arg\max_j P(\theta_j)P_{c(j)}(F) \end{aligned} \tag{7.8}$$

where $P(\theta_j)$ is the prior probability of cluster j, proportional to the number of name translation pairs in all the training data. $P_{c(j)}$ is the probability of generating this source name under cluster j's character-based n-gram language model.

Arabic	Afghanistan, Algeria, Egypt, Iran, Iraq, Jordan, Kuwait, Pakistan, Palestine, Saudi Arabia, Sudan, Syria, Tunisia, Yemen, ⋯
Spanish-Portuguese	Angola, Argentina, Bolivia, Brazil, Chile, Colombia, Cuba, Ecuador, Mexico, Peru, Portugal, Spain, Venezuela,⋯
English	Australia, Canada, Netherlands, New Zealand, South Africa, UK, USA,⋯
Russian	Belarus, Kazakhstan, Russia, Ukraine
East European	Bosnia and Herzegovina, Croatia, Yugoslavia
French	Benin, Burkina Faso, Cameroon, Central African Republic, Congo, Gabon, Ivory Coast
German	Austria, Germany, Switzerland
French (2)	Belgium, France, Haiti
Korean	North Korea, South Korea
Danish-Swedish	Denmark, Norway, Sweden
Single Clusters	China Japan Indonesia Israel ⋯

Table 7.1: Typical name origin clusters (n=45)

7.2.2 Identify Origin Clusters with Name Translation Pairs

In addition to the bilingual name pairs whose origins are manually labeled for origin clustering, we have a lot more name pairs without origin labels. We want to classify these name pairs into appropriate clusters, and retrain each cluster's classification and transliteration models with augmented training data.

We adopt the metric defined in formula 7.5 for name pair classification. Given a name translation pair (F, E) , the most likely cluster j^* is defined as:

$$\begin{aligned} j^* &= \arg\max_j P(\theta_j|F,E) \\ &= \arg\max_j P(\theta_j)P(F,E|\theta_j) \\ &= \arg\max_j P(\theta_j)P_{c(j)}(F)P_{e(j)}(E) \end{aligned} \tag{7.9}$$

As the sizes of character vocabularies are relatively small (30 for the English vocabulary and 6000+ for the Chinese one), we can use large N's in N-gram LM. A suffix array language model based on the implementation described in (Zhang & Vogel (2005)) allows the access to history with arbitrary length, and thus is a good candidate for this task.

After we classify these unlabeled name pairs into appropriate clusters, we retrain the origin classifiers with augmented training data (original labeled name pairs plus newly classified name pairs), similar to the co-training algorithm (Blum & Mitchell (1998)). In our case, we combine decisions from two independent classifiers: source character LM (CLM) and target letter LM (ELM), and select name pairs which are confidently and consistently classified by both classifiers for model re-training.

Define the confidence measure of classifying name F into cluster j based on classifier k (k= CLM, ELM):

$$p_k(\theta_j|F) = \frac{P(\theta_j)P_{k(j)}(F)}{\sum'_j P(\theta'_j)P_{k(j')}(F)} \tag{7.10}$$

Define the most likely cluster based on CLM:

$$j_c^*(F) \equiv \arg\max_j p_{clm}(\theta_j|F), \tag{7.11}$$

and the most likely cluster based on ELM:

$$j_e^*(E) \equiv \arg\max_j p_{elm}(\theta_j|E). \tag{7.12}$$

The standard co-training algorithm selects name pairs satisfying

$$\begin{cases} p_{clm}(j_c^*|F) & > & h \ \ or \\ p_{elm}(j_e^*|E) & > & h \end{cases} \tag{7.13}$$

for classifier re-training, where h is the confidence score threshold (h = 0.9 in our experiments). A more strict constraint is:

$$\begin{cases} j_c^*(F) & = & j_e^*(E); \ \ and \\ p_{clm}(j_c^*|F) & > & h; \ \ and \\ p_{elm}(j_e^*|E) & > & h. \end{cases} \tag{7.14}$$

Based on these two criteria we add confidently classified unlabeled name pairs to the labeled data, re-train two classifiers. We compare them with the baseline classifier, which is trained only from the labeled data.

7.2.3 Experiments

Our labeled and unlabeled data are 56K Chinese-English name translation pairs with origin labels, as well as 486K name pairs without origin labels. All the name lists are from the Linguistic Data Consortium Chinese-English person name lists, originally from Xinhua News Agency. We extract 3K name pairs as a development set and 3K as a test set, with the same origin distribution as in the training data. As mentioned above, 56K name pairs from 112 origins are clustered into 45 origin clusters. We evaluate name origin classification accuracies.

We classify source names and name translation pairs using different fea-

N	2	3	4	5	6	7
CLM	83.62	**84.88**	84.00	84.04	83.94	83.94
ELM	83.74	88.09	89.71	89.96	90.10	90.02
CELM	89.58	91.13	91.07	91.07	90.97	90.91
Best	CLM, N=3, ELM, N=6, Accuracy = 91.15%					

Table 7.2: Origin classification accuracies given source name and name translation pair, using different features.

Model	Baseline	Cot	CotStr
CLM (N=3)	84.88	83.95	84.97
ELM (N=6)	90.10	89.00	89.87
CELM	91.15	90.06	91.02

Table 7.3: Co-training classification accuracies on dev. set Model

tures: source language character LM (CLM), target language character LM (ELM), and the combination of both LMs (CELM). We also try N-gram LM with different Ns, and select the best configuration (different Ns for CLM and ELM). Table 7.2 shows the classification accuracy. We find that 3-gram is sufficient for the Chinese LM, while 6-gram achieves the best result for the English LM. Under these configurations, the combined CELM achieves 91.15% classification accuracy. A detailed analysis indicates that some classification errors are due to the inherent uncertainty of certain names, e. g, " (Gary Locke)", a Chinese American, is classified as a Chinese name while his country origin is USA.

We apply the name origin classifiers to the 486K name pairs without origin labels, and select confidently classified name pairs for model re-training. We apply the standard co-training constraint (**Cot**) and a more strict constraint (**CotStr**) to select qualified name pairs (see formulae 7.13 and 7.14), and re-train the origin classifiers with the augmented name translation pairs. As a result, **Cot** select 289K name pairs for model re-training, and **CotStr** select 83K name pairs. We compare their performances with the baseline model (**Baseline**) trained only on the 56K labeled name pairs.

Table 7.3 and 7.4 list the classification accuracies on the development and test set. As we can see, classifiers trained with the standard co-training con-

Model	Baseline	Cot	CotStr
CLM (N=3)	84.20	83.59	84.26
ELM (N=6)	89.52	89.45	89.81
CELM	90.76	90.25	90.92

Table 7.4: Co-training classification accuracies on eval set Model

straint (**Cot**) consistently have lower classification accuracy than the baseline classifiers, while **CotStr** achieves comparable or even better performance. One possible reason is that the **Cot** strategy aggressively adds misclassified name pairs, which mislead the baseline classifiers.

7.3 Cluster-specific Name Transliteration

We propose a phrase-based transliteration model, which effectively combines context information for name transliteration and achieves significant improvements over the traditional character-based transliteration model.

7.3.1 Phrase-based Name Transliteration

Statistical NE transliteration is similar to the statistical machine translation in that an NE translation pair can be considered as a parallel sentence pair, where "words" are characters in source and target languages. Due to the nature of name transliteration, decoding is mostly monotone.

NE transliteration process can be formalized as:

$$\begin{aligned} E^* &= \arg\max_E P(E|F) \\ &= \arg\max_E P(E)P(F|E) \end{aligned} \tag{7.15}$$

where E^* is the most likely transliteration for the source NE F, $P(F|E)$ is the transliteration model and $P(E)$ is the character-based target language model.

A transliteration model provides a conditional probability distribution of target candidates for a given source transliteration unit. It can be a single character or a character sequence, i.e., "phrase". Their transliteration candidates

can be identified from a character alignment path through Viterbi search, and the transliteration probabilities are estimated based on their co-occurrence frequency.

A naive choice of transliteration unit is the single character. However, single characters lack contextual information, and their combinations may generate too many unlikely candidates. Motivated by the success of phrase-based machine translation approaches (Wu (1997), Och *et al.* (1999), Marcu & Wong (2002) and Vogel *et al.* (2003)), we select appropriate transliteration units which are long enough to capture contextual information while flexible enough to compose new names with other units. We discover such source transliteration phrases based on the character collocation likelihood ratio test (Manning & Schutze (1999)). This test accepts or rejects a null hypothesis that the occurrence of one character f_1 is independent of another character f_2 by calculating the likelihood ratio between the independent (H_0) and dependent (H_1) hypotheses:

$$\log \lambda = \log \frac{L(H_0)}{L(H_1)} \tag{7.16}$$

$$= \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) \tag{7.17}$$
$$- \log L(c_{12}, c_1, p_1) - \log L(c_2 - c12, N - c_1, p_2).$$

L is the likelihood of getting the observed word counts under each hypothesis. Assuming the character frequency follows a binomial distribution,

$$L(k, n, x) = \begin{pmatrix} n \\ k \end{pmatrix} x^k (1-x)^{n-k}, \tag{7.18}$$

c_1 and c_2 are the frequencies of f_1 and f_2, and c_{12} is the co-occurrence frequency of f_1 and f_2. N is the total number of characters. p, p_1 and p_2 are defined as:

$$p = \frac{c_2}{N}, \tag{7.19}$$

$$p_1 = \frac{c_{12}}{c_2}, \tag{7.20}$$

$$p_2 = \frac{c_2 - c_{12}}{N - c1} \tag{7.21}$$

We calculate the likelihood ratio for any adjacent source character pairs, and select those pairs with high ratios. Adjacent character bigrams with one character overlap can be recursively concatenated to form longer source transliteration phrases. To discover translation candidates for a given source phrase, we collect all NE translation pairs containing the source phrase. For each NE pair we

- Convert the Chinese characters into their romanization form, pinyin, then align them with English letters via phonetic string alignment.
- Segment the Chinese name into a sequence of source transliteration phrases. The initial phrase alignment path can be identified from the character alignment path.
- Apply a beam search around the initial phrase alignment path, searching for the optimal alignment which minimizes the overall phrase alignment cost.

The phrase alignment cost is defined as:

$$A^* = \arg\min_{A} \sum_{i,a_i \in A} D(f_i, e_{a_i}), \tag{7.22}$$

where f_i is the ith source phrase, e_{a_i} is its target candidate under alignment A. Their alignment cost D is defined as the linear interpolation of the phonetic transliteration cost and the semantic translation cost:

$$D(f, e) = \lambda_1 \log P_{trl}(e|f) + \lambda_2 \log P_{trans}(e|f). \tag{7.23}$$

λ_1 and λ_2 are the interpolation weights, reflecting the relative contributions of the transliteration cost and the translation cost. They usually differ from cluster to cluster. For example, the transliteration cost is usually the dominant feature for most Latin language clusters. However, for Japanese cluster, the translation cost is more important. This is because Japanese names share similar characters with Chinese, thus the Japanese names are often translated semantically, using the same kanji characters. As a result, the translation feature weight is more important for the Japanese cluster ($\lambda_1 = 0$ in this case). We

Arabic	穆罕默德 mohamed 阿卜杜勒 abdul 艾哈迈德 ahmed
	尤: yo (0.27) y(0.19) you(0.14)···
English	约翰 john 威廉 william 彼得 peter
	尤: u(0.25) you(0.38) joo(0.16)···
Russian	弗拉基米尔 vladimir 伊万诺夫 ivanov -耶维奇 -yevich
	尤： yu(0.49) y(0.08) iu(0.07)···

Table 7.5: Transliteration unit examples from three name origin clusters

empirically select the interpolation weight for each cluster, and the combined model with optimal interpolation weights achieves the best performance.

We extract phrase transliteration pairs from top N alignment paths, associated with their confidence scores. We estimate their transliteration probabilities according to the fractional alignment frequencies (confidence scores). We also include frequent sub-name translations (first, middle and last names) in the transliteration dictionary. Table 7.5 shows some typical transliteration units (characters or phrases) from three clusters. They are mostly names or sub-names capturing cluster-specific transliteration patterns. It also illustrates the same source character having different transliteration candidates with different probabilities in different clusters, which justify the cluster-specific transliteration modeling.

7.3.2 Language Model and Decoding

For each cluster a target character language model is trained from target NEs. We use the N-gram models with standard smoothing techniques. During monotone decoding, a source NE is segmented into a sequence of transliteration units via maximum length string matching, and each source unit is provided with a set of candidate translations with corresponding probabilities. A transliteration lattice is constructed to generate all transliteration hypotheses, and the

one with the minimum transliteration and language model costs is selected as the final hypothesis.

7.3.3 NE Transliteration Evaluation

We first evaluate the name transliteration results for each cluster. In this scenario we know each source name's true origin cluster. Secondly we evaluate the overall results without the origin label information. We need to classify a given source name into the most likely cluster, then use the cluster-specific transliteration and language models to translate the source name into the target language. Obviously, name origin classification errors will almost surely lead to name transliteration errors.

The transliteration hypotheses are evaluated based on three metrics:

- Top1 accuracy (**Top1**), the percentage of correct NE transliterations in the top 1 hypothesis, compared with human translations (**reference**);
- Top 5 accuracy (**Top5**), the percentage of correct NE transliterations in the top 5 hypotheses, compared with reference translations;
- Character error rate (**CER**), the percentage of incorrect characters (inserted, deleted and substituted English letters) when the top 1 transliteration hypothesis is aligned with the reference translation.

Our baseline system is a character-based general transliteration model, where 56K NE pairs from all clusters are merged to train a general transliteration model and language model (**General**). We compare the CERs of several typical clusters using the general model and cluster-specific models, and the results are shown in Table 7.6. Because in the training data more than half of the names are from Latin language clusters (Arabic, English, French, Spanish-Portuguese etc.), the general model favors Latin name transliteration patterns. As a result, it obtains reasonable (20-30%) CERs on these clusters, but strikingly high (over 70%) CERs on other oriental language clusters such as Chinese, Korean and Japanese, even though the Chinese cluster has the most name translation pairs for training. Examples in Table 7.2 show a Chinese name's western style

Cluster	Data size	General CER (%)	Char. CER (%)	Phrase CER (%)	Avg. trans. per src. phrase
Arabic	8336	22.88	18.93	14.47	4.58
Chinese	27093	76.45	1.69	1.71	3.43
English	8778	31.12	29.21	17.27	5.02
French	2328	27.66	18.81	9.07	3.51
Japanese	2161	86.94	38.65	29.60	7.57
Russian	4407	29.17	9.62	6.55	3.64
Spanish	8267	18.87	15.99	10.33	3.61

Table 7.6: Cluster-specific transliteration comparison

transliteration, "Van Tylen", under this general model, while the correct translation should be "Fan Zhilun".

When applying the character-based (**char**) cluster-specific models, transliteration CERs are consistently decreased for all clusters (ranging from 6.13% relative reduction for the English cluster to 97% for the Chinese cluster). As one may expect, the most significant error reduction occurs on the oriental language clusters because cluster-specific models are able to represent their unique transliteration patterns. When we apply the phrased-based transliteration models, CERs are further reduced by 23% 51% for most clusters, thanks to the context information encapsulated in the transliteration phrases. Because most Chinese names are translated according to the pinyins of single characters, phrase-based transliteration slightly decreases the performance.

Additionally, different clusters vary a lot in their transliteration accuracies. The Chinese cluster achieves 96.09% top 1 accuracy and 1.69 CER with the character-based model because Chinese name translation patterns are rather regular: a single Chinese character's pinyin is often the correct translation. Other clusters have CERs ranging from 7% to 30%, which is partly due to less training data (e.g, the Japanese cluster), and partly because of the language-specific transliteration patterns. We measure the average number of translations per source phrase for each cluster, as shown in Table 7.6. This feature reflects the transliteration pattern regularity of different clusters, and seems a good indicator of the transliteration CERs. For example, the Russian clus-

Cluster	Source	Reference	General	Cluster-Specific	BabelFish
Arabic	纳吉 萨布里 艾哈迈德	Nagui Sabri Ahmed	Naji Saburi Ahamed	Naji Sabri Ahmed	In natrium 吉 萨 cloth Aihamaide
Chinese	范志伦	Fan Zhilun	Van Tylen	Fan zhilun	Fan Zhilun
English	罗伯特 斯特德沃德	Robert Steadward	Robert Stdwad	Robert Sterdeward	Robert Stead Warder
French	让-吕克 科雷捷	Jean-luc Cretier	Jean-luk Crete	Jean-luc Cretier	Let - Lu Keke lei Jie
Japanese	小林隆治	Kobayashi Ryoji	felinonge	Kobayashi Takaji	xiaolin prosperous governs
Russian	弗拉基米尔 萨姆索诺夫	Vladimir Samsonov	Frakimir Samsonof	Vladimir Samsonov	弗拉基 mil sum rope Knoff
Spanish	鲁道夫 卡多索	Rodolfo Cardoso	Rudouf Cardoso	Rodolfo Cadozo	Rudolph card multi- ropes

Figure 7.2: Transliteration examples from some typical clusters

Model	Top1(%)	Top5(%)	CER(%)
General (char)	3.78±0.69	5.84±0.88	50.29±1.21
Cluster (char)	51.08±0.84	56.50±0.87	14.00±0.34
Cluster (phrase)	56.00±0.84	62.66±0.91	12.84±0.41

Table 7.7: Transliteration result comparison

ter has 3.64 translations per source phrase on average, and obtains 6.55% CER with around 4400 training data, while the English cluster includes more flexible translation patterns, with 5.02 translations per source phrase, and gets 17.27% CER with over 8700 names for training. Transliteration examples from some typical clusters are shown in Table 7.2. Here we compare the hypotheses from the general model, the phrased-based cluster-specific model, and BabelFish , a online machine translation system. The general model tries to transliterate every name in the Latin romanization, regardless of each name's original languages. The BabelFish system sometimes incorrectly translates source characters based on their semantic meanings, and the results are difficult to understand.

We also compare the overall performance on all the test data, where we first classify the source name into the most likely cluster, then transliterate this name with the appropriate models. The results are shown in Table 7.7. Because the general model performs rather poorly when transliterating oriental names, which account for almost half of the test data, the overall CER (50%) is rather high. Note that these results is also comparable to other state-of-the-art statistical name transliteration systems (Virga & Khudanpur (2003)). The character-based cluster-specific transliteration model dramatically improves the top 1 and top 5 transliteration accuracies from 3.78% to 51.08%, and from 5.84% to 56.50%, respectively. Consistently, the CER is also reduced from 50.29% to 14.00%. Phrase-based transliteration further increases the top 1 accuracy by 9.3%, top 5 accuracy by 10.7%, and reduces the CER by 8%, relatively. All these improvements are statistically significant.

7.4 Summary

In this chapter we propose a cluster-specific model for name transliteration. Noticing that name transliteration patterns are highly dependent on the name's origin, we group closely related origins into clusters. Starting from a list of bilingual NE translation pairs with their origin, we build language and transliteration models for each origin. Models from different origins are recursively merged until the optimal number of clusters is reached. For each cluster we train language and transliteration models from merged NE translation pairs. Given a source name, we first select appropriate models by classifying it into the most likely cluster, then we transliterate the source name with the corresponding models under the statistical machine translation framework. This cluster-specific transliteration framework dramatically improves the transliteration performance over the general transliteration model. Further more, we propose a phrase-based transliteration model, which effectively combines context information for name transliteration and achieves significant improvements over the traditional character based transliteration model.

Chapter 8

NAMED ENTITY INFORMATION-THEORETIC MEASURE

8.1 Introduction

Existing evaluation metrics for both automatic speech recognition (ASR) and machine translation (MT), such as word/character error rate or Bleu scores, do not differentiate the information carried by different words. Content words, functional words or even punctuation marks are treated equally. However, considering the utility of ASR/MT hypotheses, the correct recognition and translation of key words, such as named entities, should be more important than a comma. On the other hand, the effectiveness of these key words has been proved in other natural language processing tasks, such as information retrieval and question answering. In this project, we propose to investigate the information loss caused by misrecognized and mistranslated named entities.

8.2 LM-based Information Theoretic Measure

The information carried by a word is context-dependent. Intuitively, the word "agency" carries more information by itself than the case when it is followed

by "Xinhua News". In the latter case given the first two words, one naturally expects the most likely next word is "agency". Therefore we calculate the information carried by each word depending on its history. Using language models,

$$Info(w_i|w_{i-1}, w_{i-2}) = -\log P(w_i|w_{i-1}, w_{i-2}), \tag{8.1}$$

where P is a 3-gram language model trained from general-domain text. This training data represents the prior knowledge about the world, similar to human's common sense. When i ¡= 2, the 3gram model is backoffed to 2-gram and 1-gram models. The information carried by a sentence, a phrase or a named entity is the sum of the information carried by each word:

$$Info(S : w_1^n) = \sum_{i=1}^{n} Info(w_i|w_{i-1}, w_{i-2}). \tag{8.2}$$

Given a machine generated (ASR or MT) hypothesis, we measure the information loss from misrecognized or incorrectly translated named entities. First we manually annotate NEs in the reference sentence (manual speech transcripts or human reference translations). We calculate the information carried by each NE. Given these reference NEs, we try to find the corresponding NEs in the hypothesis via automatic NE tagging. Reference NEs that cannot be fully matched cause NE information loss.

8.3 NE Alignment

To find the reference NEs in the ASR/MT hypotheses is not as straightforward as one might expect. Errors in NE speech recognition, automatic tagging and NE translation cause various problems:

- NEs may be misrecognized or incorrectly translated, thus it is impossible to find the correct NEs in the hypotheses;
- Even if NEs are correctly recognized or translated, automatic NE tagging may not be able to find them as the NE tagger tends to make more mis-

takes on imperfect transcripts;

- The recognized or translated NEs may not be the same as the reference NEs, but they carry the same meaning, e.g., ”UN” and ”United Nations”. Therefore a synonym NE list is necessary to include acceptable variations of NEs.

For example, in the following ASR sentence,

> **Ref:** @ORG{IBM Corp.} announced its merge with @ORG{Leveno} early today. **ASR:** @ORG{IBM} cops announced its merge with the @ORG{Leveno} early today.

Automatic NE tagging identifies ”IBM” and ”Leveno” as organization NEs. Notice that ”Leveno” occurs in both the manual transcript and the ASR hypothesis, and can be perfectly aligned without any information loss. ”IBM Corp.” in the manual transcript should be aligned with ”IBM” in the ASR hypothesis. From information-theoretic point of view, we can calculate the information loss of the missing word ”Corp.” given ”IBM” as its history:

$$Info(\text{"}corp.\text{"}|\text{"}IBM\text{"}) = -\log P(\text{"}corp.\text{"}|\text{"}IBM\text{"}).$$

From user's utility perspective, ”IBM” and ”IBM Corp.” could be considered as equivalent, so we can generate a synonym NE list and consider certain NE partial matches acceptable.

Sometimes NEs in the generated hypothesis are more difficult to align. This problem is more serious for matching NEs in machine translation hypotheses and reference translations. Because of the inherent non-monotone and one-to-many mapping in machine translation, there are more variations regarding to NE translation hypotheses. For example,

> **Src:** @ORG{新华社} @LOC{广州} 3 月 16 日 电 （ 记者 @PER{陈 冀} ） 最 新 统计 字 显示 ， 今年 1 至 2 月 ， @LOC{广 东省} 高 新 技术 产品 出口 37.6 亿 美元， 同 比 增长 34.8%, 占 全省 出口 总值 的 25.5% . **Ref:** @ORG{Xinhua News Agency} , @LOC{Guangzhou}, March 16 (Reporter @PER{Chen Ji}) The latest statistics show that

> from January through February this year , the export of high-tech products in @LOC{Guangdong Province} reached 3.76 billion US dollars , up 34.8% over the same period last year and accounted for 25.5% of the total export in the province . **MT:** @LOC{guangzhou}, march 16 (@ORG{xinhua}) – the latest available statistics show that from january to february , @LOC{guangdong}, the export of high-tech products 3.76 billion dollars , compared with 34.8% , the province 's total exports of 25.5% .

NEs from the reference translation and MT hypothesis are extracted, as shown below:

> **RefNE:** xinhua news agency — guangzhou — chen ji — guangdong province **HypNE:** guangzhou — xinhua — guangdong

For automatically NE alignment, we consider that each reference NE can be aligned to any hypothesis NEs. Because correct NE translations may only differ in their word orders (e.g., "FIFA Executive Committee" vs. "the Executive Committee of FIFA"), the alignment cost is based on word-to-word string editing distance. In other word, we try to find the optimal word-to-word alignment path between two NEs, and we consider the total number of unaligned words in both NEs as the distance measure. With this approach we also consider the NE partial match, and calculate NE information loss from missing words. For example, the alignment cost between "xinhua news agency" and "xinhua" is the sum of the alignment costs for "xinhua" and "xinhua" (cost=0), "news" and NULL (cost=constant) and "agency" and NULL (cost=constant). We calculate the string alignment cost for all the NE pairs, and select the pairs with the minimum alignment cost as aligned NEs. We also identify the word alignment path within each pair and calculate the information loss from missing word. To deal with automatic NE tagging errors, when the reference NEs are not perfectly aligned, we also search for its occurrence in the hypothesis sentence.

As a result, with the above extracted NE pairs, the NE "guangzhou" is perfectly aligned. The NE "guangdong province" is not found in the hypothesis, but the closest match is the NE "guangdong". From the word alignment path

we identify the missing word "province", and the corresponding information loss:

$$Info(\text{"}province\text{"}|\text{"}in\text{"},\text{"}guangdong\text{"}) = 1.69\ bit.$$

Similarly, the NE "xinhua news agency" is aligned to "xinhua", with the information loss:

$$Info(\text{"}news\text{"}|\text{"}xinhua\text{"}) = 5.53\ bit$$
$$Info(\text{"}agency\text{"}|\text{"}xinhua\text{"},\text{"}news\text{"}) = 0.01\ bit$$

So the total information loss is 5.54 bit. Finally, the person name "chen ji" is missing in the hypothesis, and the information loss is 24.2 bits. The total NE information loss in the above sentence is 31.43 bits.

8.4 Experiments

We select 6M English words and 5.5M Chinese words from newswire resources as our training data. The Chinese and English texts are parallel, i.e., they are sentence-by-sentence translations. Based on formula 8.2, we calculate the total amount of information carried by the source sentence in Figure 2 (328.95 bits), the reference translation (325.24 bits), and the MT hypothesis (248.96 bits). We notice that the source and target translations carry the same amount of information while the MT hypothesis loses quite some information. Even so, the information carried by the MT hypothesis may not be the same as the reference translation's information.

8.4.1 NE Information Loss from ASR Hypothesis

We apply the above information loss measure on speech recognition hypotheses. We experiment with both Chinese and Arabic speech. The Chinese test set is 1 hr Chinese broadcast news speech from Hub 4 1997 test data. It contains 104 speech segments with 9176 words, and the ASR hypothesis from Janus

	Perfect	Acceptable
Chinese (WER=18.2)	23.71%	23.37%
Arabic (WER=20.2)	24.81%	24.81%

Table 8.1: NE Information Loss for Chinese and Arabic ASR

ASR system has a word error rate (WER) of 18.2%. The Arabic test set is 103 speech segments from FBIS test data. It contains 13277 words, and the ASR hypothesis has a word error rate of 20.2%. We compare the information loss under two conditions: perfect match and acceptable match where a synonym NE list is created to allow flexible NE matches. The result is shown in Table 1. We find that 23.7% of total NE information is lost due to Chinese NE speech recognition errors, and 24.8% total NE information is lost for Arabic NE speech recognition. We find that on average the NE information loss is about 5% higher than the WER, and this confirms our assumption that existing WER is a lower bound of the actual information loss.

8.4.2 NE Information Loss from MT Hypothesis

We also apply the information loss measure on machine translation hypotheses. We experiment with both Chinese-English and Arabic-English translation tasks. The Chinese test set is 200 Chinese newswire sentences from NIST MT evaluation test data. The Arabic test set is 203 Arabic newswire sentences from NIST MT evaluation test data. We select one reference translation for each test set, and manually annotate NEs in the reference translations. The machine translation hypotheses are from CMU STTK system, with and without augmented NE translation functions. The MT evaluation scores (Bleu and NIST) are shown in Table 8.2 and 8.3. We create a synonym NE list for each test set to evaluate the information losses under "acceptable conditions". Table 8.2 and 8.3 also show the percentage of NE information loss from machine translation. As we can expect, the better the MT quality is, the less the NE information loss is. We also find that adding the NE translation module significantly reduces the NE information loss by 25% and 12% for Chinese and Arabic under the perfect match condition, and 49% and 50% under the acceptable condition.

	MT Eval Metric		NE InfoLoss Measure	
	Bleu	NIST	Perfect	Acceptable
STTK	19.98	7.82	52.92%	35.81%
STTK+NE	20.79	7.98	39.70% (-24.95%)	18.38% (-48.67%)

Table 8.2: NE Information Loss for Chinese-English MT

	MT Eval Metric		NE InfoLoss Measure	
	Bleu	NIST	Perfect	Acceptable
STTK	43.37	9.03	34.60%	20.51%
STTK+NE	44.04	9.17	30.32% (-12.37%)	10.11% (-50.71%)

Table 8.3: NE Information Loss for Arabic-English MT

8.5 Discussion

Our analysis focuses on the information loss caused by ASR and MT errors. We did not measure the information noises introduced by misrecognized or incorrectly translated NEs. Obviously these noises will significantly bring in unwanted information and spread errors over context words around NEs. So the information loss we estimate is the lower bound of the negative effect from misrecognized and incorrectly translated NEs. It will be interesting to explore the information noises in a quantitative manner.

Additionally, when certain names cannot be reliably recognized or translated, it may be more reasonable to transcribe them as name class tags in ASR or keep the original foreign names in MT rather than introducing errors by boldly recognize or translate them. One may measure how much information loss there are in such a situation.

Finally, we measure the information carried by each word based on a language model, which is trained with general domain text. The training data represents the prior world knowledge. However, this world knowledge will differ from person to person, as each individual has a unique knowledge base. Even for the same person, his/her information needs will vary depending on the document he/she reads, the specific task the reader fulfills and other factors. The task-oriented information utility measure such as answering certain questions after reading a document may be relevant.

8.6 Summary

We present an information-theoretic measure to estimate the information loss from ASR and MT. We estimate the information loss caused by NE speech recognition and machine translation. This measure provides another perspective to measure the utility of ASR and MT hypotheses. We observe significant information loss reduction (about 50%) using our NE translation techniques.

Chapter 9

Conclusions

In this chapter we conclude the dissertation by summarizing the thesis work and proposing several directions for future research.

9.1 Summary

We proposed an effective language-independent framework to extract and translate NEs from text and speech. Within this framework, we developed various features and algorithms, applied them to text and speech NE extraction and NE translation tasks in multiple language pairs.

We adopted the hidden Markov model (HMM) as our baseline NE extraction system, and trained and evaluated NE taggers in different languages (Arabic, Chinese and English). With different resources and different problems to solve, we expanded the baseline model in the following ways:

- We used a bootstrapping technique to train a Chinese NE tagger from imperfectly labeled data. The re-trained NE tagger achieved better NE extraction performance.

- We adapted the NE tagger trained from broadcast news data to NE extraction from meeting transcripts. The adaptive NE tagging model incorporated global and local context information, and significantly improved the NE extraction performance.

- We developed a context-based NE extraction model to identify possible NE types and locations from ASR hypothesis. This approach, combined with speech recognition confidence measures and information retrieval techniques, corrected NE speech recognition errors and improved NE extraction performance.

For NE translation, we developed several language-independent phonetic and semantic (NEs and their context) features to capture different similarity measures between source and target NE pairs. We incorporated these features into an NE translation framework to solve various NE translation problems in different language pairs (Chinese-English, Arabic-English and Hindi-English) with varying input data streams (text and speech) and resources (monolingual and bilingual):

- To align NE translations from sentence-aligned bilingual corpora, where NEs have been labeled independently in both languages;

- To project NEs within a sentence-aligned bilingual corpus, where NEs have been labeled for only one language, and we attempted to find their translations in the other language;

- To search for the target NE translation from monolingual corpora, given a source NE and possibly its context information as the query.

We achieved significant improvements on NE alignment and translation accuracy. When we incorporated the translated NEs into machine translation systems, we also improved the overall machine translation quality.

Finally we developed a cluster-specific name transliteration framework. By grouping names from similar origins into one cluster and training cluster-specific character and phrase transliteration and language models, we managed to dramatically reduce the name transliteration error rates. The combined NE translation techniques reduced NE information loss by 50%.

9.2 Conclusion

NE extraction and translation are important tasks for information extraction, machine translation and crosslingual information retrieval and question answering. In this thesis work, we mainly focus on speech NE extraction and NE translation. We draw the following conclusions from our thesis work:

1. When reasonable amount of training data is available, existing NE extraction methods achieve satisfactory performances on well-formed text. Applying them to speech from a different genre, performances drop significantly. Adaptation strategies such as domain and topic adaptation can improve the speech NE extraction performance.

2. ASR errors are another difficult problem for speech NE extraction. Context-based model help identify partial information about candidate NEs (in particular, their positions and types). True NEs can be identified from topic-relevant documents, which are retrieved using confidently recognized NE context in the ASR hypothesis.

3. For NE translation, frequent NEs can be reliable translated using bilingual lexicon and sentence-aligned corpus. Less frequently occurring NEs are the most difficult to translate. Accessing additional monolingual information via information retrieval and utilizing translingual NE similarity features significantly improve the NE translation accuracy.

4. Effectively capturing origin-specific transliteration patterns, the cluster-specific name transliteration substantially improves transliteration accuracy.

5. The proposed information-theoretic framework estimates NE information loss from speech recognition and machine translation. It provides an alternative to existing word-matching NE extraction and translation evaluation metrics such as precision, recall, F-score, as well as Bleu and NIST scores for machine translation evaluation.

9.3 Discussion and Future Research Directions

Although we have developed a series of approaches to various NE extraction and translation problems, this problem has not been fully solved. There remain many intriguing research problems which can be further explored. Here we propose some possible directions for future research:

9.3.1 Improve Monolingual NE Tagging with Crosslingual NE Alignment

NE tagging is one of the key components for NE translation. When manually NE annotation data is not enough, bootstrapping method using unlabeled data becomes more attractive. We demonstrate the monolingual bootstrapping in section 3.3. Similarly, crosslingual bootstrapping based on NE alignment is possible.

In the sentence-aligned bilingual corpus, NEs are automatically tagged for each language. As experiment results in section 5.1 showed, automatic NE tagging errors severely affected NE alignment quality. We applied a variable-length sliding window around the initial NE boundary to correct some initial NE tagging errors. As a result, the NE-aligned bilingual corpus contained less NE tagging errors for both languages. Using these error-corrected NE tagging data, one can train more accurate NE tagger for each language. One may apply the retrained NE taggers on the parallel corpus again and hopefully further improve the NE tagging quality and NE alignment accuracy, thus reduce more NE tagging errors and obtain parallel data with higher NE label accuracies.

9.3.2 Search World Wide Web for NE Translation

In Section 5.3 we searched a pre-indexed monolingual corpus for NE translation. This strategy was appropriate for our research since we could query with arbitrary number of words, experimented with different indexing units: sentences, paragraphs or documents. We could also evaluate the NE transla-

tion coverages of different corpora. However, the NE translation accuracy was also hindered by the limited information included in the monolingual corpus. Some articles have been outdated for ten years and they did not provide satisfactory coverage to translate recent NEs.

One can search a much larger and constantly updated corpus, the World Wide Web. The Web contains tremendous resources, constantly updating and increasing. There are several search engines providing APIs for web search applications, but they often set constraints on query length, the maximum number of query words one can ask. So one needs to find the most discriminating words to formulate effective queries, such as topic-relevant NEs or context words. These query words are critical to efficiently find a small number of target language documents containing the NE translation. One can make use of various information from web search for NE translation, such as the number of returned results (for NE translation verification), top N returned snippets (the most specific documents but possibly with less translation coverage), returned URLs and the whole document from each URL, which are more complete but more computationally expensive to identify the translation.

Additionally, rather than only searching target language web pages, one may search mixed-language web pages (Zhang *et al.* (2005) and Huang *et al.* (2005b)) that contain both a source NE and its translation in the same web page. One can extract NE translations from the returned web pages based on their phonetic, semantic, context similarities as well as occurrence frequencies. This approach can be extended to translate not only NEs, but also other words or phrases which we do not have reliable translations given existing bilingual resources.

9.3.3 Measure Acoustic Similarity for Speech NE Error Correction and Extraction

In section 6.2.3, we calculated the string transliteration cost between the recognized NE hypothesis and the retrieved candidate NE, which implicitly estimated their phonetic similarity. Due to speech recognizer errors, the pronun-

ciation of the recognized NE hypothesis could be very different from the true word(s), thus the phonetic comparison may be unreliable. One may explicitly model the pronunciation similarity by calculating the speech recognizer's confusion matrix, which represents the most confusable and often misrecognized words, characters, or phonemes. With such matrix, one can generate phonetically similar words or characters for the hypothesized NEs. Even if the top one hypothesized NE is phonetically incorrect, there are other variants which the retrieved candidate NEs can be compared with. Similarly, ASR N-best hypotheses or confusion network can be used to generate more hypothesized NEs.

9.3.4 NE Extraction and Translation Evaluation Method

We evaluated NE extraction and translation performances using precision, recall and F-scores. When applying to machine translation task, we adopted the widely used Bleu and NIST scores. However, all these evaluation metrics are based on word matching. A phonetically equivalent name translation is considered incorrect and awarded no credit even if its spelling is only slightly different from the reference translation, although for human readers that's perfectly acceptable. We proposed an initial information-theoretic metric to measure the actual NE information loss to human readers, based on the context of NEs. One may expand this metric by additionally considering information noise caused by incorrect NE recognition and translation. To evaluate the information loss for general machine translation task, one may combine the information-theoretic measure with the word alignment information between the hypothesis and reference sentences, as described in METEOR(Banerjee & Lavie (2005)).

Appendix A

Penn TreeBank Part-of-Speech Tag Set

A. PENN TREEBANK PART-OF-SPEECH TAG SET

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Table A.1: Penn TreeBank Part-of-Speech Tag Set

Bibliography

AL-ONAIZAN, Y. & KNIGHT, K. (2002). Translating named entities using monolingual and bilingual resources. In *ACL*, 400–408. 2.4

APPELT, D., HOBBS, J., ISRAEL, D. & TYSON, M. (1993). Fastus: A finite-state processor for information extraction from real world texts. In *Proceeding of IJCAI-93*. 2.2.1, 6.2

ARBABI, M., FISCHTHAL, S.M., CHENG, V.C. & BART, E. (1994). Algorithms for arabic name transliteration. *IBM Journal of Research and Development*, **38**, 183. 2.4

BANERJEE, S. & LAVIE, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan. 9.3.4

BIKEL, D.M., MILLER, S., SCHWARTZ, R. & WEISCHEDEL, R. (1997). Nymble: a high-performance learning name-finder. In *Proceedings of Applied Natural Language Processing*, 194–201. 2.2.2, 3.1, 3.1, 6.2

BLUM, A. & MITCHELL, T. (1998). Combining labeled and unlabeled data with cotraining. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, ACM. 7.2.2

BORTHWICK, A. (1999). *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University. 2.2.2

BRILL, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, **21**, 543–565. 2.2.2

BROWN, P.F., COCKE, J., PIETRA, S.A.D., PIETRA, V.J.D., JELINEK, F., LAFFERTY, J.D., MERCER, R.L. & ROOSSIN, P.S. (1990). A statistical approach to machine translation. *Comput. Linguist.*, **16**, 79–85. 2.3.3.2, 7.1.1

BROWN, P.F., PIETRA, V.J.D., PIETRA, S.A.D. & MERCER, R.L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19**, 263–311. 2.3.3.2, 4.2, 4.3.2, 5.2.1

BROWN, R. (2002). Example-based machine translation, a tutorial. AMTA Tutorials. 2.3.3.1

BROWN, R.D. (2000). Automated generalization of translation examples. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000)*, 125–131, Saarbr‘‘
cken, Germany. 2.3.3.1

CALIFF, M.E. & MOONEY, R.J. (1997). Relational learning of pattern-match rules for information extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, 9–15, Madrid, Spain. 2.2.1

CARRERAS, X., MÀRQUES, L. & PADRÓ, L. (2002). Named entity extraction using adaboost. In *Proceedings of CoNLL-2002*, 167–170, Taipei, Taiwan. 2.2.2

CHENG, P.J., TENG, J.W., CHEN, R.C., WANG, J.H., LU, W.H. & CHIEN, L.F. (2004). Translating unknown queries with web corpora for crosslingual information retrieval. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR '04)*, 146–153, ACM Press, Sheffield, United Kingdom. 2.4

CHIANG, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 44th Annual Meeting on Association for Computational Linguistics*, 263–270, Association for Computational Linguistics. 2.3.3.2

CHINCHOR, N. (1998). Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference(MUC7)*. 1.1

COLLINS, M. (2001). Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 489–496, Association for Computational Linguistics, Morristown, NJ, USA. 2.2.2

DEMPSTER, A.P., LAIRD, N.M. & RUBIN, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, **B**, 39. 4.1

DODDINGTON, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference (HLT)*, San Diego, CA. 5.1.4

DORR, B.J., JORDAN, P.W. & BENOIT, J.W. (1998). A survey of current paradigms in machine translation. Tech. Rep. CS-TR-3961. 2.3

FARWELL, D. & WILKS, Y. (1990). Ultra: A multi-lingual machine translator. Technical Report MCCS-90-202, Computing Research Laboratory, New Mexico State University. 2.3.1

FLORIAN, R., ITTYCHERIAH, A., JING, H. & ZHANG, T. (2003). Named entity recognition through classifier combination. In W. Daelemans & M. Osborne, eds., *Proceedings of CoNLL-2003*, 168–171, Edmonton, Canada. 2.2.2

GRISHMAN, R. (1997). Information extraction: Techniques and challenges. *Summer Convention on Information Extraction (SCIE)*, 10–27. 2.1, 2.2.1

GRISHMAN, R. & SUNDHEIM, B. (1995). Design of the muc-6 evaluation. In *Proceedings of MUC-6*. 6.2

HUANG, F. & VOGEL, S. (2002). Improved named entity translation and bilingual named entity extraction. In *Proceedings of the 2002 International Conference on Multimodal Interfaces (ICMI '02)*. 2.4

HUANG, F., ZHANG, Y. & VOGEL, S. (2005a). Mining key phrase translations from web corpora. In *Proceedings of the HLT-EMNLP 2005*, Vancouver, BC, Canada. 2.4

HUANG, F., ZHANG, Y. & VOGEL, S. (2005b). Mining key phrase translations from web corpora. In *the Proceedings of the Human Language Technology and Empirical Methods for Natural Language Processing (HLT-EMNLP)*, Vancouver, BC, Canada. 5.3.2, 9.3.2

KNIGHT, K. & GRAEHL, J. (1997). Machine transliteration. In P.R. Cohen & W. Wahlster, eds., *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 128–135, Association for Computational Linguistics, Somerset, New Jersey. 2.4, 7

KUBALA, F., SCHWARTZ, R., STONE, R. & WEISCHEDEL, R. (1998). Named entity extraction from speech. In *DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA. 6.2

LAFFERTY, J., MCCALLUM, A. & PEREIRA, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learnning*. 2.2.2

LAVIE, A., LANGLEY, C., WAIBEL, A., LAZZARI, G., PIANESI, F., COLETTI, P., BALDUCCI, F. & TADDEI, L. (2001). Architecture and design considerations in nespole!: a speech translation system for e-commerce applications. In *Proceedings of HLT 2001 Human Language Technology Conference*, San Diego, California. 2.3.4

LAVIE, A., VOGEL, S., LEVIN, L., PETERSON, E., PROBST, K., FONT, A., REYNOLDS, R., CARBONELL, J. & COHEN, R. (2003). Experiments with a hindi-to-english transfer-based mt system under a miserly data scenario. 2.3.2

LAWRENCE, S. & GILES, C.L. (1999). Accessibility of Information on the Web. *Nature*, **400**, 107–109. 1.1

LYMAN, P., VARIAN, H.R., CHARLES, P., GOOD, N., JORDAN, L.L. & PALE, J. (2003). How much information? 2003. 1.1

MANNING, C.D. & SCHUTZE, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press. 7.1.1, 7.3.1

MARCU, D. & WONG, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadephia, PA. 7.3.1

MCCALLUM, A. & LI, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL)*. 2.2.2

MELAMED, I.D. (2000). Modles of translational equivalence among words. *Computational Linguistics*, **26(2)**, 221–249. 5.1.2

MENG, H., LO, W.K., CHEN, B. & TANG, K. (2001). Generating phonetic cognates to handle named entities in english-chinese cross-language spoken document retrieval. In *Proceedings of the ASRU-2001*, Trento, Italy. 2.4, 7

MILLER, D., BOISEN, S., SCHWARTZ, R., STONE, R. & WEISCHEDEL, R. (2000). Named entity extraction from broadcast news. In *the sixth conference on Applied Natural Language Processing*, 316–324, Seattle, WA. 6.2

MITAMURA, T., NYBERG, E. & CARBONELL, J. (1991). An efficient interlingua translation system for multi-lingualdocument production. 2.3.1

MOORE, R.C. (2003). Learning translations of named-entity phrases from parallel corpora. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. 2.4

NEY, H. (1999). Speech translation: Coupling of recognition and translation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 517–520, Phoenix, AR. 2.3.4

NGAI, G. & FLORIAN, R. (2001). Transformation-based learning in the fast lane. In *Proceedings of NAACL'01*, 40–47, Pittsburgh, PA. 2.2.2

OARD, D. (2003). The surprise langauge exercises. *ACM Transactions on Asian Language Information Processing*, **2**. 5.2

OCH, F.J., TILLMANN, C. & NEY, H. (1999). Improved alignment models for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 20–28, University of Maryland, College Park, MD. 2.3.3.2, 7.3.1

OGILVIE, P. & CALLAN, J. (2001). Experiments using the lemur toolkit. In *In Proceedings of the 2001 Text REtrieval Conference (TREC 2001),National Institute of Standards and Technology, special publication 500-250.*, 103–108. 5.3.1

PALMER, D., OSTENDORF, M., & BURGER, J. (2000). Robust information extraction from automati-cally generated speech transcriptions. *Speech Communication*, **32**, 95–109. 6.2

PAPINENI, K., ROUKOS, S., WARD, T. & ZHU, W.J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics*, 311–318. 2.3.3.2, 5.1.4

RILOFF, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artifical Intelligence*, 1044–1049. 2.2.1

ROBINSON, P., BROWN, E., BURGER, J., CHINCHOR, N., DOUTHAT, A., FERRO, L. & HIRSCHMAN, L. (1999). Overview: Information extraction from broadcast news. In *Proceedings of DARPA Broadcast News Workshop*, 27–30. 6.1

S. MILLER, H.F.L.R.R.S.R.S.R.W., M. CRYSTAL & THE ANNOTATION GROUP (1998). Bbn: Description of the sift system as used for muc-7. In *Proceedings of 7th Message Understanding Conference*, Fairfax, VA. 2.2.2

SEKINE, S., GRISHMAN, R. & SHINNOU, H. (1998). A decision tree method for finding and classifying names in japanese texts. In *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Cananda. 2.2.2

SHERMAN, C. (2001). Google Fires New Salvo in Search Engine Size Wars. http://searchenginewatch.com/searchday/article.php/2158371. 1.1

STALLS, B. & KNIGHT, K. (1998). Translating names and technical terms in arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*, Montreal, Quebec Canada. 2.4, 7

TANG, M., LUO, X. & ROUKOS, S. (2002). Active learning for statistical natural language parsing. In *ACL 2002*. 3.3

UCHIDA, H. (1985). Fujitsu machine translation system atlas. In *Proc. of International Sympositon MT*. 2.3.1

VENUGOPAL, A., VOGEL, S. & WAIBEL, A. (2003). Effective phrase translation extraction from alignment models. In *ACL*, 319–326. 2.3.3.2

VIRGA, P. & KHUDANPUR, S. (2003). Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL-2003 Workshop on Multi-lingual Named Entity Recognition*, Japan. 7, 7.3.3

VITERBI, A. (1967). Error bound for convolutional codes and asymptotically optimum decoding algorithm. *IEEE Transaction on Information Theory*, **13**, 260–269. 3.1

VOGEL, S., NEY, H. & TILLMANN, C. (1996). Hmm based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, Copenhagen, Denmark. 2.3.3.2, 4.2

VOGEL, S., ZHANG, Y., HUANG, F., TRIBBLE, A., VENOGUPAL, A., ZHAO, B. & WAIBEL, A. (2003). The CMU statistical translation system. In *Proceedings of MT Summit IX*, New Orleans, LA. 2.3.3.2, 5.1.4, 5.2.1, 5.3.4.2, 7.3.1

WAHLSTER, W., ed. (2000). *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin. 2.3.4

WAIBEL, A., LAVIE, A. & LEVIN, L.S. (1997). Janus: A system for translation of conversational speech. *Künstliche Intelligenz*. 2.3.1

WU, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, **23**, 377–404. 2.3.3.2, 7.3.1

WU, D., NGAI, G. & CARPUAT, M. (2003). A stacked, voted, stacked model for named entity recognition. In W. Daelemans & M. Osborne, eds., *Proceedings of CoNLL-2003*, 200–203, Edmonton, Canada. 2.2.2

YAMADA, K. & KNIGHT, K. (2001). A syntax-based statistical translation model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 523–530, Association for Computational Linguistics, Morristown, NJ, USA. 2.3.3.2

YAROWSKY, D. & NGAI, G. (2001). Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *the Proceedings of NAACL*, 200–207. 2.4, 5.2.1

ZECHNER, K. (2001). Automatic summarization of spoken dialogs in unrestricted domains. In *Ph.D Thesis, Language Technology Institute, Carnegie Mellon University*. 6

ZHAI, L., FUNG, P., SCHWARTZ, R., CARPUAT, M. & WU, D. (2004). Using n-best lists for named entity recognition from chinese speech. In *the Proceedings of the HLT/NAACL 2004*, Boston, MA. 6.2

ZHANG, Y. & VINES, P. (2004). Using the web for automated translation extraction in cross-language infor-mation retrieval. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR '04)*, 162–169, ACM Press, Sheffield, United Kingdom. 2.4

ZHANG, Y. & VOGEL, S. (2005). An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05)*, The European Association for Machine Translation, Budapest, Hungary. 7.2.2

ZHANG, Y., VOGEL, S. & WAIBEL, A. (2003). Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proceedings*

of International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'03), Beijing, China. 2.3.3.2

ZHANG, Y., HUANG, F. & VOGEL, S. (2005). Mining translations of oov terms from the web through cross-lingual query expansion. In *the Proceedings of the 28th Annual International ACM SIGIR*, Salvador, Brazil. 5.3.2, 9.3.2

ZHAO, B. & VOGEL, S. (2003). Word alignment based on bilingual bracketing. In R. Mihalcea & T. Pedersen, eds., *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, 15–18, Association for Computational Linguistics, Edmonton, Alberta, Canada. 2.3.3.2

ZHOU, G. & SU, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, PA. 2.2.2