Satellite Workshop

On

Language, Artificial Intelligence and Computer Science for Natural Language Processing Applications (LAICS-NLP)

October 19, 2006
Department of Computer Engineering
Faculty of Engineering
Kasetsart University, Bangkok, Thailand.

http://naist.cpe.ku.ac.th/LAICS-NLP/

Organising Chair :
Dr. Bali Ranaivo Penang, Malaysia.

On behalf of the organising committee, I would like to express a very warm welcome to all the participants of the Workshop organised within the LAICS-NLP Summer School 2006. For the first time, a Summer School on "Language, Artificial Intelligence and Computer Science for Natural Language Processing applications (LAICS-NLP)" could take place in South-East Asia. It has made possible thanks to the initiative of researchers involved in the STIC-ASIA project "Multilingual Language Processing".

During this one week Summer School, one day workshop has been dedicated to students, researchers and practioners from Asia to give them the opportunity of presenting their works. The objective is to make possible the meetings and discussions between the participants who are involved directly or indirectly in the field of computational linguistics and natural language processing. We can say that it is a success as papers submitted and accepted come from Nepal, India, Bangladesh, Thailand, Malaysia, Singapore and Indonesia. The variety of the topics to be presented proves that Asian countries are interested in the development of language engineering systems, and therefore are ready to participate in any international project.

One day workshop is not sufficient regarding the number of papers submitted. The organising committee had to choose papers original by their contents or reflecting advanced researches. A very hard task was to choose the three best papers, submitted by undergraduate and postgraduate students, allowing their first presenters to get free registration to attend the LAICS-NLP Summer School. I would like to congratulate personally the authors of the three best papers which come from India, Malaysia, and Indonesia. I wish them the best for their career as researcher.

This event could not be made possible without the international collaboration between France, India, Malaysia, and Thailand. I would like to express my gratitude and appreciation to all members of the organising committee, authors, and the wonderful NAIST team (Specialty Research Unit in Natural Language Processing and Intelligent Information System Technology) which had the hard task to manage most of administrative problems. Because most events cannot be realised without valuable sponsorships, I would like to sincerely thank the French Ministry of Foreign Affairs, French Embassies in India and Thailand, CNRS (the French National Center for Scientific Research, France), INRIA (the French National Institute for Research in Computer Science and Control), NECTEC (the Thai National Electronics and Computer Technology Center), and Kasetsart University for their financial supports and trusts on this Summer School and workshop.

I really hope that this will not be the first and last time that this kind of event is organised in Asia. International collaborations are needed if we want to move one step ahead and be part of the development of the world.

I wish to have another "Monsoon School" next year!

Ranaivo-Malançon Bali
Chairwoman
LAICS-NLP Summer School Workshop
Kasetsart University, Bangkok, Thailand
19 October 2006

Satellite Workshop

On

Language, Artificial Intelligence and Computer Science for Natural Language Processing Applications (LAICS-NLP)

October 19, 2006
Department of Computer Engineering
Faculty of Engineering
Kasetsart University, Bangkok, Thailand.

http://naist.cpe.ku.ac.th/LAICS-NLP/

Organising Chair

- Bali Ranaivo, USM, Penang, Malaysia

Program Committees

- Asanee Kawtrakul, Kasetsart Univ., Bangkok, Thailand
- Claire Gardent, LORIA, Nancy, France
- Eli Murguia, IRIT, Toulouse, France
- Farah Benamara, IRIT, Toulouse, France
- Leila Amgoud, IRIT, Toulouse, France
- Monojit Choudhury, IIT, Kharagpur, India
- Patrick Saint-Dizier, IRIT, Toulouse, France
- Sudeshna Sarkar, IIT, Kharagpur, India
- Tang Enya Kong, USM, Penang, Malaysia
- Thanaruk Theeramunkong, SIIT, Bangkok, Thailand
- Thierry Poibeau, LIPN, Paris, France
- Virach Sornlertlamvanich, TCL, Bangkok, Thailand

Local Organisers:

- Asanee Kawtrakul
- Mukda Suktarachan
- Patcharee Varasrai
- Achara Napachot
- Areerat Thongbai
- Taddao Kleepikul
- Chaikorn Yingsaree
- Vasuthep Khunthong
- Phukao Soraprasert
- Aurawan Imsombut
- Sutee Sudprasert
- Chalathip Thumkanon
- Thana Sukwaree
- Chalermpon Sirigayon
- Chaveevan Pechsiri
- Vee Satayamas
- Rapepun Piriyakul

Main Objectives

Language processing is now a major field in Computer science, with large scale applications, which are still under intense research, such as question-answering, machine translation, automatic summarization, etc. most of them in a multilingual setting. Language processing technology requires knowledge from a large variety of disciplines: applied linguistics, computer science and artificial intelligence, ergonomics and the science of interaction, psychology, etc.

The goal of this summer school, in a short period of time, is to present the foundations and the most recent advances of the different topics of interest to any language processing practitioner, with in view the development of well targeted applications. This summer school is more application oriented than most western summer schools such as ESSLII, most notably.

Besides ***courses***, a ***1 day workshop*** will be organized in the middle of the school where groups or individuals attending will have the opportunity to present their work. The objective is to enhance cooperation and to have a better view of what's being done in Asia in computational Linguistics.

Workshop Schedule

8.30 - 8.45	Opening	
8.45 - 9.05	Indra Budi	Information Extraction for the Indonesian Language
9.05 - 9.25	Chaveevan Pechsiri, Asanee Kawtrakul	Causality Knowledge Extraction based on Causal Verb Rules
9.25 - 9.45	Asif Ekbal	Named Entity Recognition in Bengali
9.45 - 10.05	Lim Lian Tze, Nur Hussein	Fast Prototyping of a Malay WordNet System
10.05 - 10.25	Aurawan Imsombut, Asanee Kawtrakul	Taxonomic Ontology Learning by using Item List on the Basis of Text Corpora in Thai
10.25 - 10.45	Ong Siou Chin, Narayanan Kulathuramaiyer, Alvin W. Yeo	Discovery of Meaning from Text
10.45 - 11.00	Tea/Coffee Break	
11.00 - 11.20	Swati Challa, Shourya Roy, L. Venkata Subramaniam	Analysis of agents from call transcriptions of a car rental process
11.20 - 11.40	Loh Chee Wyai, Alvin W. Yeo, Narayanan K.	Integration Techniques for multimodal Speech and Sketch map-based system
11.40 - 12.00	Vishal Chourasia	Phonological rules of Hindi and Automatic Generation of Pronunciation Dictionary for Speech Recognition
12.00 - 12.20	Ayesha Binte Mosaddeque	Rule based Automated Pronunciation Generator
12.20 - 12.40	Sourish Chaudhuri	Transliteration from Non-Standard Phonetic Bengali to Standard Bengali
12.40 - 13.00	Bal Krishna Bal	The Structure of Nepali Grammar
13.00 - 14.00	Lunch	
14.15 - 14.35	Rahul Malik, L. Venkata Subramaniam, Saroj Kaushik	Email Answering Assistant for Contact Centers
14.35 - 14.55	Shen Song, Yu-N Cheah	Extracting Structural Rules for Matching Questions to Answers
14.55 - 15.15	Rapepun Piriyakul, Asanee Kawtrakul	"Who" Question Analysis
15.15 - 15.30	Tea/Coffee Break	
15.30 - 15.50	Stephane Bressan, Mirna Adriani, Zainal A. Hasibuan, Bobby Nazief	Mind Your Language: Some Information Retrieval and Natural Language Processing Issues in Development of an Indonesian Digital Library
15.50 - 16.10	Suhaimi Ab. Rahman, Normaziah Abdul Aziz, Abdul Wahab Dahalan	Searching Method for English-Malay Translation Memory Based on Combination and Reusing Word Alignment Information
16.10 - 16.30	Sudip Kumar Naskar	A Phrasal EBMT System for Translating English to Bengali
16.30 - 16.50	Zahrah Abd Ghafur	Prepositions in Malay: Instrumentality
16.50 - 17.10	Patrick Saint-Dizier	"Multilingual Language Processing" (STIC-Asia project)

Contents

Information Extraction for the Indonesian Language

Indra Budi

Faculty of Computer Science University of Indonesia
Kampus UI Depok 16424
Email: indra@cs.ui.ac.id

Abstract
A modern digital library should be providing effective integrated access to disparate information sources. Therefore the extraction of semi-structured or structured information - for instance in XML format - from free text is one of the great challenges in its realization. The work presented here is part of a wider initiative aiming at the design and development of tools and techniques for an Indonesian digital library. In this paper, we present the blueprint of our research on information extraction for the Indonesian language. We report the first result of our experiments on name entity recognition and co-reference resolution.

1. Introduction

The purpose of information extraction (IE) is to locate and to extract specific data and relationships from texts and to represent them in a structured form [7, 8]. XML is a particularly suited candidate for the target data model thanks to its flexibility in representing data and relationships and to its suitability to modern Internet applications.

Indeed, IE is potentially at the heart of numerous modern applications. For instance and to name a few, IE can be used in software engineering to generate test cases from use case scenario; in database design, IE can be used to generate Entity Relationship Diagrams from analysis cases; in the legal domain, IE can be used to extract patterns from legal proceedings. The list is open-ended, however, we choose for this article and for the sake of simplicity a domain that can be apprehended by the non-expert: we try and extract information about events that are meetings from news articles. A meeting is an event for which we wish to identify the location (place, city and country), the date (day, month, year) and the list of participants (name, quality and nationality). Fig 1.1 illustrates the expected output corresponding to the following sample text.

*Menteri Luar Negeri Inggris **Mike O'Brien**[1] kemarin berada di Jakarta. **Dia**[2] bertemu dengan **Megawati Soekarnoputri**[3] di Istana Negara. **Megawati**[4] adalah wanita pertama yang menjadi presiden di Indonesia.*

(British Foreign Office Minister **Mike O'Brien** had been in Jakarta yesterday. **He** held meeting with **Megawati Soekarnoputri** at the State Palace. **Megawati** is the first woman who become president in Indonesia)

The components highlighted in italic in Fig 1.1 require global, ancillary, or external knowledge.

We need models and techniques to recognize named entities and their relationships. There are two main approaches in building rules and pattern for the information extraction task, namely, knowledge engineering and machine learning [1].

```
<meeting>
  <date>05/12/2003</date>
  <location>
    <name>Istana Negara</name>
    <city>Jakarta</city>
    <country>Indonesia</country>
  </location>
  <participants>
    <person>
      <name>Megawati Soekarnoputri</name>
      <quality>Presiden</quality>
      <country>Indonesia</country>
    </person>
    <person>
      <name>Mike O'Brien</name>
      <quality>Menteri Luar Negeri</quality>
      <country>Inggris</country>
    </person>
  </participants>
</meeting>
```

Fig 1.1 Structured information in XML

In a knowledge engineering approach experts handcraft an instance of a generic model and technique. In a machine learning approach, the instance of the model and technique is learned from examples with or without training and feedback.

Following [5], we consider that the information extraction process requires the following tasks to be completed: named entity recognition, co-reference resolution, template element extraction and scenario template extraction. Named entity recognition (NER) identifies names of and references to persons, locations, dates and organization from the text, while co-reference resolves references and synonymies. Template element extraction completes the description of each entity by adding, in our example, quality and nationality to

persons for instance. Finally, scenario template extraction associates the different entities, for instance, the different elements composing an event in our example. The extraction tasks are usually leveraging several features the most essential of which being linguistic. These include morphology, part of speech of terms, and their classification and associations in thesauri and dictionaries. It also leverages the context in which terms are found such as neighboring terms and structural elements of the syntactical units - propositions, sentences, and paragraphs, for instance. Clearly, because of the morphological and grammatical differences between languages, the useful and relevant combinations of the above features may differ significantly from one language to another. Techniques developed for the English language need to be adapted to indigenous linguistic peculiarities. It is also possible that entirely new and specific techniques need to be designed.

Our research is concerned with the design and implementation of information extraction suite of tools and techniques for the Indonesian language and to study the genericity and peculiarities of the task in various domains. We report in this paper our first results in the comparison of knowledge based and machine learning based named entity recognition as well as a first attempt of co-reference resolution.

2. Named Entity Recognition (NER)

In our running example, the NER task should identify that *Mike O'Brien* and *Megawati Soekarnoputri* are persons, *Istana Negara* and *Jakarta* are locations (and possibly that the former is a place and the latter a city), *Presiden* and *Menteri Luar Negeri* are qualities of persons

2.1 Approaches

Approaches to named entity recognition (see [1]) can be classified into two families: knowledge engineering approaches and machine learning approaches.

Knowledge engineering approaches are expert-crafted instances of generic models and techniques to recognize named entity in the text. Such approaches are typically rule-based. In a rule-based approach the expert design rules to be used by a generic inference engine. The rule syntax allows the expression of grammatical, morphological and contextual patterns. The rules can also include dictionary and thesauri references. For example, the following rule contributes to the recognition of persons.

If a proper noun is preceded by a title **then** the proper noun is name of person

We have asked educated native speakers to design rules combining contextual, morphological, and part of speech features that assign classes to terms and groups of terms in the text. They based their work on the analysis of a training corpus.

In machine learning approaches, a generic computer program learns to recognize named entities with or without training and feedback. General machine learning models exist that do not necessitate the mobilization of expensive linguistic expert knowledge and resources.

Using a training corpus, in which terms and groups of terms are annotated with the class they belong to, and a generic association rule mining algorithm, we extracted association rules combining the identified contextual, morphological, and part of speech features. For example, if a sequence of terms $<t_1, t_2>$ occurs in the training corpus, where f2 is an identified feature of t2 and nc2 is the name class of t2. We obtain a rule of the following form where support and confidence are computed globally.

$$<t_1, f_2> => nc_2, \textit{(support, confidence)}$$

If the training corpus contains the sentence: "*Prof. Hasibuan conducted a lecture on information retrieval*" in which the term "Hasibuan" is of class person, we produce a rule of the following form.

⟨Prof., Capitalized_word(X)⟩ => person_named(X)

The rule support and confidence depend on the occurrences of the expression "*Prof. X*" with X a person or not in the training corpus. The complete list of rule forms and features used can be seen in [2].

In both approaches, the rules produced are then used for NER. The NER process consists of the following stages: tokenization, feature assignment, rule assignment and name tagging. The left hand side of a rule is the pattern. The right hand side of a rule is the identified named entity class. The following is an example of an actual rule as encoded in the implemented tested.

```
IF    Token[i].Kind="WORD" and Token[i].OPOS and
      Token[i+1].Kind="WORD" and Token[i+1].UpperCase
      and Token[i+1].OOV
THEN  Token[i+1].NE = "ORGANIZATION"
```

The tokenization process identifies tokens (words, punctuation and other units of text such as numbers etc.) from the input sentence. The feature assignment component labels the tokens with their features: the basic contextual features (for instance identifying preposition, days, or titles), the morphological features, as well as the part of speech classes. See [3] for a complete list of features and details of the labeling process. The rule assignment component selects the candidate rules for each identified token in the text.

The rule is then applied and terms and group of terms are annotated with XML tags. The syntax of the tags follows the recommendation of MUC [9]. The following is the output of the system for the second sentence in our running example.

<ENAMEX TYPE=PERSON">Megawati</ENAMEX>
adalah wanita pertama yang menjadi presiden di <ENAMEX
TYPE="LOCATION">Indonesia</ENAMEX>.

2.2 Performance Evaluation

We comparatively evaluate the performance of the two approaches with a corpus consisting of 1.258 articles from the online versions of two mainstream Indonesian newspaper Kompas (kompas.com) and Republika (republika.co.id). The corpus includes 801 names of person, 1.031 names of organization, and 297 names of location.

In order to measure and compare the effectiveness of the two approaches we use the recall, precision and F-Measure metrics as defined in [6] as a reference for the Message Understanding Conference (MUC)

On the one hand, our results confirm and quantify the expected fact that the knowledge engineering approach performs better (see table 2.1) than the machine learning approach. Of course, this comes at the generally high cost of gathering, formalizing and validating expert knowledge. The machine learning approach, on the other hand, yields respectable performance with minimum expert intervention (it only requires the annotation of the training corpus).

Table 2.1. Knowledge-engineering versus machine learning method for NER

Method	Recall	Precision	F-Measure
Knowledge Engineering	63.43%	71.84%	67.37%
Machine Learning	60.16%	58.86%	59.45%

A finer grain analysis the results, manually going through the correct, partial, possible, and actual named entities (See [3] for definitions), seems to indicate that the machine learning approach induces more partial recognition. This is avoided by the knowledge engineering approach, which allows a more effective usage of the variety of features available.

3. Co-reference Resolution

Co-reference resolution attempts to cluster terms or phrases that refer to the same entity (markables) [7,9]. Terms or phrases are pronouns or entities that have been recognized in a named entity recognition phase.

In our running example, the co-reference resolution process should identify that Dia_2 refers to *Mike O'Brien*$_1$ and that *Megawati*$_4$ refers to *Megawati Soekarnoputri*$_3$. In other words, the system should produce two clusters: {*Mike O'Brien*$_1$ *and Dia*$_2$} and {*Megawati Soekarnoputri*$_3$ *and Megawati*$_4$}.

3.1 Approaches

Our first attempt to implement co-reference algorithms aimed at comparing two machine learning methods: an original method based on association rules and a state of the art method based on decision trees. Both methods use the same features. We consider nine different features. See [4] for details of features.

The state of the art method [10] is based on decision trees and the C4.5 algorithm. Each node of the tree corresponds to a decision about a particular feature and leafs are Booleans represent co-references.

We also devised an original association rule method. We mine association rules that are capable of testing the pairwise equivalence of markables. The association rules are obtained from a training corpus and selected because their support and confidence are above given thresholds. The rules have the form $X \Rightarrow Y$, where X represents features of a pair of markables and Y is a Boolean indicating whether the markables are co-references or not. Each feature corresponds to an attribute. Therefore the association rules have the following form.

<attr1, attr2, attr3, attr4, attr5, attr6, attr7, attr8, attr9> $\Rightarrow$ *<isEquiv>*

The left-hand-side (LHS) of the rule is a list of values for the attributes for a pair of markables. The right-hand-side (RHS) is the variable *isEquiv*. It is true if the markables are equivalent and false otherwise. Indeed, we consider negative rules that indicate that pairs of markables are not co-referenced.

Fig 3.1 illustrates the general architectures of the system in the testing phase. It is assumed that pronoun tagging and named entity recognition has been done and that association rules are readily available from the training phase. For a pair of markables, if several association rules are applicable, the rule with the highest confidence is applied. If the RHS in rule is true then the markables are marked equivalent and not equivalent otherwise. After all the pairs of markables have been checked then we group the markables that are equivalent. We randomly choose a representative for each class. We can now output a document in which markables are tagged with the representative of their class.

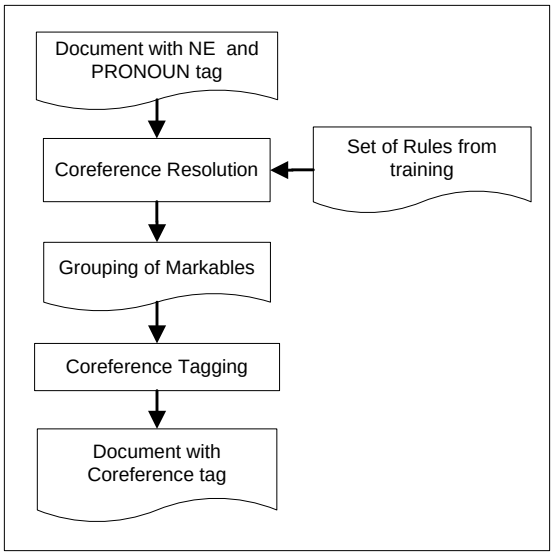


Fig 3.1. The association rules for co-reference resolution system architecture

3.2 Performance Evaluation

We use a corpus of 100 articles from the Indonesian online newspaper Republika (republika.co.id). The articles contain 43,234 words and 5,783 markables consisting of 3,383 named entities (person, location, organization) and 2,400 pronouns in the corpus.

In order to measure and compare the effectiveness of the two approaches we use the recall, precision and F-Measure metrics as defined in [6] as a reference for the Message Understanding Conference (MUC).

Table 3.1 shows the results for both methods. It shows that association rules yield comparable performance to the one of the state-of-the-art method based on decision tree. This result is to be put in the perspective of the one of the previous section in which we saw that the association rule based method performed respectably. This opens the way for a generic association rule based information extraction system.

Table 3.1. Association rules versus decision tree method for co-reference resolution

Method	Recall	Precision	F-Measure
Association Rules	74.38	93.17	82.70
Decision Tree	74.31	93.05	82.60

4. Conclusions and Future Task

We have presented the blueprint of our research project on information extraction for the Indonesian language. We have presented our preliminary results for the tasks of named entity recognition and co-reference resolution. In particular we have compared several techniques based in either knowledge engineering or machine learning, with the objective in mind to find the most economical yet effective and efficient solution to the design of the necessary tools specific to the Indonesian language and to the application domain selected.

At the same time that we explore this compromise and look for a generic and adaptive solution for the Indonesian language, we continue developing the next components of a complete information extraction system starting with ad hoc and state of the art (possibly designed for English or other foreign languages) solutions. We also expect to devise and evaluate novel methods based on association rules, which would give us, if effective and efficient, a uniform framework for all information extraction tasks. The reader notice that these methods do not exclude the utilization of global, ancillary, and external knowledge such as gazetteers document temporal and geographical context, etc.

References:

[1] Appelt, Douglas E. and Israel, David J., "Introduction to Information Extraction Technology", Tutorial in IJCAI-99.

[2] Budi, I. and Bressan, S., "Association Rules Mining for Named entity Recognition" in proceeding of WISE Conference, Roma, 2003.

[3] Budi, I. and et. al, "Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach", in proceedings of the 8th International Conference on Discovery Science, Singapore, October, 2005.

[4] Budi, I., Nasrullah and Bressan, S., "Co-reference Resolution for the Indonesian Language Using Association Rules", submitted to IIWAS 2006.

[5] Cunningham, Hamish., "Information Extraction - a User Guide (Second Edition)", accessed on http://www.dcs.shef.ac.uk/~hamish/IE/userguide/ at 5th March 2003

[6] Douthat, A.: *The Message Understanding Conference Scoring Software User's Manual*, In Proceedings of the 7th Message Understanding Conference (MUC-7), 1998.

[7] Grishman, Ralph., "Information Extraction: Techniques and Challenges" Lecture Notes in Computer Science, Vol. 1299, Springer-Verlag, 1997.

[8] Huttenen, Silja., Yanbarger, Roman., and Grishman, Ralph., "Diversity of Scenarios in Information Extraction", Proceedings of Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain, 2002.

[9] MUC.: MUC-7 Co-reference task definition. Proceedings of the Seventh Message Understanding Conference, 1998.

[10] Soon, W.M., Yong Lim, D. W., and Ng, H.T.: A Machine Learning Approach to Co-reference Resolution of Noun Phrases. Computational Linguistics, Volume 27, (2001) 521-544.

Causality Knowledge Extraction based on Causal Verb Rules

Chaveevan Pechsiri, Asanee Kawtrakul

Department of Computer Engineering,
Kasetsart University
Phaholyothin Rd., Bangkok, Thailand 10900
Tel. +662-942-8555, Fax.: +622-579-0358
e-mail : itdpu@hotmail.com, ak@.ku.ac.th

Abstract

The aiming of this paper is to automatically extract the causality knowledge from documents for the contribution knowledge sources of the question-answering system. This paper is only concern of extracting the causality knowledge from a single sentence or EDU (Elementary Discourse Units) with two problems of the causality identification and the zero anaphora. Then, we propose the usage of causal verbs rules mined from the specified sentence pattern by ID3 to extract the causality knowledge of the single EDU. However, our intra-causal EDU extraction model shows the 0.87 precision and the 0.73 recall.

1 Introduction

Causality knowledge extraction from textual data is an important task to gain useful expressions of Know-Why for the question answering system. There are various forms of causality or cause-effect expression such as in the form of intra-NP, inter-NP, and inter-sentence [Chang and Choi,2004]. In according to our research, we separate the causality knowledge into 2 group based on the elementary discourse unit (EDU) as defined by [Carlson and et al., 2003]. Our EDU is often expressed as a simple sentence or clause. These EDUs will be used to form the causality relation which will be expressed in two forms, an intra-causal EDU and an inter-causal EDU. We define the intra-causal EDU as an expression within one simple EDU with or without an embedded EDU. This is equivalent to the intra-NP form or the inter-NP form[Chang and Choi,2004]. Also, the inter-causal EDU is defined as an expression within more than one simple EDU. Hence, it is equivalent to the inter-sentences of [Chang and Choi, 2004]. However, this paper is a part of our research of causality knowledge extraction and it works on only the intra-causal EDU extraction.

Several techniques [Marcu,1997; Girju and Moldovan,2002; Girju, 2003; Inui and et al.,2004; Chang and Choi,2004] have been used for extracting cause-effect information varying from one sentence to two adjacent sentences. The recent researches [Girju, 2003] uses the causal verbs from the lexico syntactic pattern, 'NP1 cause-verb NP2' pattern, to identify the causal question, yet this has problem with verb ambiguity being solved by learning technique. Later, Chang and Choi [2004] uses NP pair from this lexico syntactic pattern to extract the causality from one sentence. In our work, we are aiming to extract the intra-causal EDU from Thai documents by using the causal verb rules mined from the specified sentence patterns of "NP1 Verb NP2 Preposition NP3" where only NP2 can have null value. The reason that we use this specified pattern is about 50% of the intra-causal EDU, from the corpus behavior study, occurred within this pattern. Thai has specific characteristics, such as, zero anaphora and nominal anaphora. All of these characteristics are involved in the two main problems of causality extraction: the first is how to identify the interesting causality from documents and the second is the zero anaphora. From all of these problems, we need to develop a framework which combines NLP techniques to form the EDU for mining the specified sentence pattern.

In conclusion, unlike other methods where the emphasis is based on the Lexico syntactic pattern [Girju and Moldovan, 2002; Chang and Choi,2004], our research uses the causal verb rules based on the specified sentence pattern to identify causality for the intra-causal EDU

extraction. Our research will be separated into 6 sections. In section 2, related work is summarized. Problems in causality mining from Thai documents will be described in section 3 and in section 4 our framework for causality extraction. In section 5, we evaluate our proposed model and a conclusion in section 6.

2 Related Work

Girju's work [Girju and Moldovan, 2002] consists in finding patterns to be used to extract causal relations from a learning corpus, where the aim of the extraction of causal relations from the inter-noun phrase is to aid in question identification as in [Girju, 2003]. In their research [Girju and Moldovan, 2002], causal verbs were observed, by the pattern <NP1 verb NP2> in documents whereas NP pairs were the causative relationships referenced by WordNet. The causal verbs were used to extract the causal relation with a particular kind of NP; e.g. phenomena NP, *"An earthquake generates Tsunami"*. The problem of research was that the causal verb became ambiguous; e.g. *"Some fungi produce the Alfa toxin in peanut"* was a causal sentence while *"The Century Fox produces movies"* was a non causal sentence. Girju and Moldovan [2002] solved the problem by using a C4.5 decision tree to learn the annotated corpus with syntactic and semantic constraints. The precision of their causality extraction was 73.91%. However, some of our causal verbs in the intra-causal EDUs are expressed as a general verb followed by a preposition, such as 'เป็น..จาก/be..from' , 'ได้รับ.. จาก/get..from', etc., whereas the lexico syntactic patterns can not cover this kind of causal verb, for example: "*ใบเป็นแผลจากเชื้อรา /leaf is scars from fungi.)"*,

Chang and Choi [2002]'s work aimed to extract only causal relations between two events expressed by a lexical pair of NP and the cue phrase with the problems of causality identification. Naïve Bayes classifier was used to solve their problems. They defined the cue phrase used in their work as "a word, a phrase, or a word pattern which connects one event to the other with some relation"; e.q. *"caused by"*, *"because"*, *"as the result of"*, *"Thus"* etc. And their lexical pair was a pair of causative noun phrase, and effective noun phrase that must occur explicitly within one sentence. They obtained 81% of precision for causality extraction. However, there are more than two NPs contained in our intra-causal EDU and this extra NP can not be hold as a part of cue phrase.

Hence, we are aiming at learning causal verb rules from the specified sentence pattern by using ID3 with all five features from the sentence pattern to extract the intra-causal EDU expressions

3 Problems in Causality Extraction

To extract the cause-effect expressions, there are two main problems that must be solved; to identify interesting cause-effect events from Thai documents, and to solve an implicit noun phrase.

3.1 Causality identification

Like many languages, to identify the causality expressions in Thai uses an explicit cue phrase [Chang and Choi , 2002] to connect between cause and effect expressions. In order to avoid non-necessary tasks in whole text analysis, the causal verb which is the linking verb between the causative NP and the effective NP will be used to indicate the cause-effect expression. Although the causal verb is used to identify whether it is a causality or non causality expression, we still have a problem of the causal verb ambiguity. For example:

Causality:

a. *"ใบพืช/**Plant leaf** มี/**has** จุดสีน้ำตาล /**brown sports** จาก/**from** เชื้อรา/**fungi**"*
b. *"คนไข้/**The patient** ตาย/**dies** ด้วย/**with** โรคมะเร็ง/**cancer**"*

Non causality:

c. *"ใบพืช/**Plant leaf** มี/**has** จุดสีน้ำตาล /**brown sports** จาก/**from** โคนใบ/**the leaf base**"*
d. *"คนไข้/**patient** ตาย/**dies** ด้วย/**with** ความสงสัย/**suspicion**"*

This problem of causal verb ambiguity can be solved by learning the EDUs with ID3 from the specified sentence pattern. The result from ID3 learning is plenty of causal verb rules which need to be verified before identifying the intra-causal EDU.

3.2 Zero anaphora or implicit noun phrase

Regardless of whether the noun phrase is in the intra-causal EDU, it will contain the implicit noun phrases; such as zero anaphora. For example:

*"โรคไข้หวัดนก /**The Bird flu disease*** *เป็น /**is*** *โรคที่ สำคัญโรคหนึ่ง /**an important disease .*** *Φ เกิด /**occur*** *จาก / **from*** *ไวรัส H1N5/ **H1N5 virus.** "*

where Φ is zero anaphora = Bird flu disease.

This problem can be solved by using the heuristic rule that the previous subject of the noun phrase will be the ellipsis one.

4 A Framework for Causality Extraction

There are three steps in our framework. First is corpus preparation step followed by causality learning, and causality extraction steps as shown in figure 1.

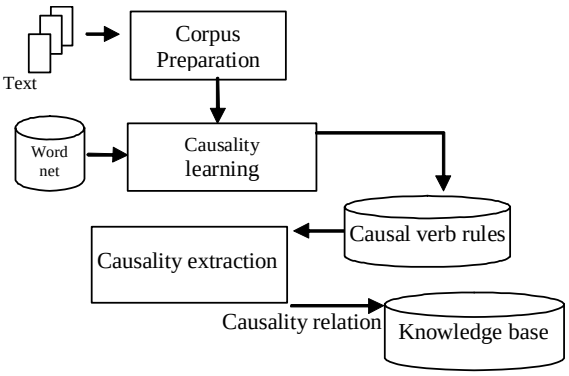


Figure 1. The frame work for causality extraction

4.1 Corpus Preparation

This step is the preparation of corpus in the form of EDU from text. The step involves using Thai word segmentation tools to solve a boundary of a Thai word and tagged its part of speech [Sudprasert and Kawtrakul, 2003], including Name entity [Chanlekha and Kawtrakul, 2004], and word-formation recognition [Pengphom, et al 2002] to solve the boundary of Thai Name entity and Noun phrase. After the word segmentation is achieved, EDU segmentation is then to be dealt with. According to Charoensuk et al. [2005], EDU segmentation will be generated and be kept as an EDU corpus for the next step of learning.

4.2 Causality learning

There are two processes involved in this step, which are Feature annotation for learning, Rule mining, and verification.

4.2.1. Feature annotation for learning

Due to the problems in the intra-causal EDU identification, the causal verb will be used as a feature in this process to extract causality. Because some causal verbs are ambiguity, we have to learn this causal verb feature along with the other four features: NP1, NP2, Preposition, and NP3, from the specified sentence pattern as <NP1 Verb NP2 Preposition NP3>. Then, we manually annotate these five features with the specifying "causality/non causality", and also with their concept from Wordnet after Thai-to English translation to solve the word-sense ambiguity and the variety surface forms of a word with the same concept. If the NP has modifier, the only head noun will be assigned the concept. And, if the NP means "symptom", e.g. 'ใบเหลือง/yellow leaf', 'จุดสีน้ำตาล/brown spot', etc., we will assign its' concept as 'symptom'. The annotation of the intra-causal EDU is shown by the following example:

<EDU><NP1 concept=plant organ>*ใบพืช*
</NP1><Verb concept=have> มี*</Verb><NP2*
concept=symptom> จุดสีน้ำตาล*</NP2><Preposition>*
จาก*</Preposition></NP3 concept=fungi>* เชื้อรา
</fungi>< causality></EDU>

4.2.2 Rule mining

This step is to mine the causal verb rules from the annotation corpus of the intra-causal/non causal EDU by using ID3 from *Weka(http://www.cs.waikato.ac.nz/ml/weka/)*. From this mining step, there are 30 causal verb rules from 330 EDUs of specified sentence pattern, as shown in table1.

4.2.3 Verifying

This step is to verify the rules before giving rise to identify the causal EDU. There are some rules having the same general concept which can be combined into one rule as in the following example:

R1: IF<NP1=*>^<Verb=be>^<NP2=*>^<Prep= จาก/from>^ <NP3= fungi > then causality

R2: IF<NP1=*>^<Verb=be>^<NP2=*>^<Prep= จาก/from>^ <NP3= bacteria> then causality

R3: IF<NP1=*>^<Verb=be>^<NP2=*>^<Prep= จาก/from>^ <NP3=pathogen > then causality

The R3 rule is the general concept rule of R1 and R2. Then, we have 25 rules to be verified. The test corpus from agricultural and health news domains of 2000 EDUs contain 102 EDUs of the specified sentence pattern, which only 87 EDUs are causality.

5 Evaluation

During this research, we have used documents containing 6000 EDUs from the agricultural and health news domains to extract the causal relations. We have divided this corpus into two parts. One part is for learning to determine the intra-causal EDU. The other part is used for evaluating the performance of the causality extraction with the following precision and the recall where R is the causality relation :

$$Recall = \frac{\#\,of\ samples\ correctly\ extracted\ as\ R}{\#\,of\ all\ samples\ holding\ the\ target\ relation\ R}$$

$$Precision = \frac{\#\,of\ samples\ correctly\ extracted\ as\ R}{\#\,of\ all\ samples\ extracted\ as\ being\ R}$$

The results of precision and recall are evaluated by a human. The precision of the extracted causality of the specified sentence pattern is 87% while the recall is 73 %,.

6 Conclusion

However, our model or system will be very beneficial for causal question answering and causal generalization for knowledge discovery.

Acknowledgement

The work described in this paper has been supported by the grant of NECTEC No. NT-B-22-14-12-46-06 and partially supported by a grant from FAO.

References

1.Daniel Marcu. 1997. *The Rhetorical Parsing of Natural Language Texts*, The proc. of the 35th annual meeting of the association for computational linguistics(ACL'97/EACL'97), Madrid, Spain .

2.Du-Seong Chang and Key-Sun Choi.. 2004. *Causal Relation Extraction Using Cue Phrase and Lexical Pair Probabilities*, IJCNLP 2004, Hainan Island, China.

3.George A. Miler, Richard Beckwith, Christiane Fellbuan, Derek Gross, and Katherine Miller. 1993 . *Introduction to Word Net*, An Online Lexical Database .

4.Hutchatai Chanlekha, Asanee Kawtrakul. 2004. *Thai Named Entity Extraction by incorporating Maximum Entropy Model with Simple Heuristic Information*, IJCNLP' 2004 , HAINAN Island , China.

5.Barbara J. Groz , Aravind K. Joshi, and Scott Weinstein. 1995. *Centering: A Framework for Modelling the Local Coherence of Discourse*, In Computational Linguistic 21(2),June 1995, pp. 203-225.

6.Jirawan Chareonsuk, Tana Sukvakree and Aasanee Kawtrakul. 2005. *Elementary Discourse unit Segmentation for Thai using Discourse Cue and Syntactic Information*, NCSEC 2005,Thailand.

7.Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. *Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory*, In Current Directions in Discourse and Dialogue.

8.Marilyn A. Walker, Aaravind K. Joshi, Ellen F. Prince. 1998. *Centering in Naturally Occuring Discource: An Overview , in Centering Theory of Discourse*, Oxford: Calendron Press.

9.Nattakan Pengphon, Asanee Kawtrakul and Mukda Suktarachan. 2002. *Word Formation Approach to Noun Phrase Analysis for Thai*, SNLP2002, Thailand.

10.Roxana Girju and Dan Moldovan. 2002. *Mining Answers for Question Answering* , In proc. of AAI Symposium on Mining Answers from Texts and Knowledge Bases

11.Sutee Sudprasert and Asanee Kawtrakul. 2003. *Thai Word Segmentation based on Global and Local Unsupervised Learning*, NCSEC'2003, Chonburi, Thailand.

12.Takashi Inui, K.Inui and Y Matsumoto. 2004. *Acquiring causal knowledge from text using the connective markers*, Journal of the information processing society of Japan(2004) 45(3)

.

Table1 Show causal verb rules where * means 'any'

Causal verb rules	Example of causality
IF<NP1=*>^<Verb=be>^<NP2=*>^<Prep= จาก/from>^ <NP3=pathogen > then causality	พืช **/Plant** เป็น **/is** โรค **/disease** จาก/ **from** ไวรัส/ **virus** (Plant gets disease from virus)
IF<NP1=*>^<Verb=have>^<NP2=*>^<Prep= จาก/from>^ <NP3=insect > then causality	ใบ**/Leaf** มี **/has** ตำหนิ **/defect** จาก/ **from**เพลี้ย/**aphid** (The leaf has a defect from an aphid)
IF<NP1=*>^<Verb=have>^<NP2=*>^<Prep= จาก/from> <NP3= toxicant food > then causal	ผู้ป่วย**/Patient** มี **/has** อาการท้องเสีย **/diaria symptom** จาก/ **from** อาหารเป็นพิษ/ **food poisoning** (A patient has diaria symptom from food poisoning)
IF<NP1=*>^<Verb=occur>^<NP2=*>^<Prep= จาก/from>^ <NP3*> then causal	โรค **/Disease** เกิด / **occur** จาก **/from** แบคทีเรีย**/ bacteria** (Disease is caused by bacteria)
IF NP1^<ติดเชื้อ/infect ><NP2=*>^< Prep= จาก/from>^<NP3= contaction> then causality	เด็ก **/Kid** ติดเชื้อ / **infect** จาก **/from**การสัมผัส**/ contaction** (A kid is infected by contaction)

Named Entity Recognition in Bengali

Asif Ekbal
Department of Computer Science and Engineering
Jadavpur University,Kolkata,India
Email: ekbal_asif12@yahoo.co.in/asif.ekbal@gmail.com

Abstract

A tagged Bengali news corpus, developed from the web, has been used in this work for the recognition of named entities (NEs) in Bengali language. A supervised learning method has been adopted to develop two different models of a Named Entity Recognition (NER) system, one (Model A) without using any linguistic features and the other (Model B) by incorporating linguistic features. The different tags in the news corpus help to identify the seed data. The training corpus is initially tagged against the different seed data and a lexical contextual seed pattern is generated for each tag. The entire training corpus is shallow parsed to identify the occurrence of these initial seed patterns. In a position where the context or a part of each seed pattern matches, the systems predict the boundary of a named entity and further patterns are generated through bootstrapping. Patterns that occur in the entire training corpus above a certain threshold frequency are considered as the final set of patterns learnt from the training corpus. The test corpus is shallow parsed to identify the occurrence of these patterns and estimate the named entities. Models have been tested with two news documents (Gold Standard Test Sets) and their results have been compared in terms of evaluation parameters.

1. Introduction

Named Entity Recognition (NER) is an important tool in almost all Natural Language Processing (NLP) application areas. NER's main role is to identify expressions such as the names of people, locations and organizations as well as date, time and monetary expressions. Such expressions are hard to analyze using traditional NLP because they belong to the open class of expressions, i.e., there is an infinite variety and new expressions are constantly being invented.

The problem of correct identification of NEs is specifically addressed and benchmarked by the developers of Information Extraction Systems, such as the GATE system [1] and the multipurpose MUSE system [2]. Morphological and contextual clues for identifying NEs in English, Greek, Hindi, Romanian and Turkish have been reported in [3]. The shared task of CoNLL-2003 [4] were concerned with language independent NER. An unsupervised learning algorithm for automatic discovery of NEs in a resource free language has been presented in [5]. A framework to handle the NER task for long NEs with many labels has been described in [6]. For learning generalized names in text an algorithm, NOMEN, has been presented in [7]. NOMEN uses a novel form of bootstrapping to grow sets of textual instances and their contextual patterns. A joint inference model has been presented in [8] to improve Chinese name tagging by incorporating feedback from subsequent stages in an information extraction pipeline: name structure parsing, cross-document co-reference, semantic relation extraction and event extraction. It has been shown in [9] that a simple-two stage approach to handle non-local dependencies in NER can outperform existing approaches that handle non-local dependencies, while being much more computationally efficient. But in Indian Languages, no work in the area of has been carried out as yet.

The rest of the paper is organized as follows. Section 2 deals with the NER task in Bengali. Section 3 shows the evaluation techniques and results. Finally, conclusion is drawn in Section 4.

2. Named Entity Recognition in Bengali

Bengali is the fifth language in the world, second in India and the national language of Bangladesh. NER in Indian Languages (ILs) in general and in Bengali in particular is difficult and challenging. In English, NE always appears with capitalized letters but there is no concept of capitalization in Bengali. In the present work, a supervised learning system based on pattern directed shallow parsing has been used to identify named entities in Bengali using a tagged Bengali news corpus. The corpus has been developed from a widely used Bengali newspaper available in the web and at present it contains around 34 million wordforms. The location, reporter, agency and date tags in the tagged corpus help to identify the location, person, organization and miscellaneous names respectively and these serve as the seed data of the systems. In addition to these, most frequent NEs are collected from the different domains of the newspaper and used as the seed data. The systems have been trained on a

part of the developed corpus. The training corpus is partially tagged with elements from the seed list that serve as the gazetteer. The initial contextual lexical seed patterns that are learnt using the seed data and constitute a partial named entity grammar, identify the external evidences of NEs in the training corpus. These evidences are used to shallow parse the training corpus to estimate possible NEs that are manually checked. These NEs in turn help to identify further patterns. The training document is thus partially segmented into NEs and its context patterns. The context patterns that appear in the training document above a certain threshold frequency are retained and are expected to be applicable for the test documents as well in line with the maximum likelihood estimate. Initially the NER system has been developed using only the lexical contextual patterns learned (NER system without linguistic features i.e. Model A) from the training corpus and then linguistic features have been used along with the same set of lexical contextual patterns (NER system with linguistic features i.e. Model B) to develop it. The performance of the two systems has been compared using the three evaluation parameters namely Recall, Precision and F-Score.

2.1. Tagging with Seed Lists and Clue Words

The tagger places the left and right tags around each occurrence of the named entities of the seed lists in the corpus.

For example, <person> সোনিয়া গান্ধী (*sonia Gandhi*) </person>, <loc> কোলকাতা (*kolkata*) </loc> and <org> যাদবপুর বিশ্ববিদ্যালয় (*jadavpur viswavidyalya*) </org>.

For the Model A, the training corpus is tagged only with the help of different seed lists. In case of Model B, after tagging the entire training corpus with the named entities from the seed lists, the algorithm starts tagging with the help of different internal and external evidences that help to identify different NEs. It uses the clue words like surname (e.g.,মিত্র [*mitra*], দত্ত [*dutta]* etc.), middle name (e.g., চন্দ্র [*Chandra*], নাথ [*nath*] etc.), prefix word (e.g., শ্রীমান [*sriman*], শ্রী [*sree*], শ্রীমতী [*srimati*] etc.) and suffix word (e.g., বাবু [*-babu*], দা [*-da*], দি [*-di*] etc.) for person names. A list of common words (e.g., নেতা [*neta*], সাংসদ [*sangsad*], খেলোয়াড় [*kheloar*] etc.) has been kept that often determines the presence of person names. It considers the different affixes (e.g. - ল্যান্ড, [*-land*] - পুর [*-pur*], -লিয়া [*-lia*] etc.) that may occur with location names. The system also considers the several clue words that are helpful in detecting organization names (e.g., কোং. [*kong*], লিমিটেড [*limited*] etc.). Tagging algorithm also uses the list of words (e.g., কবিতা [*kabita*], কর [*kar*], ধর [*dhar*] etc.) that may appear as part of named entity as well as the common words. These clue words are kept in order to tag more and more NEs during the training of the system. As a result, more potential patterns are generated in the lexical pattern generation phase.

2.2. Lexical Seed Patterns Generation from the Training Corpus

For each tag T inserted in the training corpus, the algorithm generates a lexical pattern p using a context window of maximum width 4 (excluding the tagged NE) around the left and right tags, e.g., p = [l_{-2} l_{-1} <T> l_{+1} l_{+2}] where $l_{\pm i}$ are the context of p. Any of $l_{\pm i}$ may be a punctuation symbol. In such cases, the width of the lexical patterns will vary.

The lexical patterns are generalized by replacing the tagged elements by the tags. These generalized patterns form the set of potential seed patterns, denoted by P. These patterns are stored in a Seed Pattern table, which has four different fields namely: pattern id (identifies any particular pattern), pattern, type (Person name/ Location name/ Organization name/ Miscellaneous name) and frequency (indicates the number of times any particular pattern appears in the entire training corpus).

2.3. Generation of new Patterns through Bootstrapping

Every pattern p in the set P is matched against the entire training corpus. In a place, where the context of p matches, the system predicts where one boundary of a name in the text would occur. The system considers all possible *noun, verb and adjective inflections* during matching. At present, there are 214 different verb inflections and 27 noun inflections in the systems. During pattern checking, the maximum length of a named entity is considered to be six words. Each named entity so obtained in the training corpus is manually checked for correctness. The training corpus is further tagged with these newly acquired named entities to identify further lexical patterns. The bootstrapping is applied on the training corpus until no new patterns can be generated. The patterns are added to the pattern set P with the 'type' and 'frequency' fields set properly, if they are not already in the pattern set P with the same 'type'. Any particular pattern in the set of potential patterns P may occur many times but with different 'type' and with equal or different 'frequency' values. For each pattern of the set P, the probabilities of its occurrence as Person, Location, Organization and Miscellaneous names are calculated.

For the candidate patterns acquisition under each *type*, a particular threshold value of probability is chosen. All these acquired patterns form the set of *accepted* patterns and this set is denoted by *Accept Pattern*.

Any particular pattern may appear more than once with different *type* in the *Accept Pattern* set. So, while testing the NER systems, some identified NEs may be assigned more than one named entity categories (*type*). Model A cannot cope with this NE-classification disambiguation problem at present. Model B uses different linguistic features, as identified in Section 3.1 to deal with this NE-classification disambiguation problem.

3. Evaluation and Results

The set of accepted patterns is applied on a test set. The process of pattern matching can be considered as a shallow parsing process.

3.1. Training and Test Set

A supervised learning method has been followed to develop two different models of a NER system. The systems have been trained on a portion of the tagged Bengali news corpus. Some statitics of training corpus is as follows:

Total number of news documents = 1819, Total number of sentences in the corpus = 44432, Average number of sentences in a document = 25, Total number of wordforms in the corpus = 541171, Average number of wordforms in a document = 298, Total number of distinct wordforms in the corpus = 45626.

This training set is initially tagged against the different seed lists used in the system and the lexical pattern generation, pattern matching, new pattern generation and candidate pattern acquisition procedures are sequentially performed.

Two manually tagged test sets (Gold Test Sets) have been used to evaluate the models of the NER system. Each test corpus has been collected from a particular news topic (i.e. international, national or business).

3.2. Evaluation Parameters

The models have been evaluated in terms of Recall, Precision and F-Score. The three evaluation parameters are defined as follows:

Recall (R) = (No. of tagged NEs) / (Total no. of NEs present in the corpus) * 100%

Precision (P) = (No. of correctly tagged NEs) / (No. of tagged NEs) * 100%

F-Score (FS) = (2 * Recall* Precision) / (Recall + Precision) *100%

3.3. Evaluation Method

The actual number of different types of NEs present in each test corpus (Gold Test Set) is known in advance and they are noted. A test corpus may be used in generating new patterns also i.e. it may be utilized in training the models after evaluating the models with it. The test sets have been ordered in order to make them available for inclusion in the training set. The two different test sets can be ordered in 2 different ways. Out of these 2 different combinations, a particular combination has been considered in the present work. It may be interesting to consider the other combination and observe whether the results vary.

Each pattern of the *Accept Pattern* set is matched against the first Test corpus (Test Set1) according to the pattern matching process described in Section 2 and the identified NEs are stored in the appropriate NE category tables. Any particular pattern of the *Accept Pattern* set may assign more than one NE categories to any identified NE of the test set. This is known as the NE-Classification disambiguation problem. The identified NEs, assigned more than one NE categories, should be further verified for the correct classification. Model A cannot cope with this situation and always assigns the highest probable NE category to the identified NE. One the other hand different linguistic patterns, used as the clue words for the identification of different types of NEs (in Section 2), are used in order to assign the actual categories (NE type) to the identified NEs in Model B. Once the actual category of a particular NE is explored, it is removed from the other NE category tables.

The same procedures described in Section 2 are performed for this test set (Test Set1) in order to include it in to the training set. Now, the resultant *Accept Pattern* is formed by taking the union of the initial *Accept Pattern* set and the *Accept Pattern* set of this test corpus. This resultant *Accept Pattern* set is used in evaluating the NER models with the next test corpus (Test Set 2 in the order).. So in each run, some new patterns may be added to the set of *Accept Pattern*. As a result, the performance of the NER systems (models) gradually improves since all the test sets have been collected from a particular news topic. The Bengali date, Day and English date can be recognized from the different date tags in the corpus. Some person names, location names and organization names can be identified from the reporter, location and agency tags in the corpus.

The three evaluation parameters are computed for each individual NE category i.e. for person name, location name, organization name and miscellaneous name.

3.4. Results and Discussions

The performance of the systems with the help of two news documents (test sets), collected from a particular news topic, has been presented in Tables 1 and 2. Here, following abbreviations have been used: Person name (PN), Location name (LOC), Organization name (ORG) and Miscellaneous (MISC).

	NE category	R	P	FS
Model B	PN	71.6	79.2	**75.30**
	LOC	66.2	74.2	**69.71**
	ORG	64.7	73.4	**68.8**
	MISC	33.2	98.3	**49.63**
Model A	PN	69.1	73.2	71.09
	LOC	64.7	67.1	65.87
	ORG	62.3	63.7	62.68
	MISC	33.2	98.3	49.63

Table 1: Result for Test Set 1

	NE category	R	P	FS
Model B	PN	72.8	81.2	**76.80**
	LOC	67.9	76.5	**71.96**
	ORG	66.3	75.1	**70.40**
	MISC	37.2	99.1	**54.09**
Model A	PN	69.9	73.8	71.8
	LOC	65.3	68.1	66.67
	ORG	63.6	64.1	63.84
	MISC	37.2	99.1	54.09

Table 2: Result for Test Set 2

It is observed from Tables 1 and 2 that the *NER system with linguistic features* i.e. Model B outperforms the *NER system without linguistic features* i.e. Model A in terms of Recall, Precision and F-Score. Linguistic knowledge plays the key role to enhance the performance of Model B compared to Model A. Improvement in Precision, Recall and F-Score values with test set 2 occurs as test set 1 is included in this case as part of the training corpus. Whenever any pattern of the set of accepted patterns (*Accept Pattern)* produces more than one NE categories *(type*) for any identified (from the test corpus) NE, Model A always assigns that particular NE category (type) which has the maximum probability value for that pattern. This often produces some errors in assigning NE categories to the identified NEs. So the precision values diminish and as a result the F-Score values get affected. Model B solves this problem with the help of linguistic knowledge and so its precision as well as the F-Score values are better than Model A. At present, the systems can only identify the various date expressions but cannot identify the other miscellaneous NEs like monetary expressions and time expressions.

4. Conclusion and Future Works

Experimental results show that the performance of a NER system employing machine learning technique can be improved significantly by incorporating linguistic features. The lexical Patterns can further be generalized by replacing each of $l_{\pm l}$ of p with its lexical category (i.e. each word is replaced by it's part of speech information). So, a Part-of-Speech (POS) tagger could be more effective in generating more and more potential general patterns. In presence of the POS tagger, the potential NEs can be identified by the *regular expression* of the form Noun^+ (i.e. NE is always a sequence of noun words).

Currently, we are working to include the HMM based part of speech tagger and a rule based chunker in to the systems. More linguistic knowledge could be helpful in NE-classification disambiguation problem and as a result precision values of different NE categories would increase. Observation of the results with the various orders of the test sets would be an interesting experiment.

References

[1] H. Cunningham, Gate, a general architecture for text engineering, *Computing and the Humanities*, 2001.

[2] D. Maynard, V. Tablan, K. Cunningham, and Y. Wilks, Muse: a multisource entity recognition system, *Computing and the Humanities*, 2003.

[3] S. Cucerzon and David Yarowsky, Language independent named entity recognition combining morphological and contextual evidence, *Proceedings of the 1999 Joint SIGDAT conference on EMNLP and VLC*, 1999.

[4] F. Erik, Tjong Kim Sang and Fien De Meulder, Introduction to the CoNLL-2003 Shared Task: Language Independent Named Entity Recognition, *Proceedings of the CoNLL-2003*, Edmonton, Canada, 2003, pp.142-147.

[5] A. Klementiev and D. Roth, Weakly Supervised Named Entity Transliteration and Discovery from Multilingual Comparable Corpora, In *Proceedings of the COLING-ACL 2006*, Sydney, Australia, 17-21 July, pp. 817-824.

[6] D. Okanohara, Y. Miyao, Y. Tsuruoka and J. Tsujii, Improving the Scalibility of Semi-Markov Conditional Random Fields for Named Entity Recognition, In *Proceedings of the COLING-ACL 2006*, Sydney, Australia, 17-21 July, pp.465-472.

[7] R. Yangarber, W. Lin and R. Grishman, Unsupervised Learning of Generalized Names, *In Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002).*

[8] Heng Ji and Ralph Krishnan, Analysis and Repair of Name Tagging Errors, In *Proceedings of the COLING-ACL 2006*, Sydney, Australia, 17-21 July, pp.420-427.

[9] Vijay Krishnan and Christopher D. Manning, An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition, In *Proceedings of the COLING-ACL 2006*, Sydney, Australia, 17-21 July, pp.1121-1128.

Fast Prototyping of a Malay WordNet System

LIM Lian Tze and Nur HUSSEIN
Computer Aided Translation Unit
School of Computer Sciences
Universiti Sains Malaysia
Penang, Malaysia
{liantze,hussein}@cs.usm.my

ABSTRACT

This paper outlines an approach to produce a prototype WordNet system for Malay semi-automatically, by using bilingual dictionary data and resources provided by the original English WordNet system. Senses from an English-Malay bilingual dictionary were first aligned to English WordNet senses, and a set of Malay synsets were then derived. Semantic relations between the English WordNet synsets were extracted and re-applied to the Malay synsets, using the aligned synsets as a guide. A small Malay WordNet prototype with 12429 noun synsets and 5805 verb synsets was thus produced. This prototype is a first step towards building a full-fledged Malay WordNet.

KEYWORDS

WordNet, Malay, lexical knowledge base, fast prototyping

1 INTRODUCTION

Traditional dictionaries compile lexical information about word meanings by listing them alphabetically by the headwords. While this arrangement is convenient for a human reader who wants to look up the meaning of a word, it does not provide much information about explicit semantic relations between words, besides the usual synonyms and antonyms.

WordNet [6, 8] is a lexical database system for English words, designed based on psycholinguistic principles. It organises word meanings (senses) on a semantic basis, rather than by the surface morphological forms of the words. This is done by grouping synonyms into sets, and then defining various relations between the synonym sets (synsets). Some examples of the semantic relations defined include hypernymy (the `is-a` relation) and meronymy (the `part-of` relation).

Armed with such semantic relations, WordNet became an invaluable resource for natural language processing (NLP) researchers in tackling problems like information retrieval, word sense disambiguation, and question answering. As the original WordNet contains only English words, there have been efforts to create WordNet-like systems for other languages. See the Global WordNet Association's website [4] for a list of such projects.

Currently, no WordNet-like lexical database system exist for the Malay language. Such a resource will be useful indeed for NLP research involving Malay texts. While the construction of a complete WordNet-like system is a daunting undertaking which requires lexicographic expertise, it is possible to build a prototype system semi-automatically using resources accessible at our site. The prototype Malay WordNet system and data can then be further scrutinised, fine-tuned and improved by human lexicographers.

The main aim of developing this prototype was to explore the design and tools available in a WordNet system, rather than a full attempt to develop high quality Malay WordNet data. Therefore, the methods we adopted are not as extensive as other efforts in constructing non-English WordNets, such as the work reported in [1, 2].

2 METHODOLOGY

We describe how a prototype Malay WordNet can be constructed semi-automatically using a English-Malay bilingual dictionary, the original English WordNet, and alignments between the two resources.

The developers of the English WordNet, the Cognitive Science Laboratory at Princeton University, have made available some useful tools that allow the custom development of WordNet-like systems [7]. They include:

- English WordNet database files,
- WordNet Browser, a GUI front-end for searching and viewing WordNet data,
- WordNet database search functions (as C library functions),
- `GRIND`, a utility tool for converting lexicographer input files into WordNet database files.

If lexicographer input files for Malay words can be created following the required syntax, `GRIND` can be used to process them to produce Malay WordNet database files, to be viewed using the WordNet browser. This can be done

by first establishing a set of Malay word synsets and the semantic relations between them, and then generating the lexicographer files.

2.1 Malay Synsets

Kamus Inggeris Melayu Dewan (KIMD) [5] is an English-Malay bilingual dictionary and provides Malay equivalent words or phrases for each English word sense. Linguists at our research group had previously aligned word senses from KIMD and WordNet 1.6. Not all KIMD and WordNet 1.6 senses were included; only the more common ones were processed.

Here are some example alignments for some senses of *dot*, *consolidation* and *integration*:

Listing 1: Aligned senses of *dot*

```
kimd (dot, n, 1, 0, [small round spot, small circular shape], <titik,
     bintik> ).
wordnet (110025218, 'dot', n, 1, 0, [a very small circular shape] ).
```

Listing 2: Aligned senses of *consolidation*

```
kimd (consolidation, n, 1, 0, [act of combining, amalgamating], <
     penggabungan, penyatuan>).
wordnet (105491124, 'consolidation', n, 1, 0, [combining into a
     solid mass]).
wordnet (100803600, 'consolidation', n, 2, 0, [the act of combining
     into an integral whole]).
```

Listing 3: Aligned senses of *integration*

```
kimd (integration, n, 1, c, [act of c. (combining into a whole)], <
     penyepaduan, pengintegrasian>).
wordnet (100803600, 2, 'integration', n, 2, 0, [the act of combining
     into an integral whole]).
```

(The 9-digit number in each English WordNet sense above is a unique identifier to the synset it belongs to.)

A set of Malay synsets may be approximated based on the KIMD–WordNet alignment using Algorithm 1.

Algorithm 1 Constructing Malay synsets

```
for all English synset es do
  ms-equivs ⇐ empty //list of Malay equivalent words
  ms ⇐ null //Equivalent Malay synset
  for all s ∈ {KIMD senses aligned to es} do
    add Malay equivalent(s) of s to ms-equivs
  end for
  ms ⇐ new synset containing ms-equivs
  Set ms to be equivalent Malay synset to es
end for
```

Following this algorithm, the following Malay synsets are derived from the sense alignments in Listings 1–3. The corresponding English WordNet synsets are also shown:

- (*titik, bintik*)
  (110025218: point, dot; [a very small circular shape])
- (*penggabungan, penyatuan*)
  (105491124: consolidation; [combining into a solid mass])
- (*penggabungan, penyatuan, penyepaduan, pengintegrasian*)
  (100803600: consolidation, integration; [the act of combining into an integral whole])

2.2 Synset Relations

For this fast prototyping exercise, we have decided to create semantic relations between the Malay synsets based on the existing relations between their English equivalents. Algorighm 2 shows how this can be done.

Algorithm 2 Creating relations between Malay synsets

```
Require: lookup_ms(es):
  returns Malay synset equivalent to English synset es
Require: lookup_es(ms):
  returns English synset equivalent to Malay synset ms
Require: get_target(R, es):
  returns target (English) synset of English synset es for
  relation R
  for all Malay synset ms do
    es ⇐ lookup_es(ms)
    for all relation R with a pointer from es do
      ms′ ⇐ null
      es′ ⇐ es
      if R is transitive then
        repeat
          es′ ⇐ get_target(R, es)
          ms′ ⇐ lookup_ms(es′)
        until es′ = null or ms′ ≠ null
      else
        es′ ⇐ get_target(R, es)
        ms′ ⇐ lookup_ms(es′)
      end if
      if ms′ ≠ null then
        add (R, ms′) to list of relations that applies to ms.
      end if
    end for
  end for
```

As an example, the hypernymy relation holds between the English synsets (*point, dot*) and (*disk, disc, saucer*). Therefore, a hypernymy relation is established between the corresponding Malay synsets (*bintik, titik*) and (*ceper, piring*).

However, while searching for target synsets for a relation R, it is always possible that there is no Malay equivalent for an English synset. If R is transitive, as are hypernymy and meronymy, we continue to search for the next target synset in the transitive relation chain, until we reach the last English synset in the chain.

To illustrate, consider the English and Malay synsets in Figure 1. The English synset (*disk, disc, saucer*) has the hypernym (*round shape*), which in turn has the hypernym (*shape, form*). While (*round shape*) does not have a corresponding Malay synset in our data, (*shape, form*) does have one as (*bentuk, corak*). Therefore, a hypernymy relation is established between (*ceper, piring*) and (*bentuk, corak*).

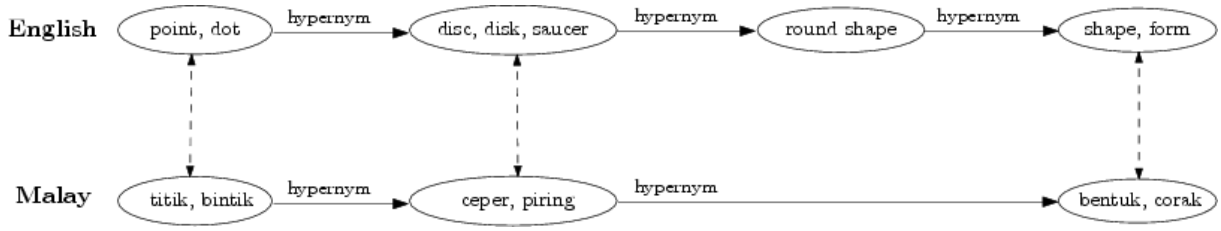


Figure 1: English and Malay synsets forming a hypernymy chain

2.3 Lexicographer Files

WordNet systems organise synsets of different syntactic categories, i.e. nouns, verbs, adjectives and adverbs, separately. In addition, the English WordNet also assign semantic fields to the synsets, such as `noun.location`, `noun.animal` and `verb.emotion`. Synsets of different categories are to be stored in separate lexicographer files, the names of which correspond to their semantic fields.

For each Malay synset identified in section 2.2, we look up f, the semantic field of its equivalent English synset. The Malay synset, together with its relations and target synsets, is then appended to the lexicographer file f.

3 IMPLEMENTATION

The procedures described in sections 2.2 and 2.3 were implemented as a suite of tools called `LEXGEN` in C and Java. As a first step, only noun and verb synsets were processed with `LEXGEN`. Since KIMD does not provide Malay glosses, `LEXGEN` reuses glosses from English WordNet. The resulting lexicographer files were then put through `GRIND`, producing a small Malay WordNet system.

4 RESULTS

The prototype Malay WordNet system currently contains 12429 noun synsets and 5805 verb synsets. Its small coverage of the English WordNet (81426 noun synsets and 13650 verb synsets) is understandable as only a subset of KIMD and WordNet senses was used in the earlier alignment work. The prototype also includes the hypernymy, hyponymy, troponymy, meronymy, holonymy, entailment and causation relations. Figure 4 shows the Malay synset (*bintik, titik*) and its hypernyms as viewed in the WordNet Browser.

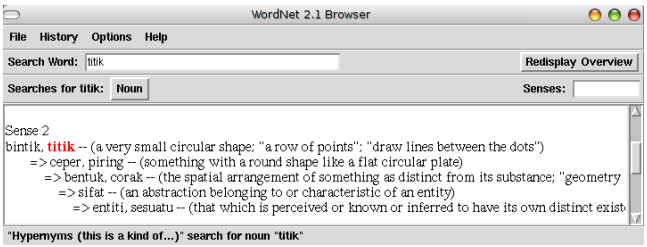


Figure 2: Malay WordNet as viewed in Browser

5 DISCUSSION

The Malay WordNet prototype is adequate for demonstrating what a WordNet system has to offer for Malay. This is especially helpful to give a quick preview to users who are not yet familiar with the WordNet or lexical sense organisation paradigm. However, as acknowledged at the very beginning, its current quality is far from satisfactory.

Part of the problem is in the dictionary used. The KIMD–WordNet alignment work was part of a project to collect glosses for English word senses from different dictionaries. As such, the suitability of Malay equivalents to be lemmas were not the main concern: all Malay equivalents were simply retained in the alignment files.

This leads to unsuitable Malay WordNet synset members in some cases: since KIMD is a unidirectional English to Malay dictionary, not all Malay equivalents it provides can stand as valid lemmas. For example, KIMD provides *orang, anggota, dan lain-lain yang tidak hadir* (literally 'person, member, etc. who are not present') as the Malay equivalent for English *absentee*. While this is valid as a Malay gloss or description for the synset, it is unsuitable to be a member lemma of a synset.

In addition, we also lack Malay gloss information for the Malay synsets as these were not provided in KIMD. The prototype Malay WordNet, therefore, is forced to have English text as glosses, intead of Malay glosses.

We also noted that the English WordNet provide verb frames, e.g. Somebody —s something for a sense of the verb *run*. The first problem is that we have yet to establish a list of verb frames for Malay. Secondly, even if there were, there is not necessarily a one-to-one mapping between the English and Malay verb frames. Thirdly, as the English verb frames are hard-coded into `GRIND` and WordNet, extensive re-programming would be required to use these utilities on different languages. Therefore, we have not attempted to handle Malay verb frames for this prototype.

`GRIND` imposes a maximum of sixteen senses per word form in each lexicographer file. This might be a problem if there are Malay words that are very polysemous. Possible alternatives are:

- further split the senses into different lexicographer files so that each file would not contain more than sixteen senses of the same word,
- aim for coarser sense distinctions, or
- re-program `GRIND`.

Finally, the derivation of Malay synsets from the KIMD–WordNet alignments may be flawed. This is because multi-

ple KIMD senses may be aligned to a WordNet sense, and vice versa. Referring back to Listing 2 and the list of Malay synsets at the end of Section 2.1, we see that the Malay words *penggabungan* and *penyatuan* from *one* KIMD sense now appear in *two* synsets. To non-lexicographers, such as the authors of this paper, it is unclear how this situation should be handled. Are there now two senses of *penyatuan* and *penggabungan*, or should the Malay synsets (*penggabungan, penyatuan*) and (*penggabungan, penyatuan, penyepaduan, pengintegrasian*) be merged? Since there are opinions that the English WordNet is too fine-grained, the synsets can perhaps be merged to avoid the problem for Malay WordNet. Nevertheless, we think a lexicographer would be more qualified to make a decision.

6 FUTURE WORK

The aim of work on the prototype Malay WordNet is but to explore the architecture and software tools required in a WordNet system. Future work will focus more on systematically compiling lexical data for a Malay WordNet system by lexicographers and linguistic experts. We highlight some issues of interest here.

- A Malay monolingual lexicon or dictionary should be used to determine the Malay synsets, the gloss text for each synset, as well as the synset's semantic field.
- The semantic fields are hard-coded into `GRIND` and WordNet. Therefore, if we are to have localised semantic fields in Malay, e.g. `noun.orang` (`noun.person`) and `noun.haiwan` (`noun.animal`), or to add new fields, `GRIND` and WordNet will need to be modified.
- Semantic relations need to be defined between the Malay synsets. This may be aided by machine learning strategies, such as those used in [1], besides human efforts.
- A list of Malay verb frames need to be drawn up and assigned to each verb sense.
- Currently, the Malay word senses are ordered at random. Ideally, the senses should be numbered to reflect their usage frequency in natural texts. A sense-tagged Malay corpus will help in this, as was done in the English WordNet [7, p.112].
- It would also be interesting to align the Malay WordNet to EuroWordNet [3], which contains wordnets for several European languages. As EuroWordNet is aligned to English WordNet 1.5, some re-mapping would have to be performed if we wish to re-use the KIMD–WordNet alignment, or the prototype, as a rough guide.

7 CONCLUSION

Creating a new set of Wordnet lexicographer files from scratch for a target language is a daunting task. A lot of work needs to be done in compiling the lexicographer input files and identifying relations between synsets in the language. However, we have been successful in rapidly constructing a prototype Malay Wordnet by bootstrapping the synset relations off the English Wordnet. Hopefully, this will lay the foundation for the creation of a more complete Malay Wordnet system.

REFERENCES

[1] J. Atserias, S. Climent, X. Farreres, G. Rigau, and H. Rodríguez. Combining multiple methods for the automatic construction of multilingual wordnets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, Tzigov Chark, Bulgaria, 1997.

[2] I. Azarova, O. Mitrofanova, A. Sinopalnikova, M. Yavorskaya, and I. Oparin. RussNet: Building a lexical database for the Russian language. In *Proceedings of Workshop on WordNet Structures and Standardisation and How this affect Wordnet Applications and Evaluation*, pages 60–64, 2002.

[3] EuroWordNet. Eurowordnet: Building a multilingual database with wordnets for several European languages, 2006. URL `http://www.illc.uva.nl/EuroWordNet/`. Last accessed September 15, 2006.

[4] Global WordNet Assoc. Wordnets in the world, 2006. URL `http://www.globalwordnet.org/gwa/wordnet_table.htm`. Last accessed September 15, 2006.

[5] A. H. Johns, editor. *Kamus Inggeris Melayu Dewan*. Dewan Bahasa dan Pustaka, Kuala Lumpur, Malaysia, 2000.

[6] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235–312, 1990.

[7] R. I. Tengi. Design and implementation of the wordnet lexical database and searching software. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 4, pages 105–127. MIT Press, Cambridge, Massachusetts, 1998.

[8] WordNet. WordNet: a lexical database for the English language, 2006. URL `http://wordnet.princeton.edu/`. Last accessed September 15, 2006.

Taxonomic Ontology Learning by using Item List on the Basis of Text Corpora in Thai

Aurawan Imsombut Asanee Kawtrakul

The Specialty Research Unit of Natural Language Processing
and Intelligent Information System Technology
Department of Computer Engineering,
Kasetsart University, Bangkok, Thailand
{g4685041,ak}@ ku.ac.th

Abstract

Ontologies are an essential resource to enhance the performance of an information processing system as they enable the re-use and sharing of knowledge in a formal, homogeneous and unambiguous way. We propose here a method for the automatic learning of an taxonomic ontology based on Thai text corpora. To build the ontology we extract terms and relations. For the former we use a shallow parser, while for the extraction of taxonomic relations we use item lists, i.e. bullet lists and numbered lists. To deal with this, we need to identify first which lists contain an ontological term and to solve the problems of embedding of lists and the boundaries of the list. Then, we extract the hypernym term of the item lists by using the lexicon and the contextual features of the candidate term of the hypernym. The accuracy of the ontology extraction from item list is 0.72.

1 Introduction

Ontology is a well-known term in the field of AI and knowledge engineering. Numerous definitions have been offered, and a common acceptance of the term is to consider it as "an explicit specification of a conceptualization." [4]. We define ontology here as "a general principle of any system to represent knowledge for a given domain, with information from multiple, heterogeneous sources. Information can be represented by concepts and semantic relationships between them." An ontology can be used for many purposes. It can be used in Natural Language Processing to enhance the performance of machine translation, text summarization, information extraction and document clustering.

The building of an ontology by an expert is an expensive task. It is also a never ending process because knowledge increases all the time in real world, especially in the area of science. Hence we suggest to build ontologies automatically.

Texts are a valuable resource for extracting ontologies as they contain a lot of information concerning the concepts and their relationships. We can classify the expression of ontological relationships in texts into explicit and implicit cues. In the presence of an explicit cue, an ontological element can be detected by using the cue i.e. lexico-syntactic patterns [7] and an item list (bullet list and numbered list). Implicit cues do not have any concrete word to hint at the relationship [6]. In this work, we focus on extracting hypernym and hyponym (or taxonomic) relations because they are the most important relation in ontology and they are also skeleton of the knowledge. To deal with this we use item list for hinting taxonomic relation. We propose a method for detecting ontological item lists and for extracting the hypernym class of list items. The system selects the appropriate hypernym term from a list of candidates, choosing the most likely one (the one with the highest probability) according to the lexicon and some contextual features. We tested the system by using Thai corpora in the domain of agriculture.

The remainder of this paper is organized as follows. Section 2 presents the related works of ontology extraction from unstructured text. Section 3 describes crucial problems for extraction of an ontology by using item list. In section 4, we propose methodology for automatically extracting hypernym relation of an ontology on the basis of corpora. The experimental results and conclusions are shown in section 5 and 6, respectively.

2 Related Works

There are a number of proposals to build ontologies from unstructured text. The first one to propose the extraction of semantic relations by using lexico-syntactic patterns in the form of regular expressions was Hearst [5]. Secondly, statistical techniques have often been used for the same task. Indeed, many researches [1], [2], [8], [10] have used clustering techniques to extract ontologies from corpora by using different features for different clustering algorithms. This approach allows to process a huge set of data and a lot of features, but it needs an expert to label each cluster node and each relationship name. Another approach is hybrid. Maedche and Staab [9] proposed an algorithm based on statistical techniques and association rules of data mining technology to detect relevant relationships between ontological concepts. Shinzato and Torisawa [12]

presented an automatic method for acquiring hyponymy relations from itemization and listing of HTML documents. They used statistical measures and some heuristic rules. In this paper, we suggest to extract the ontology from itemized lists in plain text, i.e. without any HTML markup like most of the previous work, to select the appropriate hypernym of the list items.

3 Crucial Problems for the Extraction of Hypernym relation by using item list

When using item lists as cues to signal a hypernym relation, we need to identify first which lists contain an ontological term and whether the lists are coherent. Since the input of our system is plain text, we do not have any mark up symbols to show the position and the boundaries of the list. This is why we used bullet symbols and numbers to indicate the list, which is not without posing certain problems.

Embedding of lists. Some lists may have long descriptions and some of them can contain another list. This causes a problem of identification. We solve this problem by detecting each list following the same bullet symbol or order of numbering. Despite that, there are cases where an embeded list may have a continuous number. In this case, we assume that different lists talk about different topics; hence we need to identify the meaning of each one of them.

Long boundaries of description in each list item. Since some lists may have long descriptions, it is difficult to decide whether the focused item is meant to continue the previous list or start a new list.

Non-ontological list item. Quite so often authors express procedures and descriptions in list form. But the procedure list items are not the domain's ontological terms, and some description list items may not be ontology terms at all, hence the system needs to detect the ontology term list.
Figure 1 illustrates the problem of item list identification.

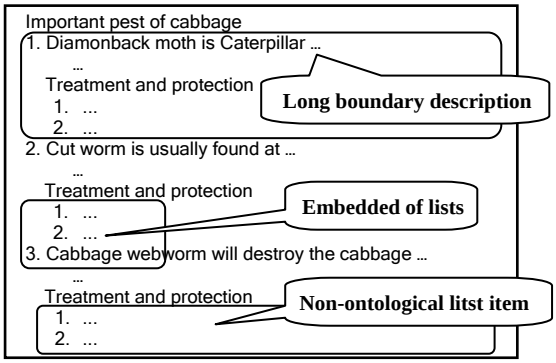


Fig. 1. Example of problems of item list identification.

4 Methodology for automatically extracting hypernym relation of an ontology

The proposed methods in this paper for taxonomic-ontology extraction from itemized lists is dealt with by a hybrid approach: natural language processing, rule based and statistical based techniques. We decompose the task into 3 steps that are Morphological Analysis and Noun phrase chunking, Item list identification and Extraction of hypernym Term of list items.

Morphological analysis and noun phrase chunking.

Just as in many other Asian languages, there are no delimiters (blank space) in Thai to signal word boundaries. Texts are a single stream of words. Hence, word segmentation and part-of-speech (POS) tagging [13] are necessary for identifying a term unit and its syntactic category. Once this is done documents are chunked into phrases [11] to identify shallow noun phrase boundaries within a sentence. In this paper, the parser relies on NP rules, word formation rules and lexical data. In Thai, NPs are sometimes sentence-like patterns; this is why it is not always easy to identify the boundary of NPs composed of several words including a verb. The candidate NP might then, be signaled by another occurrence of the same sequence in the same document. The to-be-selected sentence-like NPs should be those occurring more than one time.

Item list identification

Since an author can use item lists to describe objects, procedures and the like, this might lead to non-ontological lists. Hence we will use here only object lists, because in doing so we can be sure that it contains an ontological term. In consequence, the system will classify list types by considering only item lists that have to fulfill the constraints:

- item lists whose items are marked as the same Name Entity (NE) class. [3]
- item lists are composed of more than one item.

This works well here, because in the domain of agriculture named entities are not only names at the leave level. For example, rice is considered as plant named entity as well as varieties of rice and rice does not occur at the leave level of plant ontology. This being so, our method can efficiently identify ontological terms. Moreover, this phenomenon is very much alike in other domains such as bio-informatics. In our study we classified lists into bullet lists and numbering lists. We also considered that a bullet lists must contain the same bullet symbol and the same NE class. The system considers the same numbering list by ordering number and making sure the item belong to the same NE class.

This technique can solve the problem of embedded lists. Since we assumed that different lists talk about different topics and different NE classes, this method can distinguish between different item lists. Moreover, it works with the list item that has a long boundary as paragraph.

Extraction of hypernym Term of list items

Having identified the item list by considering bullets or numbering, the system will discover the hypernym term from a set of candidates by using lexical and contextual features. In order to generate a list of candidate hypernym terms, the system considers only the terms that occur in the preceeding paragraphs of the item list. The one closest to the first item term of the list will be considered first. Next, the most likely hypernym value (*MLH*) of term in the candidates list will be computed on the basis of an estimated function taking lexical and contextual features into account. Let $h \in H$, H is the set of candidates of possible hypernym terms while $t \in T$, T is the set of term in the item list. The estimate function for computing the most likely hypernymy term is defined as follows:

$$MLH(h,t) = \alpha_1 \cdot f_1 + \alpha_2 \cdot f_2 + ... + \alpha_n \cdot f_n$$

Where α_k is the weight of feature *k*, f_k is the feature *k* and *n* is total number of features (=5). f_1-f_4 are lexical features and f_5 is contextual feature. Each feature (f_k) is weighted by α_k and in our experiment we set all weight with the same value (=1/*n*). The system will select the candidate term that has the maximum *MLH* value to be the hypernym of the item list terms.

Lexical features. They have binary value. The features are:

f_1: Head word compatible. This feature consider that head word of candidate term is compatible with list item term or not.

$$f_1(h,t) = \begin{cases} 1; h \text{ is compatible with the head word term of } t \\ 0; \text{otherwise} \end{cases}$$

f_2: Same NE class. This feature consider that candidate term of a hypernym belong to the same NE class as list item term or not.

$$f_2(h,t) = \begin{cases} 1; h \text{ belong to the same NE class as } t \\ 0; \text{otherwise} \end{cases}$$

f_3: Different NE class. This feature consider that candidate term of a hypernym belong to the different NE class as list item term or not.

$$f_3(h,t) = \begin{cases} 1; h \text{ belong to the different NE class as } t \\ 0; \text{otherwise} \end{cases}$$

Comment: concerning NEs we distinguish between two features (f_2 and f_3), since candidate terms of hypernym can have or not have NE class. The case that candidate term do not have NE class can possible, especially when they occur at a high level of the taxonomy, e.g. /phuet trakun thua/(pulse crops). Then, when we compare the NE class of two terms it can be three possible values, that are 'same NE class', 'different NE class' and 'can not defined' (this occurs if candidate term do not have NE class). Hence, we use these two features for representing all these possible values.

f_4: Topic term. This feature consider that candidate term is the topic term of the document (short document) or a topic term of the paragraph (long document) or not. Here, topic term will be computed by using tf/idf.

$$f_4(c,h) = \begin{cases} 1; h \text{ is a topic term of the document (short document) or a topic term of the paragraph (long document)} \\ 0; \text{otherwise} \end{cases}$$

Contextual feature. This feature has a value between 0 and 1. It is similarity value of word co-occurrence vector of each candidate term and item list term. Each feature in word co-occurrence vector corresponds to a context of the sentence in which the word occurs. We select the 500 most frequent terms (*l*) in the domain of agriculture as word co-occurrence feature and represent each candidate term of a hypernym (*h*) and each list item term (*t*) with this set of term feature. Each value of this vector is the frequency of co-occurrence of word co-occurrence feature term (*l*) and considering term (*h or t*). We compute the similarity between words *h* and *t* by using the cosine coefficient [19].

$$f_5(h,t) = \cos(\vec{h}, \vec{t}) = \frac{\sum_l h_l, t_l}{\left|\vec{h}\right|\left|\vec{t}\right|} = \frac{\sum_l h_l, t_l}{\sqrt{\sum_l h_l^2}\sqrt{\sum_l t_l^2}}$$

5 Experimental Results

The evaluation of our system was based on test cases in the domain of agriculture. The measurement of the system's performance is computed with the precision and recall by comparing the outputs of the system with the results produced by the experts. *Precision* gives the number of extracted correct results divided by the number of total extracted results, while *recall* shows the number of extracted correct results divided by the number of total corrects.

From a corpus about 15,000 words, the system can extract 284 concepts and 232 relations. The accuracy of hypernyms obtained by using different features is shown in Table 1. The result indicates that contextual features are more effective than lexical feature and lexical features yield a lower value of

recall than another since some hypernym terms do not share certain lexical features such as NE class. The precision of the system using both features is 0.72 and recall is 0.71. The errors are caused by some item lists are composed of two classes, for example, disease and pest. This is why the system can not detect this item list.

Table 1. The evaluation results of ontology extraction from item list

	precision	recall
Only lexicon feature	0.47	0.46
Only contextual feature	0.56	0.55
Both lexicon and contextual feature	**0.72**	**0.71**

6 Conclusion

In this article we presented and evaluated the hybrid methodologies, i.e., rule-based and learning, for the automatic building of ontology that is composed of term extraction and hypernym relation extraction. A shallow parser is used for terms extraction and item list (bullet lists and numbered list) are used for hypernym relation extraction. We extract the hypernym term of the item lists by using the lexicon and the contextual features of the candidate term of the hypernym.

We consider our results to be quite good, given that the experiment is preliminary, but the vital limitation of our approach is that it works well only for documents that contain a lot of cue-words. Based on our error analysis the performance of the system can be improved and the methodologies can be extended to other sets of semantic relations.

Acknowledgments. The work described in this paper has been supported by the grant of NECTEC No. NT-B-22-14-12-46-06. It was also funded in part by the KURDI; Kasetsart University Research and Development Institute.

References

1. Agirre, E., Ansa, O., Hovy, E., Martinez, D.: Enriching very large ontologies using the WWW. In Proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00) (2000)
2. Bisson, G., Nedellec, C., Cañamero, D.: Designing Clustering Methods for Ontology Building-The Mo'K Workbench. In Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence, ECAI'00, Berlin, Germany (2000)
3. Chanlekha, H., Kawtrakul, A.: Thai Named Entity Extraction by incorporating Maximum Entropy Model with Simple Heuristic Information. In Proceedings of the IJCNLP' 2004, Hainan Island, China (2004)
4. Gruber, T. R. A Translation Approach to Portable Ontology Specifications. (1993)
5. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics (1992)
6. Imsombut, A. and Kawtrakul, A.. Semi-Automatic Semantic Relations Extraction from Thai Noun Phrases for Ontology Learning. The Sixth Symposium on Natural Language Processing 2005 (SNLP 2005), Chiang Rai, Thailand. (2005)
7. Kawtrakul, A., Suktarachan, A., Imsombut A.: Automatic Thai Ontology Construction and Maintenance System. Workshop on OntoLex LREC conference, Lisbon, Portugal (2004)
8. Lin, D., Pantel, P.: Concept Discovery from Text. In Proceedings of the International Conference on Computational Linguistics. Taipei, Taiwan (2002) 577-583
9. Maedche, A., Staab, S.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems, vol. 16, no. 2. (2001)
10. Nedellec, C.: Corpus-based learning of semantic relations by the ILP system, ASIUM. In Learning language in Logic, Lecture Notes in Computer Science, vol. 1925, Springer-Verlag, June (2000) 259-278
11. Pengphon, N., Kawtrakul, A., Suktarachan, M.: Word Formation Approach to Noun Phrase Analysis for Thai. In Proceedings of SNLP2002, Thailand (2002)
12. Shinzato, K., Torisawa, K.: Acquiring Hyponymy Relations from Web Documents. In Proceedings of HLT-NAACL04, Boston, U.S.A., May (2004)
13. Sudprasert, S., Kawtrakul, A.: Thai Word Segmentation Based on Global and Local Unsupervised Learning. In Proceedings of NCSEC2003, Chonburi, Thailand (2003)

Discovery of Meaning from Text

Ong Siou Chin Narayanan Kulathuramaiyer Alvin W. Yeo
Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak,
Kota Samarahan, Malaysia.
{scong, nara, alvin}@fit.unimas.my

Abstract

This paper proposes a novel method to disambiguate important words from a collection of documents. The hypothesis that underlies this approach is that there is a minimal set of senses that are significant in characterizing a context. We extend Yarowsky's one sense per discourse [13] further to a collection of related documents rather than a single document. We perform distributed clustering on a set of features representing each of the top ten categories of documents in the Reuters-21578 dataset. Groups of terms that have a similar term distributional pattern across documents were identified. WordNet-based similarity measurement was then computed for terms within each cluster. An aggregation of the associations in WordNet that was employed to ascertain term similarity within clusters has provided a means of identifying clusters' root senses.

1. Introduction

Word sense disambiguation (WSD) is a two-step process; firstly, identifying possible senses of the candidate words then selecting the most probable sense for the candidate word according to its context. Methods proposed by researchers are divided into corpus-based and dictionary-based approaches.

The corpus-based unsupervised approach as proposed by Yarowsky [13] disambiguates word senses by exploiting two decisive properties in human language: one sense per collocation and one sense per discourse. One sense per collocation indicates that words in the same collocation provide strong indication of the correct sense of a target word, while one sense per discourse picks up word senses of target words that are consistent throughout a document [13].

WordNet is a lexical database that comprises English nouns, verbs, adjective and adverbs. Entries in WordNet are represented as synonym sets (synsets) and the linkages between synsets are in hierarchical form. For instance, noun synsets in WordNet are organized into a IS-A hierarchy, representing the hyponym/hypernymy relationship. Due to its wide-coverage, WordNet is used as knowledge structure in most WSD dictionary-based approaches. WordNet's synsets and its IS-A hierarchy are the main usage of WordNet in WSD. There are works ([1], [3]) that adopted the conceptual density in WordNet's IS-A hierarchy to achieve WSD. These researchers made use of sets of words that co-occur within a window of context in a single document. Basili [3] further incorporated "natural" corpus-driven empirical estimation of lexical and contextual probabilities for semantic tagging into [1].

Patwardhan [9] discussed the use of semantic relatedness formula, namely the Adapted Lesk Algorithm, which considers the overlaps of gloss[1] for words to be disambiguated. They obtained relatedness between candidate senses in the same collocation taking into consideration only nouns, within the window of context.

Table 1: Summary of related works

Authors	Domain/ Context	SP[1]	Approach		
			C[2]	SS[3]	WN[4]
Agirre & Rigau	Collocation	Y		Y	Y
Basili et al.	Collocation			Y	Y
Chua & Kulathuramaiyer	Global				Y
Pantel & Lin	Global		Y	Y	Y
Patwardhan et al.	Collocation			Y	Y
Yarowsky	Collocation & discourse				Y
This paper	Global		Y	Y	Y

[1] Supervised
[2] Clustering
[3] Semantic Similarity
[4] WordNet

Pantel and Lin [8] presented a combination of corpus-based and dictionary-based approach. They introduced Clustering by Committee (CBC) that discovered clusters of words sharing a similar sense. Initially, they formed a tight set of cluster (committee), with its centroid represented as a feature vector. Subsequently, candidate words are assigned to these feature vectors accordingly. Overlapping feature vectors are removed to avoid discovering similar senses. They employed a cosine coefficient of words' mutual information as a similarity measure [8]. Pantel and Lin have further employed these clusters of words as a means of WSD for a corpus. They explore words that are commonly collocated with words belonging to a cluster. They suggest that words co-occurring with clustered words can be seen as belonging

[1]Gloss is the definition and/or example sentences for a synset [8].

to the same context. They then employ one sense per collocation as the WSD mean.

Our work on the other hand is not corpus specific for the WSD process. We identify concept relatedness of terms solely based on semantic relationships in WordNet. Our prime motivation has been the discovery of structures with deeper meaning. These structures are compact representations of the input documents. Comparison of related works and our proposed word sense disambiguation are summarized in Table 1.

2. Word Sense Disambiguation

We extend Yarowsky's one sense per discourse [13] to a set of documents that represents a category. We disambiguate important words from the collection of documents. Our approach of word sense disambiguation tries to identify a set of senses that are significant in characterizing a context. The context here is represented in a cluster. There are three phases in our sense disambiguation algorithm: Feature Selection, Clustering and Semantic Similarity.

2.1 Phase I: Feature Selection

Important words are extracted from the input documents using feature selection. Feature selection is a type of dimensionality reduction technique that involves the removal of non-informative features and the formation of a subset of feature from the original set. This subset of feature is the significant and representative set of the original feature set. The feature selection scheme used is information gain (IG). Debole and Sebastiani [5] define IG as how much information a word contains about a category. The higher IG score shows that a word and a category are more dependent, thus the words are more informative.

The top ten categories of Reuters-21578 are used as the dataset. We only considered nouns in our word sense disambiguation. For filtering, WordNet is used.

2.2 Phase II: Clustering

The goal of clustering is to produce a distinct, intrinsic grouping of data, such that similar data is assigned to the same cluster. Distributional clustering is used to find the structure in the set of words formed in Phase I. Distributional clustering [2] grouped words into clusters based on the probability distribution of each word in different class label. The probability distribution for word W in category C is P(C|W), that is the probability that category C will occur given word W. P(W) is the probability of a word W occurred in the corpus. Assumptions made are words, W are mutually exclusive and P(W) is an independent event. The algorithm used is from [2]. However, a small modification is done in order to obtain *n* clusters. The resulting algorithm is:

1. Sort words (obtained from phase I) by its IG score.
2. Assign $n + 1$ clusters with the top $n + 1$ words (as singleton).
3. Loop until all words have been assigned.
    a. Merge 2 clusters which are most similar (based on Kullback Leibler divergence to the mean).
    b. Create a new singleton from the next word in the list.
4. Merge 2 clusters which are most similar (based on KL divergence to the mean), resulting *n* clusters.

The probabilistic framework used is Naïve Bayes and the measure of distance employed is Kullback-Leibler (KL) Divergence to the Mean. KL Divergence to the Mean [2] is the average of KL Divergence of each distribution to their mean distribution. It improves KL Divergence's odd properties; not symmetric and infinite value, if an event has zero probability in one of its distributions.

2.3 Phase III: Semantic Similarity

Unlike [7] that employed co-occurrence of words in documents after clustering, we explore the use of semantic similarity as a quantitative indicator for sense disambiguation.

We proposed that word with similar distribution pattern in a corpus are likely to share the similar set of synsets. Therefore, words in the same cluster with target word should be able to provide a strong and consistent clue for disambiguating the target word in the context. For each pair of words in a cluster, the semantic similarity (SS) of the most similar WordNet senses, ranging from 0 to 1, is obtained. The candidate sense, *i* of a word, *W* is a sense that has semantic similarity with senses of other words in the same cluster. The semantic similarity value is taken as a score. The sense with the highest accumulated score is selected as the most probable sense for the target word, in the cluster.

1. For each cluster C
    a. For each word-pair in cluster *C*
        i. Calculate semantic similarity for all possible senses and return the sense with highest semantic similarity.
2. For each cluster C
    a. For each word W in cluster C
        i. For each sense, *i* of word, *W*, add *SS* to *SSi*.

b. Return sense, *i* with highest *SSi* as most likely sense for word, *W*

Semantic similarity of two concepts is dependent on their shared information [10] that is the Most Specific Common Abstraction (MSCA) and information content (IC). MSCA represents the closest node that subsumes both concepts in the taxonomy while IC indicates how informative a concept is. Seco [11] deduced that a concept in WordNet that has more hyponyms convey less information than concepts that are leaf. Therefore, concepts that are leaves in WordNet are the most specified. He also discussed that the IC of the MSCA based on WordNet, IC_{wn} is

$$1 - \frac{log(hypo(c) + 1)}{log(max_{wn})}$$

where hypo(c) returns the number of hyponyms, given concept, c and max_{wn} is the number of concepts available in WordNet. We used WordNet 2.0 for implementation and there are 79689 nouns in WordNet 2.0

In accordance with previous research by Seco in which 12 semantic relatedness algorithms are benchmarked, an improved version of Jiang and Conrath (with IC_{wn}) similarity measure has the highest correlation with human judgements, provided by Miller and Charles [11]. Therefore, this formula of semantic similarity is used here.

$$sim_{jcn}(c_1, c_2) = 1 - \left(\frac{ic_{wn}(c_1) + ic_{wn}(c_2) - 2 \times sim_{res'}(c_1, c_2)}{2}\right)$$

where $ic_{wn}(c)$ is the IC of the concept based on WordNet and $sim_{res'}(c_1, c_2) = \max_{c \in S(c_1, c_2)} ic_{wn}(c)$.

3. Evaluation and results

As an example, we provide a cluster in category Earn, namely *earnC2*, to illustrate our algorithm. Members of the cluster *earnC2* are:
{*record, profits, loss, jan, split, income, sales, note, gain, results, th, vs, cts, net, revs, quarter, dividend, pay, sets, quarterly, profit, tax, prior, earnings*}.

In WordNet, the target word, *loss* has eight senses. Based on words co-occurring together in *earnC2*, sense 3 of the word loss, is selected by our algorithm and it is closest in the meaning of context of *earnC2*.

sense 1: the act of losing;
sense 2: something that is lost;
sense 3: loss, red ink, red -- the amount by which the cost of a business exceeds its revenue;
sense 4: gradual decline in amount or activity;
sense 5: loss, deprivation -- the disadvantage that results from losing something;
sense 6: personnel casualty, loss -- military personnel lost by death or capture
sense 7: the experience of losing a loved one;
sense 8: passing, loss, departure, exit, expiration, going, release -- euphemistic expressions for death;

Two evaluations on the accuracy of results generated are undertaken; qualitative approaches based on manual inspection and automatic text categorization.

3.1 Qualitative approach

In qualitative approach, we examined the accuracy of the results produced by this algorithm by providing nine human judges with four clusters. The possible senses of each word, extracted from WordNet, are provided as well. The human judges are not informed of the categories the clusters represented. Using others words in the same clusters as the only clues, human judges were asked to select the most appropriate sense for the target word.

The results from each human judge are compared with the results generated by our algorithm. The score for each cluster is then normalized according to the number of words in the cluster. The average scores obtained by nine human judges are shown in Table 2.

Despite the providing of a set of terms corresponding to a cluster of related terms, the human subjects chose senses that represent the typical meaning of these words. For example: the Internet sense of the word 'Net' was chosen rather than the financial sense. We repeated the same evaluation on a human judge that has knowledge in the dataset used. The financial sense is chosen. Therefore this study has highlighted the need for human subjects with a deeper understanding of the domain to conduct the evaluation.

Table 2: Qualitative and baseline approach experimental results

Cluster	Accuracy	
	Qualitative (without knowledge)	Qualitative (with knowledge)
earnC2	0.708	0.875
crudeC2	0.635	0.857
cornC2	0.642	0.935
tradeC2	0.583	0.667
Average	**0.642**	**0.837**

3.2 Automatic Text Categorization

Based on the results of WSD, words within a cluster, which have semantic similarity above 0.45, were identified. These terms were then used as a feature set of the document category. We compared the text categorization results of using the semantically related terms (employing WSD) with the original result of feature selection using Information Gain (without WSD). The experiment was carried out using WEKA (Waikato Environment for Knowledge Analysis) by applying multinomial Naïve Bayes classifier. The experimental

setup employed is the same as used in [4]. The f-measure for each category is shown in Table 3.

Table 3: Automatic text categorization experimental result-F measure

Category	Without WSD		Employing WSD	
	Accuracy	#	Accuracy	#
acq	**0.942**	50	0.883	30
corn	0.314	50	**0.356**	30
crude	0.734	50	**0.751**	35
earn	**0.964**	50	0.958	30
grain	0.603	50	**0.621**	35
interest	**0.565**	50	0.551	35
money-fx	0.603	50	**0.649**	30
ship	0.603	50	**0.641**	40
trade	0.569	50	**0.642**	35
wheat	0.362	50	**0.381**	30

Number of feature

The results highlights that that the set of features employing WSD was not only able to capture the inherent semantics of the entire feature set, it has also been able to remove noise, whereby the performance was better for seven out of ten categories. The results also proved the ability of this reduced feature set in representing the context of documents. The newly formed feature sets have been reduced to the range of 30 to 40 (about 60% to 80%) semantically related features per category.

4. Conclusion

In this paper, we presented a word sense disambiguation algorithm based on semantic similarity using WordNet, which has been applied to collection of documents. The results of this algorithm are promising as it is able to capture root meanings of document collections. The results from text categorization also highlight the ability of our approach to capture contextual meanings of word from document collections.

5. References

[1] Agirre, E. & Rigau, G., "A Proposal for Word Sense Disambiguation Using Conceptual Distance", *Proceedings of Recent Advances in NLP (RANLP95), Tzigov Chark (Bulgary)*, 1995, pp. 258-264.

[2] Baker, L. D. & McCallum, A. K., "Distributional Clustering of Words for Text Classification", *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia*, 2002.

[3] Basili, R., Cammisa, M. & Zanzotto, F. M., "A Similarity Measure for Unsupervised Semantic Disambiguation", *Proceedings of Language Resources and Evaluation Conference, Lisbon, Portugal*, 2004.

[4] Chua, S. & Kulathuramaiyer, N., "Semantic Feature Selection Using WordNet", *2004 IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China*, 2004, pp. 166-172.

[5] Debole, F. & Sebastiani, F., "Supervised Term Weighting for Automated Text Categorization", *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing, Melbourne*, ACM Press, New York, US, 2003, pp. 784--788.

[6] Jiang, J. J. & Conrath, D. W., "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy", *Proceedings of International Conference Research on Computational Linguistics (ROCLING X), Taiwan*, 1997.

[7] Lin, D. "Using syntactic dependency as a local context to resolve word sense ambiguity", In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, 1997.

[8] Pantel, P. & Lin, D., "Discovering Word Senses from Text", In *Proceedings of ACM SIGKDD 02 International Conference on Knowledge Discovery & Data Mining, Edmonton, Alberta, Canada*, 2002.

[9] Patwardhan, S., Banerjee, S. & Pedersen, T., "Using Measures of Semantic relatedness for Word Sense Disambiguation", *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City*, 2003.

[10] Resnik, P., "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995, pp. 448-453.

[11] Seco, N., Veale, T. & Hayes, Jer., "An Intrinsic Information Content Metric for Semantic Similarity in WordNet", *Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence, Valencia, Spain*, 2004.

[12] WordNet, "Glossary of terms", http://wordnet.princeton.edu/gloss, 2005.

[13] Yarowsky, D., "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods", *Proceedings of ACL'95*, 1995.

ANALYSIS OF AGENTS FROM CALL TRANSCRIPTIONS OF A CAR RENTAL PROCESS

Swati Challa

Dept. of Computer Science & Engg.,
IIT Madras,
Chennai-600036, India.
swati.iitm@gmail.com

Shourya Roy, L. Venkata Subramaniam

IBM India Research Lab,
IIT Delhi, Block-1,
New Delhi-110016, India.
{rshourya,lvsubram}@in.ibm.com

ABSTRACT

Telephonic conversations with call center agents follow a fixed pattern, commonly known as call flow. Each call flow is a sequence of states such as greet, gather details, provide options, confirm details, conclude. We present a mechanism for segmenting the calls into these states using transcription of the calls. We also evaluate the quality of segmentation against a hand tagged corpus. The information about how the agents are performing in their calls is crucial to the call center operations. In this paper we also show how the call segmentation can help in automating the monitoring process thereby increasing the efficiency of call center operations.

1. INTRODUCTION

Call centers provide dialog-based support from specialized agents. A typical call center agent handles a few hundred calls per day. While handling the calls the agents typically follow a well-defined call flow. This call flow specifies how an agent should proceed in a call or handle customer objections or persuade customers. Within each state the agent is supposed to ask certain key questions. For example, in a car rental call center, before making a booking an agent is supposed to confirm if the driver has a valid license or not. Call centers constantly monitor these calls to improve the way agents function and also to analyze how customers perceive their offerings. In this paper, we present techniques to automatically monitor the calls using transcriptions of the telephonic conversations. Using NLP techniques, we automatically dissect each call into parts, corresponding to the states mentioned in call flow. We provide a quantitative measure to evaluate how well the call flow has been followed. We also propose a simple technique to identify if key questions are being asked within each segment or not. Using this automatic technique, we show that we are able to identify lapses on the part of the agent. This information is crucial to the call center management as it allows them to identify good and bad agents and train them accordingly. We evaluate the performance of our technique and show that it achieves good accuracy.

2. BACKGROUND AND RELATED WORK

2.1. Text Segmentation

Automatically partitioning text into coherent segments has been studied extensively. In [5] segmentation is done based on the similarity of the chunks of words appearing to the left and right of a candidate. This approach called TextTiling can be used to identify subtopics within a text. In [2] a statistical model is presented for text segmentation.

2.2. Key Phrase Extraction

Extracting sentences and phrases that contain important information from a document is called key phrase extraction. Key phrase extraction is an important problem that has been studied extensively. For example, in [4] the key phrases are learnt based on a tagged corpus. Extraction of key phrases from noisy transcribed calls has also been studied. For manually transcribed calls in [7] a phrase level significance estimate is obtained by combining word level estimates that were computed by comparing the frequency of a word in a domain-specific corpus to its frequency in an open-domain corpus. In [9] phrase level significance was obtained for noisy transcribed data where the phrases are clustered and combined into finite state machines.

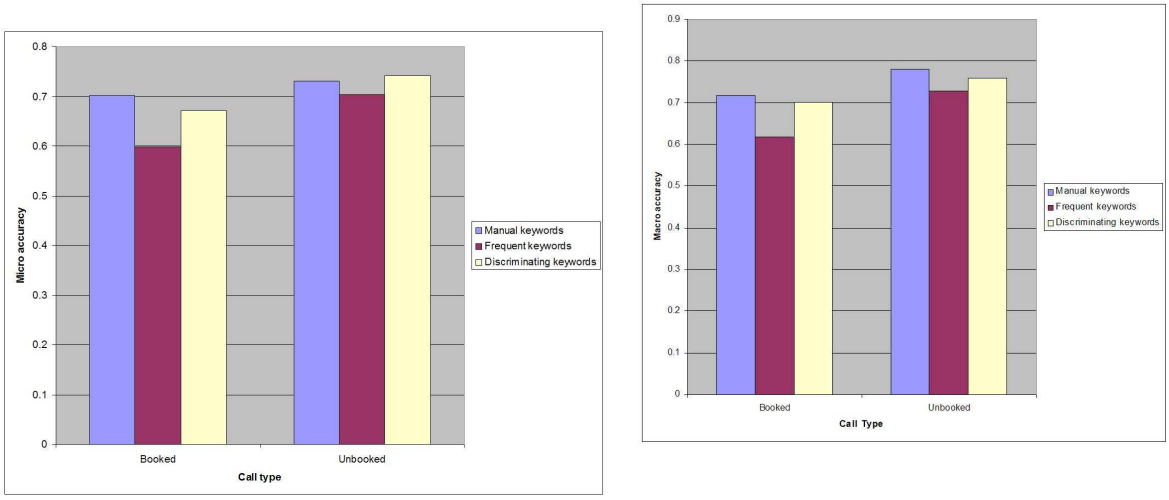


Figure 1: Micro and macro segmentation accuracy using different methods

2.3. Processing of Call Center Dialogs

A lot of work on automatic call type classification for categorizing calls [8], call routing [6], obtaining call log summaries [3], agent assisting and monitoring [7] has appeared in the past. In [1] call center dialogs have been clustered to learn about dialog traces that are similar.

3. CALL SEGMENTATION

A call can typically be broken up into *call segments* based on the particular action being performed in that part of the call. Here we present a mechanism for automatically segmenting the calls and evaluating the quality of segmentation against a hand tagged corpus. The calls are in XML format with relevant portions marked. Any given call can be divided into a maximum of nine segments. They are *Greeting, Pickup-return details, Membership, Car options and rates, Customer objection and objection handling, Personal details, Confirm specifications, Mandatory checks and details and Conclusion*. From the training set of documents which are segmented manually we extracted two sets of keywords for each segment:

- *Frequent keywords* obtained by taking the trigrams and bigrams with the highest frequency in each segment. Unigrams are avoided because most of the high frequency words are stopwords(like the,is etc).

- *Discriminating keywords* obtained by taking the ratio of the frequent phrases(includes unigrams, bigrams and trigrams) in a particular segment to their frequency in the whole corpus with preference being given to trigrams. The top 10 or 20 words are chosen as keywords for each segment.

Using these keywords we segment the booked and unbooked calls automatically with the knowledge of call flow by marking the begin and end of each segment with the corresponding XML tags.

Accuracy
To evaluate the accuracy of this segmentation we compare its performance with the manual segmentation. The accuracy metrics used are :

- *Micro Efficiency* is computed as

$$microEff = \frac{\sum \frac{turnsMatch}{turnsCount}}{n}$$

- *Macro Efficiency* is computed as

$$macroEff = \frac{\sum \frac{\sum \frac{turnsMatchInASegement}{turnsInASegment}}{segmentCount}}{n}$$

where, turnsMatch = No. of turns (one continuous line/sentence spoken by agent/customer) where automatically assigned segment is the same as manually assigned segment
turnsCount = Total no. of turns in a call
n = Total no. of calls in the test corpus
turnsMatchInASegment = No. of turns within each segment where automatically assigned segment is the same as manually assigned segment
turnsInAsegmentMatch = No. of matches in a correct manual segment

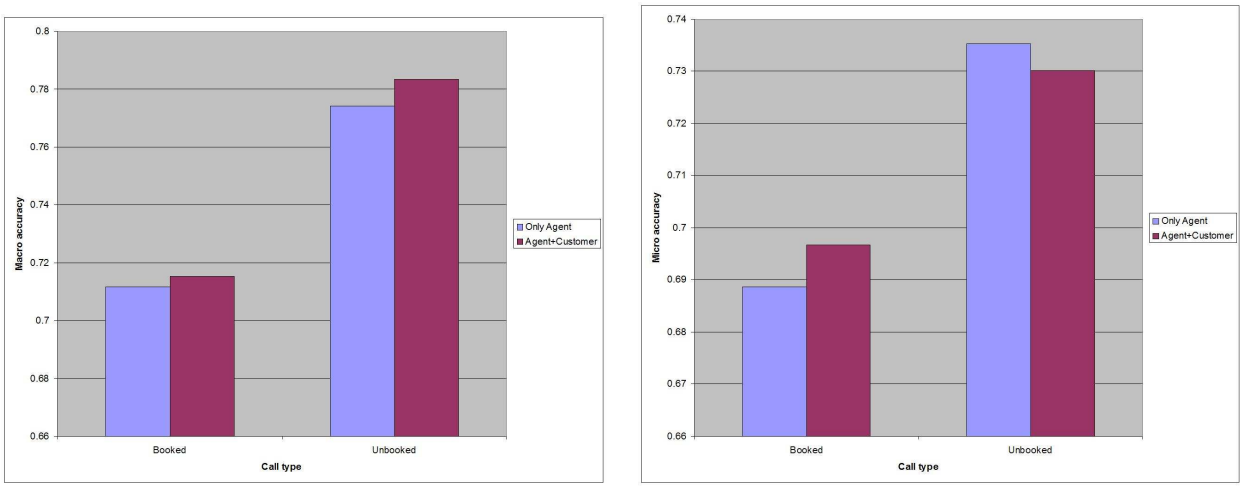


Figure 2: Segmentation accuracy on agent data and combined agent+customer transcriptions

segmentCount = No. of correct segments in the call
From Figure 1 we can see that the segmentation accuracy for manually chosen keywords is almost same as that of discriminating keywords.

4. SEGMENTATION ON AGENT DATA ALONE

The transcribed data is very noisy in nature. Since it is spontaneous speech there are repeats, false starts, a lot of pause filling words such as um and uh, etc. Further there are no punctuations and there are about 5-10% transcription errors. The ASR(Automatic Speech Recognition) system used in practice gives a transcription accuracy of about 60-70%. The number of agents in a call center are limited in number. So, we can train the ASR system on agents voices to increase the transcription accuracy to 85-90%. So if we do the segmentation on agent data alone the accuracy will be much higher compared to agent+customer data because of the low transcription accuracy of the customer. Here we extract only the agent conversation part from the corpus and repeat the above segmentation process to get the efficiency. From the results in Figure 2 we can see that the segmentation efficiency is almost equal to the efficiency using the original call transcription with both agent and customer. This is in case of manually transcribed calls.

5. EVALUATION OF AGENTS

In this section we will show how the call segmentation can help in automating the monitoring process. To see how effectively we can perform using segmentation we take a set of 60 calls and divide them manually into two sets depending on whether the call contains the key task or not. Now for each key task we look for the specific keywords in the corresponding positive and negative instances of the key task separately. For example, to check if the agent has confirmed that the credit card is not a check/debit card we can look for the keywords *check,cheque,debit,which is not,that is not*. We search for the corresponding key words in a particular segment where the key task is supposed to be present (for eg, confirming if the customer has a clean driving record should be present in *mandatory checks and details* segment) and compare the result with the keywords matches in the entire call.The comparison results are shown below for the following key tasks:

1. Ask for sale
2. Confirm if the credit card is not a check/debit card
3. Ask the customer for future reservations
4. Confirm if the customer is above 25yrs of age
5. Confirm if the customer has a major credit card in his own name

From the statistics for negative instances we can see that there are a large number of instances which are wrongly detected as containing the key task without segmentation because the keywords are likely to occur in other segments also. For example, consider the key task

Key Task	#1	#2	#3	#4	#5
No. of Calls	19	28	22	40	12
With Seg.	18	28	22	40	12
Without Seg.	18	28	22	40	12

Table 1: Statistics for positive instances

Key Task	#1	#2	#3	#4	#5
No. of Calls	41	32	38	20	48
With Seg.	38	32	38	20	48
Without Seg.	35	19	1	12	1

Table 2: Statistics for negative instances

3 i.e. the agent asking the customer for future reservations we look for keywords like *anything else,any other,help,assist*. These are likely to occur in other segments also like greeting etc. So by looking at the entire call it is not possible to capture the information if the agent has performed a particular key task or not.Hence by automating the agent monitoring process we can increase the efficiency of call center operations.

6. CONCLUSIONS

We have tried different approaches for automatically segmenting a call and obtained good segmentation accuracy. We showed that we can achieve the same segmentation accuracy using agent data alone which will reduce the transcription errors to a great extent. We also showed that segmentation helps in automating the agent evaluation process thus increasing the efficiency of call center operations.

7. FUTURE WORK

In future we plan to explore other segmentation techniques like Hidden Markov Models for automatically capturing the state information thus automatically extracting the call flow. We intend to reduce the effect of transcription errors in segmentation by using spell check techniques. So far we have hand coded the key tasks from the agent monitoring form. We also hope to automate this process.

8. REFERENCES

[1] F. Bechet, G. Riccardi, and D. Hakkani-Tur. Mining spoken dialogue corpora for system evaluation and modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,*, Barcelona, Spain, July 2004.

[2] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34:177–210, 1999.

[3] S. Douglas, D. Agarwal, T. Alonso, R. M. Bess, M. Gilbert, D. F. Swayne, and C. Volinsky. Statistical models for text segmentation. *IEEE Transaction on Speech and Audio Processing*, 13(5):652–660, 2005.

[4] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence,*, pages 668–673, San Fransisco, CA, 1999.

[5] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics,*, 23(1):33–64, 1997.

[6] H.-K. J. Kuo and C.-H. Lee. Discriminative training of natural language call routers. *IEEE Transactions on Speech and Audio Processing,*, 11(1):24–35, 2003.

[7] G. Mishne, D. Carmel, R. Hoory, A. Roytman, and A. Soffer. Automatic analysis of call-center conversations. In *Proceedings of the Conference on Information and Knowledge Management,*, Bremen, Germany, October 31-November 5 2005.

[8] M. Tang, B. Pellom, and K. Hacioglu. Call-type classification and unsupervised training for the call center domain. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop,*, St. Thomas, US Virgin Islands, November 30-December 4 2003.

[9] J. Wright, A. Gorin, and G. Riccardi. Automatic acquisition of salient grammar fragments for call-type classification. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop,*, Rhodes, Greece, September 1997.

Integration Techniques for multimodal Speech and Sketch map-based system.

Loh Chee Wyai
cheewyai@gmail.com

Alvin W. Yeo
alvin@fit.unimas.my

Narayanan K.
nara@fit.unimas.my

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Malaysia

Abstract

As described by Oviatt et al. (1997), when two or more modalities work together, the integration techniques used for combining different modalities into a whole system is very important. The integration techniques are the main determinants in guiding the design of the multimodal system. To resolve a map-based multimodal system, we propose to use natural language processing (NLP) to identify and describe the syntax and the semantics of prepositions within the spoken speech, in particular, when it relates to maps or directions. From the results, the preposition syntactic and the semantic behaviours will be used to integrate with the sketch. This integration technique is expected to produce a better solution in map-based multimodal system. One of the possible frameworks to be used is PrepNet for processing the speech-text.

1. Introduction

Speech and sketch are two modalities that humans naturally use to communicate with each other especially when they want to relate certain items like locating a place in the map or passing some information, which requires sketching a diagram for better understanding. Sketching with pen and paper comes naturally to most of us (Li et al., 2005) and speech is a main medium in our daily human communication (Atsumi et al., 2004). The combination of these two modalities allows much more precise and robust recognition (Zenka and Slavík, 2004). Yet the integration technique in determining the correct pair of multimodal inputs remains a problem in multimodal fusion of both speech and sketch (Oviatt et al., 1997).

2. Related Work

- **INTEGRATION TECHNIQUE WITH TIME INTERVAL**

Currently there are a few methods used to integrate the speech with sketch in multimodal systems. The commonly used methods are the Unification-based Multimodal Integration Technique in resolving multimodal fusion (Oviatt et al., 2000; Oviatt, 1999; Oviatt and Olsen, 1994). In this integration technique, temporal constraint is used as the integration parameter. In order to use this temporal constraint, the speech and sketch inputs need to be time-stamped to mark their beginning and their end. The main drawback here is the use of time interval as the integration parameter. The continuous stream of spoken and sketch inputs would consist of several sentences and sketched objects that might not be arranged in the correct order.

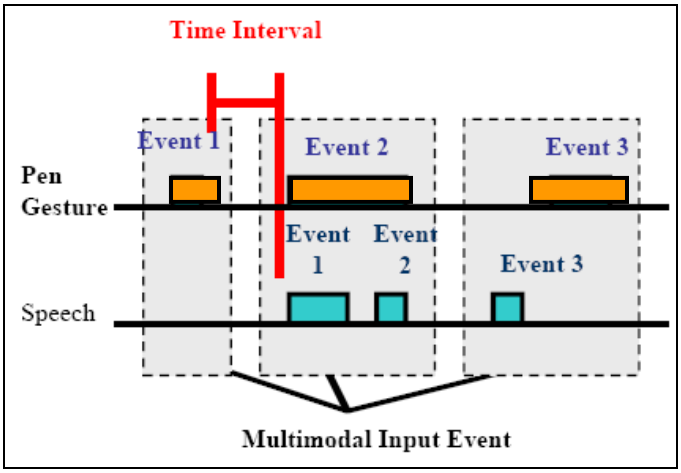


Figure 1: Integration technique using time interval adopted from (Lee and Yeo, 2005).

For a map-based system, events overlapping will occur frequently. For example in Figure 1, this is a condition where inputs from one event interfere in the time slot allocated for other event. This condition is shown in event 1 and 2 in Figure 1 where speech input in event 1 interferes in the event 2's times slot. This will lead to the wrong pairing of the mode inputs. For instance, the pen gesture input for event 1 happen to be no pairing as the speech input did not happen during the time interval and the pen gesture input for event 2 would be paired with speech input for event 1 and event 2 since the time interval between them is the same. Thus, this will lead to the wrong pairing of inputs.

In addition, the discarding of correct inputs would occur when the second input complemented to the first input in the same event did not occur within the preset time interval. Normally the time interval used is 4 seconds after the end of the first input, as used in unification integration architecture. If a sketch input occurs first, then the system would wait for 4 seconds to capture the corresponding speech to complete the event and vice versa. If no input were detected within that time interval, then the whole event (presently with only one input) would be discarded. This condition is shown in event 1, although there is a pair of speech and pen gesture inputs for event 1, but since the spoken input did not occurred within the time interval, then the pen gesture event would be cancelled. This would lead to the incorrect discard of inputs when the input actually occurred out of the time interval. Four

conditions that fail in fulfilling the time interval integration for a map-based system are shown in Figure 2.

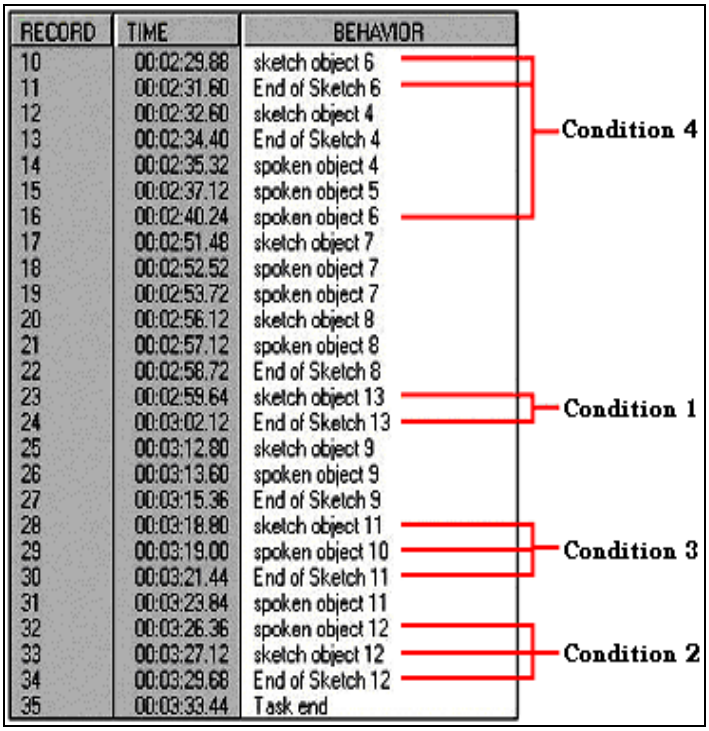


Figure 2: Different types of conditions occurring in speech and sketch when using multimodal map-based system adopted from (Lee and Yeo, 2005).

The first condition (Condition 1) is the absence of speech events for an object. This condition occurs when users did not talk about the object. The second condition (Condition 2) is where the speech event occurs before the onset of the sketch event for the same object. In this condition, based on the time interval integration technique, it would only accept the speech event for the object after the user's sketch event. Therefore, this speech event is not successfully found though it actually occurred. The failure in accepting the speech input leads to the corresponding sketch input being discarded. The third condition (Condition 3) occurs when the wrong pair of speech and sketch events is integrated. This condition normally happens when users described more than one object while performing a sketch event. The last condition (Condition 4) is where speech or sketch event for an object does not occur within the time-out interval (4 seconds). This occurrence is directly discarded by this integration technique that is based on time-out interval.

- **INTEGRATION TECHNIQUE WITH SPATIAL QUERY**

Lee and Yeo (2005) propose a technique, using spatial integration to match both the speech and sketch inputs in a map retrieval system. In this integration technique, an Object Identification process is used to identify the occurrences of the spatial object within sentences. If the object name is found within the sentence, this name would be captured, stored as a new spatial object and the sentences are then broken down into words. Language Parsing Rules in natural language processing (NLP) is adapted to identify the occurrence of the objects within the sentences.

However, the grammar parsing rules adapted by Lee and Yeo (2005) here is only the basic parsing rule, in which simple syntactic approach is used to interpret the speech inputs. The rule is mainly used to extract the objects from the sentences and only the preposition of location is accepted as an element of spatial information. Then, the preposition is checked to describe the relationships between objects in terms of topology, direction, and distance.

The sketch inputs for the system were limited to spatial object with polygon data type in the spatial scene only. Topological, relative directional and relative distance relations are taken into account as shown in the Table 1 below.

Reference Object	Object ID	Topology	Direction	Distance (map unit)
1	2	Disjoint	Southwest	More than 0.00
2	1	Disjoint	Northeast	More than 0.00

Table 1: Object-relation using Topological, relative directional and relative distance relations.

Then, the preposition relationships between objects in terms of topology, direction, and distance were used to integrate the sketch inputs of topology, relative direction and relative distance, as shown at Figure 2 below.

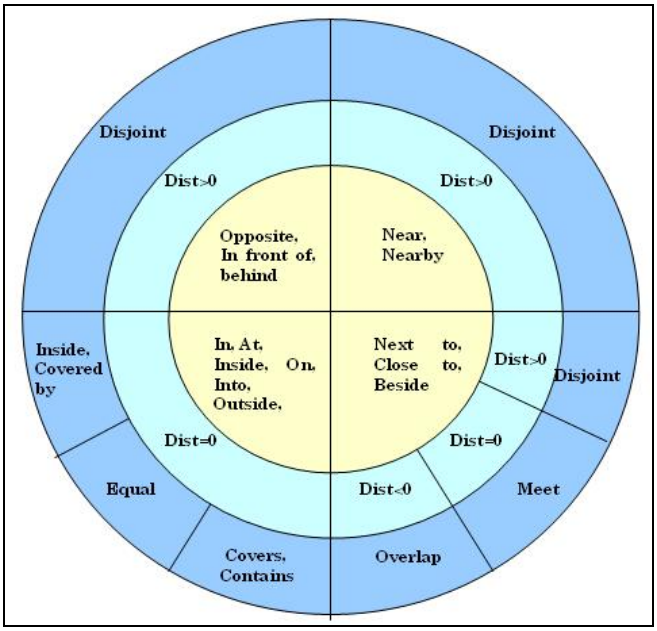


Figure 3: Topological model (adopt from Lee and Yeo, 2005).

Based on Table 2, the success rate from Lee and Yeo (2005) using this integration technique is around 52% success rate compare to Unification-based Integration Technique.

	Success rate (%)	Success rate (%)
Analysed items	Unification-based Integration Technique	Spatial Information Integration Technique
Integration accuracy	36	52
Integrated spatial object	36	50
Integrated spatial description	63	100

Table 2: Summary of results obtained from Unification-based and Spatial Information integration technique.

3. Proposed Integration Technique

We propose the prepositions be used because the prepositions used in a map-based system appears to have very useful categories such as knowledge extraction since they convey basic meanings of much interest like localisations, approximations, means, etc. Thus, a semantic preposition grammar-parsing rule can be applied to cope with the needs in interpreting speech inputs. By categorizing the different senses that prepositions may have in a map-based system, the possibilities of a correct integration are likely to occur. PrepNet is one of the frameworks can be used to describe the syntax and semantics of prepositions used to interpret speech inputs in a map-based system.

As for Lee and Yeo (2005) spatial query technique, the limitation is that the semantics of words was not taken into consideration.

For example the preposition of *next to*, the results obtained from PrepNet contains facet, gloss, syntactic and semantic frame. The facet and gloss result from PrepNet defines *next to* as precise position, which A is in contact with B. The syntactic frame shown A next to B and the semantic frame shown A: next to (D, B), where D is location and B is place or thing. With these extra information extracted, the sketched objects will get more information to relate its objects with the speech, which suggest a higher chance of accurate integration.

As for the sketch, users normally use lines to represent roads, rivers, or railway tracks and polygons represent regions, boundaries, or buildings (Blaser, 2000). Semantic sketch recogniser, Vectractor is used to identify lines and polygon that are available on a map. By using Vectractor, all the lines and polygons identified were represented as buildings, roads and rivers in Scalable Vector Graphic (SVG) format. These can be used to match with the preposition results from the processed speech input as a possible integration technique.

For example an object A is *next to* object B. By using SVG, the objects were not only identified as polygons or lines; it will also identify the relative distance between the objects, by calculating the coordinates available in SVG format between the objects. If the relative distance between object A and object B falls into the category of *next to*, which A is in contact with B in a relative distance, then the results will be used to match with the results from the speech.

4. References

[1] Atsumi Imamiya, Kentaro Go, Qiaohui Zhang, Xiaoyang Mao (2004). Overriding Errors in a Speech and Gaze Multimodal Architecture. In Department of Computer and Media University of Yamanashi, Japan. *Proceedings of IUI 2005*, Madeira, Funchal, Portugal.

[2] Blaser, A. (2000). *Sketching Spatial Queries*. PhD Thesis. National Center of Geographic Information and Analysis, Orono, University of Maine.

[3] Lee B., Yeo A., (2005). Integrating Sketch and Speech Inputs using Spatial Information. *Proceedings of ICMI 2005*, Trento, Italy.

[4] Lee B., Yeo A., (2005). Multimodal Spatial Query. Master Thesis. Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia.

[5] Li J., Dai G., Xiang, A., Xiwen Zhang (2005). Sketch Recognition with Continuous Feedback Based On Incremental Intention Extraction. In Intelligence Engineering Lab & Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. *Proceedings of IUI 2005*, San Diego, California, USA.

[6] Oviatt, S. L. and Olsen, E. (1994). Integration Themes in Multimodal Human-Computer Interaction. In Shirai, Furui, and Kakehi (eds.), *Proceedings of the International Conference on Spoken Language Processing*, 2, pp. 551-554. Acoustical Society of Japan.

[7] Oviatt, S. L., DeAngeli, A., and Kuhn, K. (1997). Integration and Synchronization of Input Modes During Multimodal Human-Computer Interaction. *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)*, pp. 415-422. New York: ACM Press.

[8] Oviatt, S. L. (1999). Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. *Proceedings of Conference on Human Factors in Computing Systems: CHI '99*, New York, N.Y.: ACM Press, pp. 576-583.

[9] Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000). Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human Computer Interaction 2000*, Vol. 15, no. 4, pp. 263-322.

[10] Zenka R., Slavik P. (2004). Supporting UI Design by Sketch and Speech Recognition. In Czech Technical

University in Prague, Czech Republic. *Proceedings of the TAMODIA 2004*, Prague, Czech Republic.

Phonological rules of Hindi and Automatic Generation of Pronunciation Dictionary for Speech Recognition

Vishal Chourasia
School of Computer Science
Devi Ahilya Vishwavidhyalaya, Indore 452001, India
&
Tata Institute of Fundamental Research
Homi Bhabha Road, Mumbai 400005, India.
Email: chourasiavishal@yahoo.com

1 Introduction

A pronunciation dictionary is a list of words along with their pronunciation in terms of phonetic units. It is an essential language resource for text-to-speech and speech-to-text conversion programs. This paper deals with some pronunciation rules that capture the regularities in text to phoneme mapping of Devanagari script used by Hindi language and use of these rules in automatic generation of a pronunciation dictionary for use in automatic recognition of Hindi speech.

1.1 Corpus of Phonetically Rich Sentences

For training general purpose, speaker independent speech recognition, speech data need to be collected from a large number of speakers. Hence, there is need for a compact text corpus that consists of sets of sentences that capture all phonetic contexts. Phonetically rich sentences can be selected by a program from a large corpus of text. A set of such sentences were derived by processing the Hindi news archives available at [1]. This process involved transliterating the UTF-8 code employed by the online newspaper, retaining short sentences, and generating sets of sentences that are phonetically rich (i.e., the sentence set contains most phonemes of the language) [2].

1.2 Text Normalisation

Text normalisation is a process in which the text is transformed for further processing; some of the steps are removing punctuation, expanding abbreviations etc. A program was written that converts all the numbers appearing in sentences into its Roman equivalent according to Hindi pronunciation. For example, the number "123" is pronounced as एक सौ तेईस in Hindi, and transliterated to "ek sO teyIs". Such text normalisation for numbers will reduce the size of dictionary.

2 Generation of Pronunciation Dictionary

Hindi uses Devanagari script. There is a **near** one-to-one correspondence between a grapheme and the associated phoneme. However there are such exceptions, formulation of exceptions in the form of rules helps in automatic generation of pronunciation dictionary. In this section, we describe two sets of such rules (and exceptions): pronunciation rules for phoneme /a/ in certain phonetic contexts and grapheme to phoneme rule for 'Anuswara'.

Word	Transliteration	Pronunciation	Conditions for retention of phoneme /a/
सक्रिय	sakriya	s a k r **i y a**	last syllable is "ya" and penultimate
राष्ट्रीय	rAStrIya	r A S t r **I y a**	syllable ends with high vowel I, i, U, u
भाग्य	bhAgya	bh **A g y a**	the last syllable is a consonant cluster
छात्र	chAtra	ch **A t r a**	and the second consonant is a
कत्ल	katla	k a **t l a**	semivowel(/y/, /r/, /l/, /w/)
महत्व	mahatwa	m a h a **t w a**	
यह	yaha	ya **h a**	word-final syllable is "ha"

Table 1: Rules, with examples, for retention of word-final /a/ in Hindi words

2.1 Context Dependent Pronunciation rules for phoneme /a/

The pronunciation rules for phoneme /a/ in Hindi language can be grouped into 2 main categories: deletion and modification.

2.1.1 Deletion of word-final /a/

A phoneme /a/ at the end of a word is normally not pronounced (i.e., deleted) in Hindi language. However, there are exceptions to this rule: If the word-final syllable is a consonant cluster, and ends with a semivowel and phoneme /a/, then the phoneme /a/ is **not** deleted. Table 1 shows the conditions under which a word-final /a/ is retained [4]. We studied the status of /a/ in such cases by inspecting spectrograms of appropriate words spoken by multiple speakers. We discovered that word-final /a/ is to be deleted, when /r/ is the first component of a word final consonant cluster with any semivowel. Also, a word-final /a/ is **not** deleted if the word-final syllable is a cluster of nasals. These additional rules are shown in Table 2.

Word	Transliteration	Pronunciation	Comments
प्रसन्न	prasanna	p r a s a **n n a**	Retain /a/ if the last syllable
जन्म	janma	j a **n m a**	is a cluster of nasal sound
पूर्व	pUrwa	p U **r w**	delete /a/, when /r/ is the first
कार्य	kArya	k A **r y**	component of word final consonant cluster with any semivowel

Table 2: Pronunciation rules discovered in this work

2.1.2 Deletion of /a/ in a middle syllable

If a word consists of 3 of more syllables, and the last syllable consists of any vowel other than phoneme /a/ , then the penultimate /a/ is to be deleted [3]. For example the word मरने ("marane") [to die] is represented as "m a r n e", after deletion of /a/ from syllable "ra". But, there is an exception to this rule and is explained in the next subsection.

2.1.3 Pronunciation of /a/ in the phoneme sequence "a h a"

If a word consists of 3 of more syllables, and contains the phoneme sequence "a h a" in the middle of the word, then the pronunciation of the first /a/ changes as follows: (a) if the syllable

Word	Transliteration	Phonemic transcription	Transcription after Modification	Hindi Pronunciation
महज	mahaja	m a h a j	m **ea** h a j	मेहज
कहती	kahatI	k a h a t I	k **ea** h **ax** t I	केहती

Table 3: Rules for pronunciation of /a/ in phoneme sequence "a h a"

Place of articulation	Hindi Word	Transliteration	Pseudo-phonemic Transcription	Modified Transcription
velar	अंक	aMka	a M k a	a **ng** k a
palatal	पूंजी	pUMjI	p U M j I	p U **nj** j I
retroflex	घंटे	ghaMTe	gh a M T e	gh a **N** T e
dental	मंत्री	maMtrI	m a M t r i	m a **n** t r i
labial	मुंबई	muMbaI	m u M b a I	m u **m** b a I

Table 4: Replacement of Anuswar by nasals according to the place of Articulation.

following "aha" ends with /a/, "aha" is pronounced as "ea h a" where 'ea' is an allophone of the phoneme /e/. (b) if the syllable following "aha" ends an vowel other than /a/, "aha" is pronounced as "ea h ax" where 'ax' is a schwa. Table 3 explains modification rule (a) and (b) with example words and their actual pronunciations.

2.2 Anuswar to Nasal Consonant Mapping

In Devanagari script, the grapheme 'anuswar' (a dot above the headline of an vowel), denotes a nasal sound following the vowel. The actual nasal to be pronounced depends on the place of articulation of the following consonant. Thus, anuswar-to-nasal is a one-to-many grapheme-to-phoneme mapping. Table 4 illustrates this rule for the 5 places of articulation.

3 Discussion and Conclusion

The set of 50,000 phonetically rich sentences generated as explained in Section 1.1 contain approximately 28,000 unique words. A pronunciation dictionary corresponding to these words were automatically generated using a Perl script that incorporates all the rules described in this paper. In the absence of such a rule set, these modifications have to done manually—this is a time-consuming and laborious job. Thus, a compilation of pronunciation rules help in the development of automatic speech recognition system for Indian languages.

Acknowledgements

The author is very much thankful to Dr. Manohar Chandwani and Dr. (Mrs.) Maya Ingle of Devi Ahilya Vishw avidyalaya, Indore for their support towards the work. The guidance and support by Dr. Samudravijaya K. of Tata Institute of Fundamental Research, Mumbai is highly acknowledged.

References

[1] http://navbharattimes.indiatimes.com/archives

[2] Vishal Chourasia, Samudravijaya K., Manohar Chandwani, "Phonetically Rich Hindi Sentence Corpus for Creation of Speech Database", in *Proc. Int. Conf. On Speech Databases and Assessment*, Jakarta, Indonesia, pp. 132-137, 2005

[3] http://people.w3.org/rishida/scripts/indic-overview

[4] Monojit Choudhury and Anupam Basu, "A Rule-based schwa deletion algorithm for Hindi", in *Proc. Int. Conf. On Knowledge-Based Computer Systems*, Vikas Publishing House, Navi Mumbai, India, pp. 343 - 353, 2002.

Rule based Automated Pronunciation Generator

Ayesha Binte Mosaddeque
Department of Computer Science & Engineering
BRAC University
66 Mohakhali, Dhaka-1212.
Bangladesh
Email: lunaticbd@yahoo.com

Abstract

This paper presents a rule based pronunciation generator for Bangla words. It takes a word and finds the pronunciations for the graphemes of the word. A grapheme is a unit in writing that cannot be analyzed into smaller components. Resolving the pronunciation of a polyphone grapheme (i.e. a grapheme that generates more than one phoneme) is the major hurdle that the Automated Pronunciation Generator (APG) encounters. Bangla is partially phonetic in nature, thus we can define rules to handle most of the cases. Besides, up till now we lack a balanced corpus which could be used for a statistical pronunciation generator. As a result, for the time being a rule-based approach towards implementing the APG for Bangla turns out to be efficient.

1 Introduction

Based on the number of native speakers, Bangla (also known as Bengali) is the fourth most widely spoken language in the world [1]. It is the official language in Bangladesh and one of the official languages in the Indian states of West Bengal and Tripura. In recent years Bangla websites and portals are becoming more and more common. As a result it has turn out to be essentially important for us to develop Bangla from a computational perspective. Furthermore, Bangla has as its sister languages Hindi, Assamese and Oriya among others, as they have all descended from Indo-Aryan with Sanskrit as one of the temporal dialects. Therefore, a suitable implementation of APG in Bangla would also help advancement of the knowledge in these other languages.

The Bangla script is not completely phonetic since not every word is pronounced according to its spelling (e.g., বধ্য /bɔḍḍ^{h}o, মধ্য /mɔḍḍ^{h}o, এখন/ɔk^{h}on, এখানে/ek^{h}ane). These cases can be handled by rules of pronunciation. Therefore, we need to use some pre-defined rules to handle the general cases and some case specific rules to handle exceptions. These issues have been discussed in more details later on.

2 Previous Work

A paper about the Grapheme to Phoneme mapping for Hindi language [2] provided the concept that, an APG for Bangla that maps graphemes to phonemes can be rule-based. No such work has yet been made available in case of Bangla.

Although Bangla does have pronunciation dictionaries, these are not equipped with automated generators and more importantly they are not even digitized. However, the pronunciation dictionary by Bangla Academy provided us with a significant number of the phonetic rules [3]. And the phonetic encoding part of the open source transliteration software ‘pata’ [4] provided a basis.

3 Methodology

In the web version of the APG, queries are taken in Bangla text and it generates the phonetic form of the given word using IPA[1] transcription. Furthermore, there is another version of the system which takes a corpus (a text file) as input and outputs another file containing the input words tagged with the corresponding pronunciation. This version can be used in a TTS[2] system for Bangla.

In terms of generating the pronunciation of Bangla graphemes a number of problems were encountered. Consonants (except for 'শ/ʃ' and 'স/s') that have vocalic allographs (with the exception of 'ে) are considerably easy to map. However there are a number of issues: Firstly, the real challenge for a Bangla pronunciation generator is to distinguish the different vowel pronunciations. Not all vowels, however, are polyphonic. 'অ/ɔ 'and 'এ/e' have polyphones ('অ/ɔ' can be pronounced as [o] or [ɔ], 'এ/e' can be pronounced as [e] or [æ], depending on the context) and dealing with their polyphonic behavior is the key problem. Secondly, the consonants that do not have any vocalic allograph have the same trouble as the pronunciation of the inherent vowel may vary. Thirdly, the two consonants 'শ/ʃ' and 'স/s' also show polyphonic behavior. And finally, the 'consonantal allographs' (য/ɟ, ্র/r, ্ব/b, ্ম/m), and the grapheme 'য়/j' complicate the pronunciation system further.

4 Rules

The rule-based automated pronunciation generator generates pronunciation of any word using rules. As explained earlier, the Bangla script is not completely phonetic in view of the fact that not every word is pronounced in accordance with its spelling. For example, the words 'অনেক/ ɔnek' and 'অতি/ o̯ti' both start with 'অ/ɔ' but their pronunciations are [ɔnek] and [o̯ti] respectively. These changes with the pronunciation of 'অ' are supported by the phonetic rules:

অ + C + ে (এ কার) > অ
অ + ই / C + ি (ই কার) > ও, where C= Consonant

An additional rule related to the pronunciation of 'অ/ɔ' is that if 'অ/ɔ' is followed by 'ন/n' without any vocalic allographs then 'অ/ɔ' is pronounced as [ɔ]. For example, 'অনল/ɔnol', 'অন্ত/ɔnto '.
Another polyphone grapheme is 'এ/e', it has two different pronunciation [e] and [æ]. For example, 'একি/eki', 'একা/æka'. This change of pronunciation is supported by the following pronunciation rule:

ে /এ + C + ই / ি / ঈ / ী /ঊ/ ু / ঊ / ূ / এ / ে / ও/ ে া > এ

ে /এ + া > ্যা, where C= Consonant

[1] International Phonetic Alphabet
[2] Text To Speech

There are some rules that have been developed by observing general patterns, e.g., if the length of the word is three full graphemes (e.g. কলম/kɔlom, খবর/k^{h}ɔbor, বাদলা/ba̤la, কলমী/kolmi etc.) then the inherent vowel of the medial grapheme (without any vocalic allograph) tends to be pronounced as [o], provided the final grapheme is devoid of vocalic allographs (e.g., কলম/kɔlom, খবর/k^{h}ɔbor). When the final grapheme has adjoining vocalic allographs, the inherent vowel of the medial grapheme (e.g. বাদলা/ba̤la, কলমী/kolmi) tends to be silent (i.e., silenced inherent vowels can be overtly marked by attaching the diacritic '্').

Hypothetically, all the pronunciations are supposed to be resolved by the existing phonetic rules. But as a matter of fact they do not; some of them require heuristic assumptions. Apart from the rules found in the pronunciation dictionary by Bangla Academy [3], some heuristic rules are used in APG. They were formulated while implementing the system. Most of them serve the purpose of generating pronunciation for some specific word pattern. All the rules are available in http://student.bu.ac.bd/~u02201011/RAPG1 .

5 Implementation

APG has been implemented in Java (jdk1.5.0_03). The web version of APG contains a Java applet that can be used with any web client that supports applets. The other version of APG is also implemented in Java. Both the versions generate the pronunciation on the fly; to be precise no look up file has been associated. Figure 1 illustrates the user interface of the web version and Figure 2 illustrates the output format of the other version.

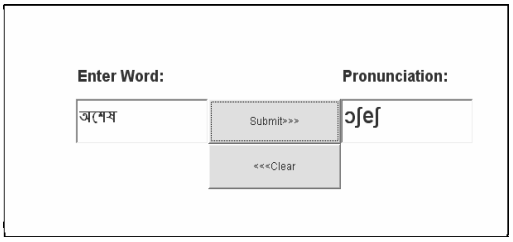


Figure 1 : The web interface of APG. The input word is 'অশেষ ' and generated pronunciation is 'ɔʃeʃ'.

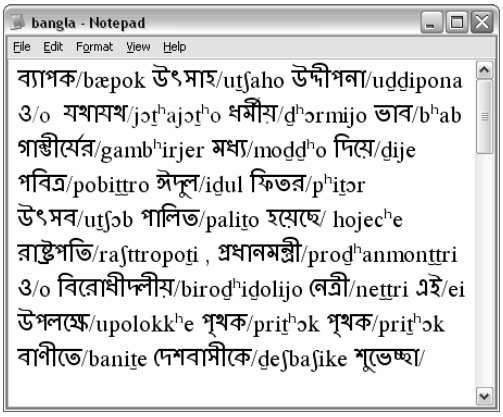


Figure 2 : the output file generated by the plug-in version of APG.

5 Result

The performance of the rule-based APG proposed by this paper is challenged by the partial phonetic nature of Bangla script. The accuracy rate of the proposed APG for Bangla was evaluated on two different corpora that were collected from a Bangla newspaper. The accuracy rates observed are shown in Table 1:

Table 1

Number of words	Accuracy Rate (%)
736	97.01
8399	81.95

The reason of the high accuracy rate of the 736-word corpus is that, the patterns of the words of this corpus were used for generating the heuristic rules. The words in the other corpus were chosen randomly. The error analysis was done manually by matching the output with the Bangla Academy pronunciation dictionary.

6 Conclusion

The proposed APG for Bangla has been designed to generate the pronunciation of a given Bangla word in a rule based approach. The actual challenge in implementing the APG was to deal with the polyphone graphemes. Due to the lack of a balanced corpus, we had to select the rule-based approach for developing the APG. However, a possible future upgrade is implementing a hybrid approach comprising both a rule based and a statistical grapheme-to-phoneme converter. Also, including a look up file will increase the efficiency of the current version of APG immensely. This will allow the system to access a database for look up. That way, any given word will first be looked for in the database (where the correct pronunciation will be stored), if the word is there then the corresponding pronunciation goes to the output, or else, the pronunciation is deduced using the rules.

References

[1] The Summer Institute for Linguistics (SIL) Ethnologue Survey (1999).

[2] Monojit Choudhury, “Rule-based Grapheme to Phoneme Mapping for Hindi Speech Synthesis”. *Proceedings of the International Conference On Knowledge-Based Computer System*s, Vikas Publishing House, Navi Mumbai, India, pp. 343 – 353, 2002. Available online at: http://www.mla.iitkgp.ernet.in/papers/G2PHindi.pdf

[3] Bangla Uchcharon Obhidhan, Bangla Academy, Dhaka, Bangladesh.

[4] Transaliteration Software - Pata, developed by Naushad UzZaman, CRBLP, BRAC University. Available online at: http://student.bu.ac.bd/~naushad/pata.html

Transliteration from Non-Standard Phonetic Bengali to Standard Bengali

Sourish Chaudhuri (sourish@iitkgp.ac.in)
Supervisor: Monojit Choudhury (monojit@cse.iitkgp.ernet.in)
Department of Computer Science and Engineering
Indian Institute Technology Kharagpur,
WB, India-721302.

ABSTRACT

In this paper, we deal with transliterations from non-standard forms of Bengali words written in English to their standard forms. Familiarity of users with standard English keyboards makes it easy for them to represent Bengali words with English letters (i.e. Roman Script). Starting from a list of commonly used Bengali words compiled from a corpus, we obtain a pronouncing dictionary using a Grapheme-to-Phoneme converter. We propose a novel method based on heuristic search techniques over the pronouncing dictionary for transliteration of a word written in Bengali using Roman script to its standard Bengali form. Applications of this technique include design of phonetic keyboards for Bengali, automatic correction of casually written Bengali in Roman English, query correction for Bengali search over the web, and searching loosely transliterated named entity. The techniques used are generic and can be readily adapted to other languages.

1. INTRODUCTION

Bengali is the language of more than 270 million speakers, spoken primarily in Bangladesh and eastern part of India. Nevertheless, there is no single standard keyboard for Bengali that is being used globally to input Bengali text into different electronic media and devices like the computers, cell phones and palm-tops. Besides, the number of letters in the Bengali alphabet is considerably larger than that of English alphabet. Therefore, the common Bengali speakers, who are familiar with the standard English keyboards like QWERTY, find it convenient to transliterate Bengali into Roman script while typing over some electronic media/device. In this work we propose a technique to transliterate a Bengali word written in English (i.e. Roman Script, henceforth RB) to the standard Bengali form (henceforth SB).

There are certain phonetically based standardized encodings for Bengali characters using the Roman scripts [1]. However, the average user, when using transliterated text messages rarely stick to the encoding scheme. Her transliterations are based on the phonetic transcription of the word, and hence, we encounter situations where the same English letter can represent different Bengali letters.

A transliteration scheme that efficiently converts noisy text to standard forms would find application in a number of fields such as:

1. Information retrieval for Bengali language
2. Chat/SMS in Bengali
3. Automatic text to speech synthesis from transliterated texts like chats, SMS, blogs, emails etc.
4. Automatic correction tools for text documents.
5. Design of a phonetic keyboard or an interface for entering Bengali text using QWERTY keyboard

The transliteration scheme might especially help in web searches for named entities. It is quite likely that the name of a person may be spelt differently by different people who are unaware of the exact spelling. In that case, a technique that can recover the actual name overcoming spelling variations would greatly improve the results. For example, the name "Saurav Ganguly" might be spelt by different sources/users as "Sourav Ganguly", "Saurabh Ganguly" or even "Sourabh Ganguly". If all these representations can be mapped to the same name, the efficiency of the search could be further increased.

Given an input word, the decoder should be able to generate the corresponding standard form. We model the problem as a noisy channel process; we assume that the standard word is distorted to the corresponding RB form while being transmitted over the noisy channel. The channel is modeled using a G2P mapper coupled with statistical methods. The novelty of the work lies in use of efficient data-structures and application of heuristic search techniques for fast retrieval of the possible SB forms. Although we report our work for Bengali, the technique is generic and easily adaptable to other languages.

.

2. BACKGROUND

There are several techniques to carry out transliteration and back-transliteration [2-6]. Previously, researchers have built transliteration engines between English and Japanese [2], Chinese [3], Bengali [4], Arabic [5] etc. Most of these works model the problem as a noisy channel process [4]. Phonetic information [3] and other representational information [6] are also commonly

used. However, most of these methods are confined to letter or syllable level phonetic transcriptions.

As we shall see shortly, such methods fail to elicit the correct transliteration in several cases that are encountered quite commonly. N-gram statistics applied over letters or syllables can alleviate the problem to some extent, but statistical methods call for a large amount of parallel data (transliterated and their standard counterpart), which is difficult to acquire. Moreover, the accuracy of the models are dependent on the training data. This results in poor system performance whenever the training and the test data are from different sources or exhibit different distributional patterns. We make use of an accurate word-level G2P converter for Bengali to circumvent the aforementioned problems.

There are several hard issues that need to be solved for transliterating RB to SB. All of these issues crop up due to many-to-many mapping between the RB and SB units. We find that there are cases where one Bengali character may be represented by more than one English characters, and also cases where one English character can stand for more than one Bengali character. For instance, the English 'a' might be used to represent both the Bengali letters[1] 'a' and 'A' e.g. 'jala' (water) and 'jAla' (net) might both be written in RB as 'jal'. Similarly, the Bengali letter 'a' might be represented using both 'a' and 'o' from the English alphabet.

To distinguish between these forms, we require context information in some cases, while in others the disambiguation can be carried out without any context information. In this work, we deal with transliterations only at word level, and hence context based disambiguation is beyond the scope of this paper.

Take the example of the Bengali word 'jala' meaning water. There are two letters in the Bengali alphabet that are pronounced 'ja'. The Itrans representation for one is 'j' while that for the other is 'y'. Thus, for the word, we might have any of the following representations: 'jala', 'jAla', 'yala', 'yAla'. We can use a lexicon to eliminate the possibilities 'yala' and 'yAla'. However, to disambiguate between the other two options, we need to know the context.

Further, there is an inherent Bengali vowel 'a' at the end of most Bengali words that do not end with some other vowel. This vowel is silent in most of the cases - a phenomenon known as *schwa deletion*. The user while transliterating a word in RB relies on the phonetic transcription of the word and might omit it. In cases where more complex letter combinations are used in Bengali (especially the conjugates), letter to letter transliterations may not be applicable. This also leads to a large Levenshtein distance between the word in SB and word in RB. For example, (non-std) khoma → kShama (std).

Further sources of error may be unintentional misspelling.

[1] In this paper, we use ITRANS [1] to represent the standard Bengali forms and owing to the phonemic nature of Bengali, the pronunciations are also written following the same convention.

3. NOISY CHANNEL FORMULATION

In this section, we formally define the problem as a noisy channel process. Let S be a Bengali word in the standard form. When S is transmitted through the noisy channel, it is converted to T, the corresponding word in RB. The noisy channel is characterized by the conditional probability $Pr(T|S)$. In order to decode T to the corresponding standard form $\delta(T)$, where δ is the decoder model, we need to determine $Pr(S|T)$, which can be specified in terms of the noisy channel model as follows.

$$\delta(T) = \underset{S}{argmax}\ Pr(T|S)Pr(S) \qquad (1)$$

The channel can be conceived as a sequence of two sub-channels. First the SB form S is converted to the corresponding phonetic form P, which is then converted to the RB form by the second channel. This is illustrated below.

$$S \rightarrow P = p1p2p6\ldots pr \rightarrow T = t1t2t3\ldots tn$$

The motivation behind this is as follows. When the user thinks of a word in SB, which he wants to represent in RB, it is the phonetic transcription of the word that he represents in the RB

Given the noisy text T, if we want to produce the source word S, we need to reverse the above process. Thus, the expression for the decoder model would be:

$$\delta(T) = \underset{S}{argmax} \sum_{P} [Pr(T|P).Pr(P|S).Pr(S)] \qquad (2)$$

In the case of Bengali, most words have only one phonetic representation, which implies that $Pr(P|S)$ is 1 for a particular P^* = G2P(S) and 0 for all other phonetic strings. Here, G2P represents the grapheme-to-phoneme mapping. Therefore, we can rewrite the Eq. 2 as

$$\delta(T) = \underset{S}{argmax}\ [Pr(T|G2P(S))\ Pr(S)] \qquad (3)$$

In the subsequent sections, we propose a framework based on the above assumption to carry out the transliteration process efficiently.

4. PROPOSED FRAMEWORK

In order to compute $\delta(T)$, we need to compute $Pr(T|G2P(S))$ and $Pr(S)$. The latter quantity is the unigram probability of the word S and can be estimated from a corpus. In order to compute the former quantity, we need a G2P converter (or a pronouncing dictionary) and a model that computes the probability of mapping an arbitrary phonetic string to a string in Roman script. It is interesting to note here that though apparently the probability of mapping a phonetic string into Roman script seems to be independent of the source language (here Bengali), in reality it is hardly so. For example, the phonemes /T/ (retroflex or the hard t) and /t/ (dental or the soft t) are both transliterated as "t" by the

Bengali speakers, whereas Telugu and Tamil speakers use “th” to represent /*t*/ and “t” to represent /*T*/.

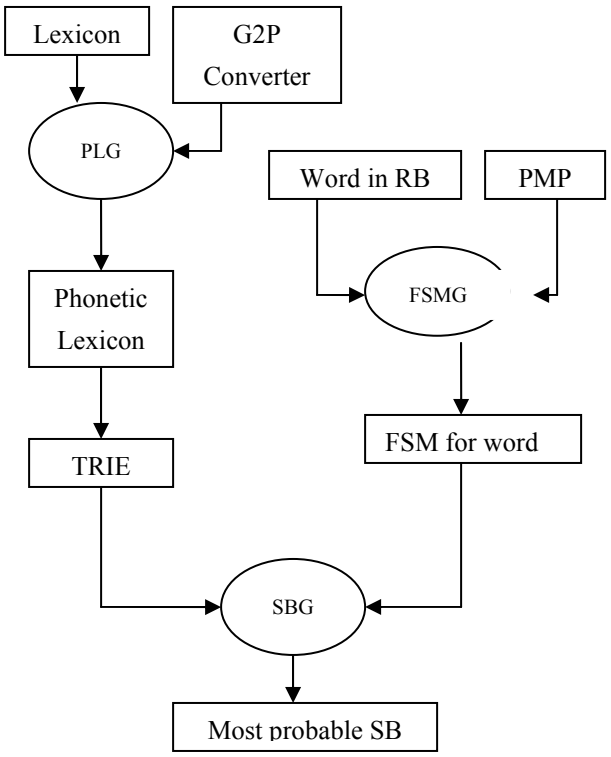


Fig1. Basic Architecture:

In Figure 1,

PLG: Phonetic Lexicon Generator

SBG: Standard Bengali Generator

FSMG: Finite State Machine Generator

Fig 1 shows the basic architecture of the system. A list of Bengali words (shown as lexicon in the figure) is converted to their phonetic counterpart using a G2P. As the preprocessing step, a forward trie is built using the phonetically transliterated words. A probabilistic finite state machine (PFSM) is constructed for T (the RB word) that represents the $Pr(T|P)$. The PFSM and the trie are searched simultaneously to find a $P^* = \text{G2P}(S)$ such that the probability $Pr(T|\text{G2P}(S))Pr(S)$ is maximized over all S.

4.1. Grapheme-to-Phoneme Conversion

The method used to simulate the first stage of the noisy channel formulation is by using a grapheme-to-phoneme converter that gives the phonetic form of the given Bengali word using IPA notations. The G2P used for this work is described in [7]. It is a rule-based G2P that also uses morphological analysis and an extensive list of exceptions. These features make the accuracy of the G2P quite high (around 95% at the word level).

4.2. Resources

A lexicon containing around 13000 most frequent Bengali words and their unigram frequencies have been obtained from the CIIL corpus. Each word is passed through the G2P and their phonetic transliterations are obtained. Thus, we obtain the phonetic lexicon consisting of the words, their phonetic representations and their frequencies.

A modest size transliterated corpus is required in order to learn the probabilities $Pr(e|p)$, where e is an English letter and p is a phoneme used in Bengali. These probabilities are calculated for each of the phonemes that are present in the lexicon. However, for this work we manually assign these probability values.

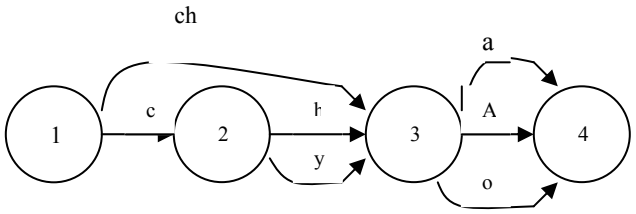
4.3. Representing the lexicon as a trie

The trie[8] is built from the phonetic lexicon consisting of the phonetic forms of the words, and their frequencies. Starting with a root node, transitions are defined using phonemes. Each node N of the trie is associated with a list of Bengali words $\text{W}(N) = \{w_1, w_2, \ldots w_k\}$ (possibly null) such that the unique phonetic string P represented by the path from the root to N is the phonetic transliteration of $\{w_1, w_2, \ldots w_k\}$. That is

$$P = \text{G2P}(w_1) = \text{G2P}(w_2) = \ldots = \text{G2P}(w_k)$$

4.4. FSM for the RB Word

Every English letter can represent one or more phoneme in Bengali. The probabilities of Bengali phonemes mapping to certain English graphemes can be learnt from a corpus. These values are then used to construct a PFSM for the RB word. Transitions in this PFSM are defined on the Bengali phonemes that might be represented by an English letter. A traversal of this PFSM to its final state gives the possible phonetic alternatives of the Bengali word.



State1 & 4: Initial and final states respectively

Figure 2. FSM for ‘*cha*’

This diagram shows the PFSM for the input RB word ‘*cha*’. The first 2 letters ‘*ch*’ may cause a transition form state 1 to 3. Alternatively, it may transition from state 1 to 2 on the possible Bengali phonemes corresponding to the letter ‘*c*’, and then to state 3 on the phonemes corresponding to ‘*h*’. It may then transition to state 4 on the possible phonemes for the letter ‘*a*’.

Note that every path from the start state to the final state is a possible phonetic transliteration of the RB string, the probability

of which is given by the product of the probabilities of the edged in the path.

5. HEURISTIC SEARCH

Argmax searches have been implemented using A* algorithms[9]. In this section, we describe the method used in this case to implement A*.

Let T be the word in RB, which we want to decode. We construct the PFSM M_T corresponding to T. Let $P = p_1p_2...p_n$ be a string of phonemes associated with the transitions $e_1, e_2, ..., e_n$ in M_T such that the transitions, in that order, denotes a path from the start state of M_T to the final state. Therefore, the probability $\Pr(T|P)$ is by definition $\Pi_{i = 1 \text{ to } n} Pr(e_i)$, where $Pr(e_i)$ is the transition probability of the edge (transition) e_i.

In order to compute $Pr(S)$, we need to know the set of words $\{w_1, w_2, ... w_k\}$, such that $G2P(w_i) = P$. This can be computed by searching for the string P in the trie. If N_P is the node in the trie representing the string P, then we define $c(N_P) = Pr(S) = \sum_{w \in W(NP)} Pr(w)$, where $Pr(w)$ is the unigram probability of the word w.

In order to search for the node N_G such that the product of the two probabilities is maximized, we simultaneously explore the M_T and the trie. We tie up the states of M_T with the nodes of trie by marking them with the same levels. Note that a node in the trie gets a unique state label. We define the cost of reaching a node N in the trie as follows.

$$g(N) = \begin{cases} 1, & \text{if } N \text{ is the root} \\ g(par(N)) \times Pr(par(N) \rightarrow N), & \text{otherwise} \end{cases}$$

where $par(N)$ is the parent of node N and $Pr(par(N) \rightarrow N)$ is the probability associated with the transition in M_T from the state tied up with $par(N)$ on the phoneme that connects $par(N)$ to N in the trie.

We define the heuristic function $h(N)$ as follows

$$h(N) = \begin{cases} c(N) & \text{if the node is a leaf node} \\ \sum_{X \in Ch(N)} c(X) + c(N) & \text{otherwise} \end{cases}$$

where, $Ch(N)$ is the set of children of N.

We apply A* search based on the priority of a path assigned using the following function.

$$f(N) = g(N) \times h(N)$$

Unlike traditional A*, here the path costs are obtained through as products rather than sum. Moreover our aim is the maximize $f(N)$ rather than minimizing it. However, note that we can carry out the computations in the logarithmic domain, such that

$$f^*(N) = \log(g(N)) + \log(h(N))$$

Moreover, since all probability values are less than 1, the logarithms are less than 0. Thus, we can take the absolute values while defining all the aforementioned functions. This in turn transforms the maximization problem into a minimization problem. It is easy to see that the heuristic function defined here is an over-estimate.

6. CONCLUSION

This paper only deals with the proposed theory for obtaining transliterations at word level. Once implemented, it will be possible to identify further parameters which can be used as part of the heuristics to generate better results. Further, developing a system for generating transliterations at sentence level is a natural progression of this work.

7. REFERENCES

[1] Chopde, A. *ITRANS (version 5.30),* 2001, http://www.aczoom.com/itrans/

[2] Bilac, S., and Tanaka, H., A Hybrid Back-Transliteration System for Japanese*, Proceedings of the 20th International Conference on Computational Linguistics : COLING*. pp.597 – 603, 2004

[3] Chen, H.H., and Lin, W.H., Backward Machine Transliteration By Learning Phonetic Similarity*,* 6th *Conference on Natural language Learning*, 2002.

[4] Bandyopadhyay, S., Ekbal, A. and Naskar, S., A Modified joint Source Channel Model for Transliteration*, Coling ACL*, 2006

[5] Al-Onaizan, Y., and Knight, K. Machine Transliteration Of Names In Arabic Text*, ACL Workshop on Computational Approaches to Semitic Languages*

[6] Bilac, S., and Tanaka, H. Improving back Transliteration by Combining Information Sources*, IJCNLP*, 2004.

[7] Mukherjee, A., Chakraborty, S., Choudhury, M., Lahiri, A., Dey, S., Basu, A. Shruti-An Embedded Text to Speech System for Indian Languages*. IEE Proceeding s on Software Engg*, 2006.

[8] Fredkin, E. Trie Memory. *Communications of the ACM*, 3(9):490-499, 1960.

[9] Och, F.J., Zens,R., Ney, H. *Efficient Search for Interactive Statistical machine Translation*, 2003.

Title

The structure of Nepali Grammar. Bal Krishna Bal, Madan Puraskar Pustakalaya, Nepal.
bal@mpp.org.np

Abstract

This paper is an attempt to provide the basic insight to the structure of the Nepali Grammar. It takes a tour of the writing system, the parts of speech, the phrase and clausal structure and finally ending in the sentential structure of the language. Research on the grammar of the Nepali language, which is highly inflectional and derivational depending upon the wide range of grammatical aspects of the language, like tense, gender, pronouns etc., is but incomplete without taking into consideration the special characteristics of the Nepali Grammar. So wherever possible and deemed necessary, illustrations are also provided. In the discussion and the conclusion section, the document also presents a brief overview of the design and implementation of the Nepali Grammar Checker, to be developed as part of the Natural Language Processing Applications Development under the PAN Localization Project, Madan Puraskar Pustakalaya, Nepal. The findings of this study are believed to be an invaluable resource and base document for the development of the Nepali Grammar Checker.

Introduction

The formulation of the first Nepali Grammar dates back to the history, some eighty years back with the "Gorkha Bhasha Chandrika Vyakaran", found to have written long before the actual study of the Nepali language started(Adhikari Hemang Raj, 2005). The Nepali language and consequently the grammar has evolved a lot in this period. Numerous books and writings on the Nepali Grammar have come out in the mean time. Inconsistencies and debate in opinion on several grammatical issues, too, have continued to exist. Nevertheless, there are also aspects whereby the Grammarians have a common meeting point.

Research methodology and objectives

The research methodology devised for the given research work is basically the qualitative approach. The secondary data, i.e. the information available on different sources about the Nepali Grammar has been compiled and analyzed. Besides, other primary data collecting methods and mechanisms like active brain storming sessions, consultations with the experts also were exercised. The findings of this research work do not in any sense capture all the aspects of the Nepali Grammar structure. Furthermore, the findings of the study presented might be subjected to changes and corrections as well, as newer concepts and ideologes emerge. The primary objectives of this research work is to initiate a base document for further research.

Results

The results section summarizes the basic structure of the Nepali Grammar. This includes the writing system, form classes (lexicon), phrase structure and clause analysis and a brief overview of the sentential structure of the Nepali langauge. Peculiar cases and characteristics of the Nepali language and Grammar are also noted.

Writing System of Nepali

Nepali is written in the Devanagari script. Although the statistics vary, basically the Nepali

language has 11 vowels and 33 consonants. Debate prevails on whether to include the alphabets, which exist in pronunciation but not in writing and vice versa in the list of vowels and consonants. The pronunciation closely resembles the writing system and hence the script is highly phonetic. The script is written from left to right with an additional line on top of each word known as “dika” in Nepali. Without the dika, a word is considered grammatically incomplete although it can be read. There is no provision of the capital and small letters in the script. The alphabets are written in two separate groups, namely the vowels and the consonants as shown in the table below.

Table 1. Alphabets of the Nepali Language

Vowels	अ,आ,इ,ई,उ,ऊ,ऋ,ए,ऐ,ओ,औ
Consonants	क,ख,ग,घ,ङ,च,छ,ज,झ,ञ,ट,ठ,ड,ढ,ण,त,थ,द,ध,न,प,फ,ब,भ,म,य,र,ल,व,श,ष,स,ह

The three alphabets क्ष,त्र,ज्ञ are regarded as special clusters or conjuncts and hence form as a combination of one or more consonants and special symbols. We talk about them a bit later.

In addition to the alphabets mentioned above, the following signs and symbols exist in written Nepali as shown in the table below.

Table 2. Additional symbols in the Nepali language

Candrabindu	ँ
Anusvar or cirabindu	ं
Vowel signs	ा,ि,ी,ु,ू,ृ,े,ै,ो,ौ
Visarga	:
Viram or halanta	्

The vowel signs ा,ि,ी,ु,ू,ृ,े,ै,ो,ौ correspond to the vowels आ,इ,ई,उ,ऊ,ऋ,ए,ऐ,ओ,औ respectively. The alphabets categorized under the vowels are often called the free form of vowels whereas the vowel signs are called the conjunct vowels. The text below illustrates the order of the writing system of some of the vowel signs.

The vowel sign ि should appear before the consonant, which however is written first and then preceded by the vowel sign in normal written practice, बि=ि before the consonant ब.
The vowel sign ी follows the consonant, सी=ी after the consonant स.

The vowel signs ु and ू are written at the foot of the consonant, लु =ल+ ु, नू=न+ू
When joined to र , the vowels - and ू are written as रु and रू.

The three special clusters क्ष,त्र,ज्ञ are formed by the combination of the other consonants with the viram or the halanta playing a significant role in the combination as shown below:

क्ष=क+्+ष
त्र=त+्+र
ज्ञ=ज+्+ञ

Form classes (Lexicon)

The Nepali Grammar consists of both the inflected and the uninflected forms, sometimes also known as the open and closed classes respectively. These constitute the parts of speech of the Nepali Grammar. The open class includes noun, adjective, verb and adverb whereas pronoun, coordinating conjunction, postposition, interjection, vocative and nuance particle come under the closed class. In addition to the two form classes mentioned above, the Nepali Grammar has yet another class named as the substitute form class. The major substitute forms in Nepali are the K-forms, or interrogative questions, the J-forms, or subordinators and the D-forms, or demonstratives.

Nominal structure

Nominal structures in Nepali include the common-noun phrase, proper-noun phrase, pronoun phrase and dependent nominals functioning as modifiers in large nominals.

Verbal forms

The nonfinite verbal forms are:

i) infinitives marked by the infinitive suffix -na or -nu (jaana or jaanu 'to go');
ii) participles marked by the suffixes -eko, -ne, -dai,-tai,-yera,-i,-ikana(gareko - 'done', garne -'doing**',** gardai – 'doing', garikana-'having done');
iii) conditionals marked by the suffix -ye(gaye - 'if go', khaaye- 'if eat', gare- 'if do')

Verb conjugation types

The verb stems in Nepali are grouped, into three types:

i) 1'st conjugation- verbs with bases ending in consonants. For eg., gara- 'do', basa -'sit', dagura -'run'.
ii) 2'nd conjugation- verbs with bases ending in the vowels ii and aa, with a single exception of jaa-'go'. For eg., di-give, li-take, khaa-eat, birsi-forget.
iii) 3'rd conjugation – verbs with bases ending in the vowels: aau, a,u,and aa in the single case of jaa -'go'.

Sentential structure of Nepali

The Nepali sentences follow the Subject, Object, Verb pattern as opposed to English which follows the Subject, Verb, Object pattern.

For eg.,

Sentence type	English	Nepali
Declarative	I eat rice. (Subject, Verb,Object)	Ma bhaat khaanchhu. (Subject, object, verb)
Interrogative	Do you eat rice?(Subject, Verb,Object)	Ke timi bhaat khanchhou?(Subject, object, verb)
Imperative	You go home.(Subject, Verb,Object)	Timi ghara jaaun.(Subject, object, verb)

Discussion

Keeping in view the high degree of derivations and inflections as well as a different sentential structure (Subject, Object, Verb) of the Nepali Language, it requires a different parsing engine as well as a morpho-syntactic analyzer for the development of natural language processing applications in Nepali like the Spell Checker, Grammar Checker, Machine Translation System etc. An in-depth and an analytical research of the prerequisites of the Nepali computational grammar is hence the need for today. These prerequisites refer to both linguistic resources and natural language processing tools.

Recent updates in the research and development of the Nepali Grammar Checker include the conceptualization of the general architecture of the system. In general, the Grammar Checker aims to check the grammatical errors such as nominal and verbal agreement, parts of speech inflections and whether the SOV pattern is observed or not.

Conclusion

The finding of this preliminary research work do not in any sense capture all the aspects of the Nepali Grammar structure. Furthermore, the findings of the study might be subjected to changes as newer concepts and ideologies emerge. However, this research work can serve as a strong base document for further research. Besides, the results of the grammar checker research and development is sure to serve a milestone for the computational linguistic works for the Nepali Language. The specific modules to be developed for the system, viz., the Stemmer Module, POS Tagger Module, Chunker and Parser Module, Grammatical Relation Finder Module etc. are all being developed for the first time for the Nepali language. The development of the modules should open doors for further research works in the Nepali language and Grammar.

Acknowledgement

This research work is supported by the International Research and Development Center (IDRC), Canada under its PAN Localization Project, Nepal Component.

References

1)http://www.thdl.org/education/nepali/
2)http://www.omniglot.com/writing/nepali.htm
3)Teach yourself Nepali by Dr. Michael Hutt and Professor Abhi Subedi (Hodder &Stoughton, first published in 1999)
4)Basic Course in Spoken Nepali by Tika Bahadur Karki and Chij Kumar Shrestha (Kathmandu, Multiple editions)
5) A Course in Nepali by David Matthews. School of Oriental and African Studies, University of London, 1992.
6) A Descriptive Grammar of Nepali and an Analyzed Corpus by Jayaraj Acharya. Geogrtown University Press,1991.
7) CoGrOO: a Brazillian-Portuguese Grammar Checker based on the CETENFOLHA Corpus. Jorge Kinoshita, Lais do Nascimento Salvador, Carlos Eduardo Dantas de Menezes. Universidade da Sao Paulo (USP), Escola Politecnica, Sao-Paulo-SP-Brasil, Unifacs – Universidade de Salvador, Nuperc, Salvador-Bahla-Brasil, Centro Universitario SENAC, Sao Paulo, SP-Brasil.

Email Answering Assistant for Contact Centers

Rahul Malik , L. Venkata Subramaniam+ and Saroj Kaushik
Dept. of Computer Science and Engineering,
Indian Institute of Technology, Delhi,
Hauz Khas, New Delhi, India.
{csd02442,saroj}@cse.iitd.ernet.in
+IBM India Research Lab,
Block I, IIT-Delhi, New Delhi, India
lvsubram@in.ibm.com

Abstract

A contact centre is a centralized office used for the purpose of receiving and transmitting a large volume of customer care requests. Now a days, customer care in technical domain is mostly based on e-mail and replying to so many emails is time consuming. We present a technique to automatically reply customer e-mails by selecting the appropriate response template. The system can have great impact in the contact centers. The system has been evaluated and it achieves a good performance.

1 Introduction

A contact centre is a centralized office used for the purpose of receiving and transmitting a large volume of customer care requests. Most major businesses use contact centers to interact with their customers. Examples include utility companies, mail order catalogue firms, and customer support for computer hardware and software. Some businesses even service internal functions through contact centers. Examples of this include help desks and sales support.

The queries asked in one domain is fixed and customers usually ask from a standard set of queries only. The contact centre agents are typically provided many templates that cover different queries asked. When a query email comes, it is first triaged and send to the appropriate agent for response. The agent selects the appropriate template and fills it to compose the reply. Selection of the template is a time consuming task as agent has to search from a lot of templates to select the correct one.

Please provide the current status of the rebate reimbursement for my phone purchases.
I understand your concern regarding the mail-in rebate. For mail-in rebate reimbursement, please allow 8-14 weeks to receive it.
I understand your concern regarding the mail-in rebate. For mail-in rebate reimbursement, please allow <replace this> (***weeks or months***) </Replace this> to receive it.

Figure 1: A Typical Query, Response/Template Pair

The e-mails are in unstructured text and automatic extraction of relevant portion of e-mail that require a response is a difficult task. In this paper, we propose a system to automatically answer customer e-mails by:

1. Extracting, both from customer query Q and response mails R, the set of queries q_i and the set of responses r_j that decompose respectively the asked mail and the response mail.
2. Matching each query q_i with its relevant response r_j.
3. Finally, new questions are answered by comparing them to previously studied questions.

2 Related Work

Little work has been done in the field of contact centres emails. (Nenkova and Bagga, 2003), (Busemann et al., 2000) learn a classifier based on existing emails using features such as words, their parts of speech, etc. When new queries come in they are automatically routed to the correct agent. Not much work has been done on automatically answering the email queries from customers. In

(Scheffer, 2004) a classifier is learnt to map questions to a set of answer templates. Our work in this paper describes methods to automatically answer customer queries.

Extracting words and phrases that contain important information from a document is called Key phrase extraction. Key phrase extraction based on learning from a tagged corpus has been widely explored (Frank et al., 1999). (Turney, 1999) describes a system for key phrase extraction, GenEx, based on a set of parameterized heuristic rules that are fine-tuned using a genetic algorithm. (Frank et al., 1999) use a set of training documents and extract key phrases of length upto 3 words to generate a naive Bayes model. The model is used to find key phrases in a new document. However, key phrase extraction technique has been mainly applied in document summarization and topic search. We mention this body of work in the context of this paper because by extracting key phrases from an email, we identify the key questions and responses.

Text similarity has been used in Information retrieval to determine documents similar to a query. Typically, similarity between two text segments is measured based on the number of similar lexical units that occur in both text segments (Salton and Lisk, 1971). However, lexical matching methods fail to take into account the semantic similarity of words. In (Wu and Palmer, 1994) similarity of words is measured based on the depth of the two concepts relative to each other in WordNet [1]. In this paper we need to identify similar questions. Also we need to match a question to its answer.

3 Proposed Algorithm

Here, we describe the approach taken by us in building the system.

3.1 E-mail triaging

In a contact center, there are different agents who look after different aspects. So, the emails are manually triaged and are forwarded to the right agent who responds to them. We replace this step with automatic triaging. We use clustering to first identify the different classes and then learn a classifier based on these classes. We classified the emails from the larger pool into equal number of query and response clusters using text clustering by repeated bisection using cosine similarity. An SVM classifier was then learnt on the classes created by the clustering. This classifier is then used to triage the emails.

3.2 Key-phrase Identification

Key phrase identification is an important step in identifying the question and answers as they can be identified by the presence of key-phrases in them. Key phrase extraction is a classification task where each potential phrase could be a key phrase or not. First of all, candidate phrases are identified. The following rules are applied to identify the candidate phrases. All the unigrams, bigrams and trigrams are identified that are continuous. Also, the candidate phrases cannot begin or end with a stopword. We use a 452 word list of stopwords. The identified phrases are passed through Porter Stemmer[2] to obtain the root. The next step is to determine the feature vector for the training and testing phases. The following features are used: $TF * IDF$, a measure of phrase's frequency in a document compared to its rarity in general use; whether $proper\ noun$ is there in the phrase or not; $first\ occurence$, which is the distance into the document of the phrase's first occurence; and $num\ of\ words$, which is simply the word count in the phrase.

For training phase, the key phrases are marked in training query and response emails. They are made to generate a model which then is used in the prediction. We use Naive Bayes as the machine learning scheme. Once the model is learned, it is used to extract the key phrases from the testing emails.

3.3 Identification of Question and Answers

The questions and answers are identified by the presence of key phrases in them. If a key phrase occurs in multiple sentences in the document, then the sentence which has maximum number of key phrases is selected. In case of a tie, the first occurence is chosen. In this manner, we identify the question and answers in emails.

3.4 Mapping Questions to Answers

Once the questions and responses have been identified we need to map each question to its corresponding response. To accomplish this mapping we first partition each extracted sentence into its

[1] http://wordnet.princeton.edu/

[2] http://www.tartarus.org/ martin/PorterStemmer

list of tokens, removed the stop words and the remaining words are passed through a stemmer (using Porter's stemming algorithm) to get the root form of every word. The tokens include nouns, verbs, adjectives and adverbs. In addition, we also keep the cardinals since the numbers also play an important role in understanding the text. We then form a matrix in the following manner. The rows of the matrix are the tokens from one sentence and the columns are the tokens from the second sentence. Each entry in the matrix $Sim(s,t)$ denotes the similarity as it has been obtained in the following manner for that pair. Also, if a word s or t does not exist in the dictionary, then we use the edit distance similarity between the two words.

The similarity between two concepts is given as (Jiang and Conrath, 1997):

$$C_Sim(s,t) = \frac{1}{IC(s) + IC(t) - 2 \times IC(LCS)}$$

where IC is defined as:
$IC(c) = -logP(c)$
and $P(c)$ is the probability of encountering an instance of concept c in a large corpus. Also, LCS is the least common subsumer of the two concepts. We are using is Wordnet.

The similarity between two sentences is determined as follows :

$$SS(S_i, S_j) = \frac{\sum_{s \in S_i} MS(s, S_j) + \sum_{s \in S_j} MS(s, S_i)}{|S_i| + |S_j|}$$

where, $MS(s, S_j)$ is the word in S_j that has the highest semantic similarity to the word s in S_i.

In addition, we used a heuristic that if a question is asked in the beginning, then the chances that its response would also be in the beginning are more.

So, the expression for score becomes:

$$score(q,r) = SS(q,r) \times (1 - |\frac{pos(q)}{N} - \frac{pos(r)}{M}|)$$

where, $pos(q)$ = position of the Question in the set of Questions of that query-email, N = number of questions in that query-email, M = number of answers in that response-email

Each answer is then mapped to a template. This is done by simply matching the answer with sentences in the templates.

Multiple questions can match the same template because different customers may ask the same question in different ways. So, we prune the set of questions by removing questions that have a very high similarity score between them.

3.5 Answering new Questions

When we get a new query, we first triage it using the SVM classifier that was described in Section 3.1. Next, we identify the questions in it using the procedure described in Section 3.3. Each of these questions now needs to be answered. For a new question that comes in, we need to determine its similarity to a question we have seen earlier and for which we know the template. The new question is mapped to the template for the existing question to which it is similar. Using the above sentence similarity criterion, we compare the new question with the questions seen earlier and return it's template.

4 Evaluation

We evaluate the system on Pine-Info discussion list web archive[3]. It contains emails of users reporting problems and responses from other users offering solutions and advice. The questions that users ask are about problems they face in using pine. Other users offer solutions and advice to these problems.

The Pine-Info dataset is arranged in the form of threads in which users ask questions and replies are made to them. This forms a thread of discussion on a topic. We choose the first email of the thread as query email as it contains the questions asked and the second email as the response as it contains responses to that email. It may not contain answers to all the questions asked as they may be answered in subsequent mails of the thread. We randomly picked up a total of 30 query-response pairs from Pine-Info. The question sentences and answer sentences in these were marked along with the mappings between them.

On the average a query email contains 1.43 questions and the first response email contains 1.3 answers and there are 1.2 question-answer pairs. We show a query and response pair from Pine-Info in Figure 2. The actual question and answer that have been marked by hand are shown in bold. In the example shown one question has been marked in the query email and one answer is marked in the response email.

In pine, there are no templates. So, we are effectively checking that whether we are able to map the manually extracted questions and answers correctly. For evaluation purposes, we use two criteria. In the first case we say the system is correct

[3] http://www.washington.edu/pine/pine-info

When printing with PINE I always lose a character or two on the left side margin. **How can I get PINE to print four or five spaces to the margin?** Printer works fine with all other applications.

Use the utility 'a2ps' to print messages with a nice margin. See links for more information.

Figure 2: Pine-Info Query, Response Pair

only if it generates the exact answer as the agent. In the second case we allow fractional correctness. That is if the query contains two questions and the system response matched the agent response in one of them then the system is 50% correct.

As already mentioned we are looking for an answer in the first response to the query. Hence, all questions in the query may not get addressed and it may not be possible to get all the mappings. In Table 1 we show the numbers obtained. Out of a total 43 questions only 36 have been mapped to answers in the manual annotation process. Using the method presented, we are able to find 28 of these maps correctly.

Table 1: Results on Pine-Info Dataset for Question-Answer Mapping

total mails	total qns	total ans	actual maps	correct maps	% correct maps
30	43	39	36	28	77.78%

Without partial correctness, the system achieves 77.78% correctness. When we consider partial correctness also, the it increases upto 84.3%.

We also tested the system real life query-response emails. We used 1320 email pairs out of which 920 were used for training the system and 400 were used for testing. We had 570 sample templates. Without partial correctness, the classification accuracy achieved was 79% and when we consider partial correctness, then it increases to 85%.

5 Conclusion and Future Work

In this paper, we have presented a technique of automatically composing the response email of a query mail in a contact centre. In the training phase, the system first extracts relevant questions and responses from the emails and matches the question to its correct response. When a new question comes, it triages it to correct class and matches its questions to existing questions in the pool. It then composes the response from existing templates.

The given system can improve the efficiency of contact centers, where the communication is largely email based and the emails are in unstructured text. The system has been tested thoroughly and it performs well on both Pine-Info dataset and real life customer query-response emails.

In future, we plan to improve our system to handle questions for which there are no predefined templates. We would also like to fill some of the details in the templates so that the agent's work can be reduced. Also, we would like to add some semantic information while extracting questions and answers to improve the efficiency.

References

Stephan Busemann, Sven Schmeier, and Roman G. Arens. 2000. Message classification in the call center. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 158–165, Seattle, Washington, April 29-May 4.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence,*, pages 668–673, San Fransisco, CA.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*.

Ani Nenkova and Amit Bagga. 2003. Email classification for contact centers. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 789–792, Melbourne, FL, USA, March.

G. Salton and M. E. Lisk. 1971. *Computer Evaluation of Indexing and Text Processing*. Prentice Hall, Englewood Cliffs, NJ.

Tobias Scheffer. 2004. Email answering assistance by semi-supervised text classification. *Journal of Knowledge and Process Management*, 13(2):100–107.

P. Turney. 1999. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council, Institute for Information Technology.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, June 27-30.

Extracting Structural Rules for Matching Questions to Answers

Shen Song and Yu-N Cheah
School of Computer Sciences
Universiti Sains Malaysia, 11800 USM Penang
summerysmile@hotmail.com and yncheah@cs.usm.my

ABSTRACT

Rule-based Question Answering(QA) systems require a comprehensive set of rules to guide its search for the right answer. Presently, many rules for QA system are derived manually. This paper presents a question answering methodology as well as proposes an automatic rule extraction methodology to obtain sufficient rules to guide the matching process between the question and potential answers in the repository.

1. Introduction

Matching is an essential part of a QA system. It decides whether the final answer is reasonable and accurate. In the past, manually extracted rules were popularly employed to support the matching function of the QA system. However, it is difficult to find a certain number of rules to suit various kinds of question-answer structures [1].

In this paper, we introduce a proposed QA methodology as well as a simple methodology for automatic rule extraction to obtain rules for matching questions to the right answers based on clustering.

2. Our QA Methodology

The overview of our QA approach is as shown in Figure 1. At the heart of our methodology lies a Response Generator that executes the QA methodology.

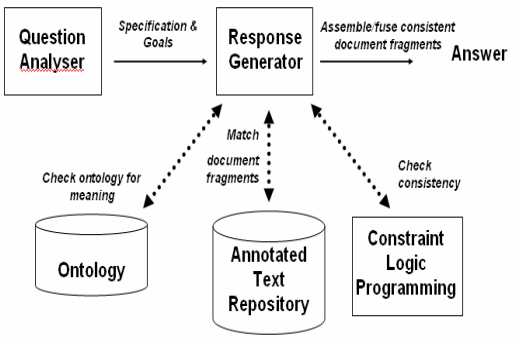


Figure 1: Overview of our QA approach

In our QA system, answers are obtained from semantically annotated text repositories. These text documents are tagged for parts-of-speech (POS). A question analysis component is also included to identify keywords and question goals (via Wh term analysis).

Our QA methodology consists of 4 steps: (1) ontology-based question and repository understanding, (2) matching, (3) consistency checking, and (4) answer assembly.

2.1 Ontology-based question and repository understanding

The domain ontology is a basic but important component in our methodology. We use the ontology to understand the words or phrases of the tagged (or analysed) question and the tagged document fragments stored in the repository. From the question point of view, the ontology facilitates query formulation and expansion. Given a question, the question analyser will analyse and tag the question with the relevant POS as well as details from the ontology. The question's type and syntax are then determined by the analyser.

2.2 Matching

After obtaining the ontology-based understanding of the question (question's goal, keywords, etc.) and repository content, the response generator searches for relevant document fragments in the repository of semantically annotated documents. The response generator employs a variety of matching rules to select the candidate responses. We have presently identified two matching rules:

1. *Structural (or syntactic) matching rules*: This is facilitated by the POS tagging of the question and document fragments in the repository. The use of structural matching rules is based on the assumption that the answer may have a similar sentence structure to the question [2].
2. *Wh analysis rules*: This is based on the idea that certain questions are answered in a particular way. For example, 'how'

questions may typically contain *preposition-verb* structures in potential answers; or 'why' questions may typically contain the word 'because' in the answers.

We later explore the possibility of automatically extracting structural rules for this purpose.

2.3 Consistency checking

Usually, there are more than one document fragments that match the question. However, not all may be consistent with each other, i.e. they may have minor conflicting information. So, we explore the use of constraints to maintain the answer's consistency. After consistency checking, we would then have a list of consistent matching document fragments.

2.4 Answer assembly

In this step, we analyse the matching fragments to eliminate redundant information and combine the remaining document fragments. Then, we compare the question words with the matching answer and check the quantification, tense and negation relationship. The semantic structure between the question and the answer is also checked to make sure that the question is explained sufficiently in the answer.

3. Automated Rule Extraction for Question Answering

For the matching phase of our QA approach, we have previously identified structural matching rules and Wh analysis rules for matching questions to their potential answers.

However, in the past, these rules are induced manually by analysing a limited number of common question-answer structures. They are not sufficient to solve a wider range of question-answer matching problems. We therefore propose a methodology to automatically induce rules to match questions and answers. Here, we focus on extracting structural matching rules only.

Our proposed methodology consists of three phases: (1) compilation of question-answer pairs, (2) analysis of question-answer pairs, and (3) rule extraction via clustering.

3.1 Compilation of question-answer pairs

We need a mass of question-answer pairs to support our rule extraction. So, collecting question-answer pairs from the Internet is a good choice for us due to the redundancy of information on the Internet. Alternatively, sample answers for comprehension tests may also be used. As an example, let us suppose a sample of the question-answer pairs obtained is as shown in Table 1.

Table 1: Question-answer pairs

No	Question	Answer
1	How do I get from Kuala Lumpur to Penang?	To travel from Kuala Lumpur to Penang, you can take a bus.
2	Where is Kuala Lumpur?	Kuala Lumpur is in Malaysia.
3	How can I travel to Kuala Lumpur from Penang?	You can travel to Kuala Lumpur from Penang by bus.
4	Where is the location of Penang?	Penang is located north of Peninsular Malaysia.
5	What can I do to get to Kuala Lumpur from Penang?	You can get to Kuala Lumpur from Penang by bus.
6	What can I do in Langkawi?	You can go scuba diving in Langkawi.

3.2 Analysis of question-answer pairs

The question-answer pairs are analysed for their structure and are tagged accordingly. The syntactic (analysed) notations of the question-answer pairs form the dataset for our rule extraction. Based on Table 1, we produce our dataset as shown in Table 2.

Table 2: Analysed question-answer pairs

No	Question	Answer
1	How vb pron vb prep location prep location	Prep vb prep location prep location, pron vb vb art vehicle
2	Where vb location?	Location vb prep location.
3	How vb pron vb prep location prep location?	Pron vb vb prep location prep location prep vehicle.
4	Where vb art n prep location?	Location vb vb adj prep adj location.
5	What vb pron vb prep vb prep location prep location?	Pron vb vb prep location prep location prep vehicle.
6	What vb pron vb prep location?	Pron vb vb adj n prep location

3.3 Rule extraction via clustering

We propose three ways of clustering the analysed dataset for rule extraction [3]:

1. Cluster only the question part
2. Cluster only the answer part
3. Cluster both the question and answer parts together.

In this paper, we describe the method of clustering the question part only.

Firstly, we cluster the question part of our analysed dataset. For this purpose, we check for the similarity in the structure of the question part. From Table 2, let us assume three clusters are obtained: Cluster A consists of rows number 1, 3 and 5; Cluster B consists of rows number 2 and 4; and Cluster C consists of row number 6 only. This is because the question structures in each cluster are deemed to be similar enough (see Table 3). Each cluster would then need to have a representative question structure. For our purpose, the most popular question structure within each cluster would be selected. For example, for Cluster A, the representative question structure would be *How vb pron vb prep location prep location.*

Next, within each cluster, we then analyse their respective answer parts and choose the most popular structure. For example, in Cluster A, rows number 3 and 5 have similar answer structures and is therefore the most popular answer structure among the three rows in Cluster A. We therefore conclude that the question structure for Cluster A would result in answers that have the structure *Pron vb vb prep location prep location prep vehicle*. The rule that can be extracted from this may be in the form:

IF *How vb pron vb prep location prep location*
THEN *Pron vb vb prep location prep location prep vehicle*

4. Using the Extracted Rules: An Example

Following the extraction of rules, the matching process can then be carried out by the Response Generator. The rules basically guide the matching process to find an answer in the repository that is able to answer a given question, at least from a structural point of view (semantic details would be resolved via the ontology). Here, the issue of similarity between the rules' specification and the structure present in the question and answer needs to be addressed [4].

Table 3: Clustered question-answer pairs

Cluster	No	Question	Question Structure	Answer	Answer Structure
A	1	How do I get from Kuala Lumpur to Penang?	How vb pron vb prep location prep location?	To travel from Kuala Lumpur to Penang, you can take a bus.	Prep vb prep location prep location, pn vb vb art vehicle
	3	How can I travel to Kuala Lumpur from Penang?	How vb pron vb prep location prep location?	You can travel to Kuala Lumpur from Penang by bus.	Pron vb vb prep location prep location prep vehicle.
	5	What can I do to get to Kuala Lumpur from Penang?	What vb pron vb prep vb prep location prep location?	You can get to Kuala Lumpur from Penang by bus.	Pron vb vb prep location prep location prep vehicle.
B	2	Where is Kuala Lumpur?	Where vb location?	Kuala Lumpur is in Malaysia.	Location vb prep location.
	4	Where is the location of Penang?	Where vb art n prep location?	Penang is located north of Peninsular Malaysia.	Location vb vb adj prep adj location.
C	6	What can I do in Langkawi?	What vb pron vb prep location?	You can go scuba diving in Langkawi.	Pn vb vb adj n prep location

As an example of a matching process, let us assume we would like to answer the question, "How do I get from Kuala Lumpur to Penang?"

Firstly, the question will be analysed and converted into the form as follow: *How vb pron vb prep location prep location?*

Based on the rule extracted above, we know this kind of structure belongs to Cluster A and the corresponding answer's structure should be: *Pron vb vb prep location prep location prep vehicle.*

Finally, from the repository, we select answers with this answer structure. Likely answers are therefore: "You can travel to Kuala Lumpur from Penang by bus" or "You can get to Kuala Lumpur from Penang by bus".

5. Concluding Remarks

In this paper, we introduced our proposed QA methodology and a methodology for automatic rule extraction to obtain matching rules based on clustering. Our research is still in the initial stages. We need to improve the rule extraction methodology by (1) improving the analysis and tagging of the question-answer pairs; (2) designing an efficient algorithm to cluster the dataset; and (3) developing a better method to aggregate similar question or answer structures into a single representative structure.

6. References

[1] Riloff, E., Thelen, M., A Rule-based Question Answering System for Reading Comprehension Tests, ANLP/NAACL-2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems, 2000.

[2] Li, W., Srihari, R.K., Li, X., Srikanth, M., Zhang, X., Niu, C., Extracting Exact Answers to Questions Based on Structural Links, Proceeding of the 2002 Conference on Multilingual Summarization and Question Answering, Taipei, Taiwan, 2002.

[3] Lin, D., Pantel, P., Discovery of Inference Rules for Question Answering, Natural Language Engineering, 7(4), 2001, pp. 343-360.

[4] Jeon, J., Croft, W.B., Lee, J.H., Finding Semantically Similar Questions Based On Their Answers, Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), Salvador, Brazil, 2005, pp. 617-618.

"Who" Question Analysis

Rapepun Piriyakul amd Asanee Kawtrakul

Department of Computer Engineering
Kasetsart University
Bangkok , Thailand
rapepunnight@yahoo.com ak@.ku.ac.th

Abstract

The purpose of this research is to automatically analysis the "Who" question for tracking the expert's knowledge. There are two main problems involved in the "Who" question analysis. The first problem is to identify the question type which relies on the cue ambiguity and the syntactic ambiguity. The second one is to identify the question focus which based on syntactic and semantic ambiguity. We propose mining features for the question identification, and the usage of focus rules for the focus identification. Furthermore, this "Who" question analysis shows the 80% precision and the 78% recall.

1. Introduction

People demand information as a response to their question about a certain fact. In the past, Information Retrieval was used to assist the people in retrieving information. The way the Information Retrieval works is that the system looks up for the frequencies of the essential words that the person is looking for over other databases of information. Information Retrieval, however, fails to be efficient at taking consideration of the desired context of the individual, which can be gained through the understanding of the true question asked by the individual. To allow this, 'Question Answering System' was introduced. Question Answering System is a type of Information Retrieval with the purpose of retrieving answers to the questions impose, done by applying techniques that will enabled the system to understand the natural language (http://.www.wikpedia.org).The Question Answering system (QA) consists of two sub systems. The first is question analysis and the second is the answering system. These two sub systems are significantly related. Since question analysis is the front end of QA system. Error analysis of an open domain QA systems found that 36.4 % of inaccurate answer came from the wrong question analysis [T. Solorio et , al 2005].

However, this paper only concerns of "Who" question of the Thai language because we can keep tracking about the expert's knowledge for solving problems. We confront many characteristics of Thai questions, such as the implicit question word, the question word can be placed at any position of the question text; the appearance of question word does not always make the question. After the question type identification, the next step of question analysis is focus analyzer which based on syntactic and semantic analysis. It is necessary to determine the question focus to gain the correct answer. To fulfill the complete question representation, we also propose to construct the Extended Feature Knowledge (EFK) to enhance the Answering System with the cooperative way.

In this paper, we have 6 sections. We begin with the introduction, and then we will discuss the problems in section 2, related works in section 3, the framework will be observed in section 4. Then we will evaluate in section 5 and plan our future study in section 6.

2. Crucial Problems

There are two main problems, the identification of "Who" question, and focus question.

2,1 Question identification

The objective of question identification is to assure that the question text is "Who" question and a true question. Since Thai question has no word marker "?" , we use set of cues to identify a question type "Who", for example: {"khrai", "dai", "a rai"..}. There are three problems of the question identification as the movement of a question cue, the question cue ambiguity and the syntactic problem.

2.1.1 Movement of question cue

A question cue can occur at any place in the interrogative sentence as shown in the following examples

a./*Khrai// khue//Elvis Pressley/*
Who was Elvis Pressley?
b. */Elvis Pressley/ /Khrai// khue/*
Who was Elvis Pressley?

Question *a* and *b* have the same meaning.

2.2.2 Question cue ambiguity

The presence of question cue, e.g. "khrai", does not always make the sentence a question. For example:

c. ***/Khrai/*** */pen/ /na-yok/ /rat-ta-montri/ /khong /pra- thet/ Thai/*
(Who is the prime minister of Thailand ?)
d. ***/Khrai/*** */pen// na-yok/ /rat-ta montri//khong/ / pra- thet// Thai// **ko//** kong// me// pun-ha/ .*
(Anyone who is a prime minister of Thailand will met the problem.)

The example *c* is a question whereas *d* is the narrative sentence as a result of the word **/ko/ (** /Ko/ is a conjunction). .

2.2.3 Syntactic problems

From the above example *a-d*, they do not specify what tense they are because the Thai language does not have verb derivation of specifying the tense. And, they also lack of the verb derivation of specifying the number. These syntactic problems cause the problem in Thai QA because the answer can be represented both individual and list of person. For exemples:

/Khrai// rien// NLP/
(Who is/are studying NLP?)
(*Who was/were studying NLP?)*

The answer can be the individual such as "A is studying NLP" or list of person such as "A, B, C and D are studying NLP" or the representation can be a group of person such as "The second year students are studying NLP".

2.2 Question Focus identification

To identify the question focus is an important to achieve the precise answer. There are many types of focus with respect to "Who" question i.e. Person's description, Organization's definition, Person or Organization's name, Person's properties. The following examples are shown the pattern of question.

e. */Khrai// sang //tuk// World Trade/*
(Who built the World Trade building?)
f. */Khrai// khue//Elvis Pressley/*
(Who was Elvis Pressley?)

Question *e*, focus is the name of person or organization but question *c*, focus is the property of Elvis Pressley. The accomplishment to automatically identify focus is based on syntactic, semantic analysis and the world knowledge. The efficiency of Answering System is based on the empowering of question analysis.

3. Related Work

Most approaches to question answering system focus on how to select precise answer .[Luc 2002] was developed question analysis phase to determine the expected type of the answer to a particular question based on some extraction functions with parameters .For instance , the question focus from "*Which was radioactive substance was Eda Charlton injected in 1915 ?"* was *substance*. Machine learning techniques are being used to tackle the problem of question classification [Solorio et . al,. 2005]. [Hacioglu ,et. all., 2003] used the first step in statistical QC (Question Classification) to design a taxonomy of question types. One can distinguish among taxonomies of having flat and hierarchical structure, or taxonomies of having a small (10-30) and large number of categories (above 50) .YorkQA [Alfonseca et, al., 2001] took inspiration from the generic question answering algorithm presented in [Simmons 1973] which was similar to the basic algorithms used in the Question Answering systems built for TREC. The algorithm carries on three procedurals steps, the first was accumulating a database of semantic structures representing penitence meaning, the second was selecting a set of appears relevant to the question. Relevance was measured by the number of lexical concepts in common between the proposed answer and the question. [Alfonseca and Marco ,2001] extended some procedure ie : the question analyzer used pattern matching based on Wh –words and simple part–of-speech information combined with the use of semantic information provide by WordNet to determine question types. QA in Webclopedia [Hovy et.al., 2004] a system that used a CONTEXT parser to parse and analyze the question. To demonstrate the question analysis part of this system, they parse input question using CONTEXT to obtain a semantic representation of the question. The phases/words from syntactic analysis were assigned significance scores according to the frequency of their type in Webclopedia question corpus (a collection of 27,000 + questions and answers) secondarily by their length, and finally by their significance scores derived from word frequencies in the question corpus.

Our work is especially deep analysis of "Who" question. Our research is integrated from QA TREC but we are modified and extended some part to be applicable for Thai QA.

4. Framework for Question Analysis

Our work is based on the preprocessing of question text on word segment, POS, and NE Recognition. The classification of Wh question is based on the syntactic analysis of interrogative pronoun and the Wh words in table 1. Posterior Bay's probability is using to confirm the accuracy of the classification.

The set of cue in table 1 is used to be a first coarse classifier for "Who".

Table 1 The set of cue word

	Thai Word	Remark
Who , Whom	*/khrai//khue/ /khue//khrai/ //phu//dai// //tan//dai// //boog-khon//nai// /khon/dai/*	*/ khrai/ //tan//dai// //boog-khon//dai//* are common noun
Which	*/ khon//nai/*	Which one
What	/a-rai/	We must combine two word together /choe/a-rai/ What name

We use the posterior Bays' probability to determine the classification of "Wh" type.

$$\Pr(q\_type / Wh\_word) = \frac{\Pr(q\_type \cap Wh\_word)}{\Pr(Wh\_word)}$$

The posterior Bayes' probability for a question with given a cue "/khrai/" of our study domain is 0.7. The probability value is use to make decision on the first step of question classification.

After classifying the question, the non question text is pruning by a set of cues. These cues act as the guards to select only the true question for the next step.

Based on our observation, we found beneficial characteristic of Thai question as the following examples:

g. */Khrai// **ko**//dai/chuay/dua/*
(Any one can help me.)

h. */Khrai/ pen// na-yok/ /rat-ta montri//khong/ / pra- thet// Thai// **ko**// kong// me// pun-ha/*
(Anyone who is a prime minister of Thailand will met the problem.)

Statement *g* and *h* are not question by the determination of /ko/.

i. */ Na-yok/ /rat-ta-montri/ /khong/ /Thai//khon/ti/**laew**/khue/ /Khrai/*
(Who was the prevous priminister of Thai ?)

j. */ Na-yok/ /rat-ta-montri/ /khong/ /Thai//khon/ti/**laew**/**laew**//khue/ /Khrai/*
(Who were the prevous priministers of Thai ?)

With word /laew/, question *i* and *j* are signified to the singular- past for question *i* and plural -past for question *j*.

In case of question with verb is "is-a" then the focus is NE .The syntactic and semantic can not identify the type of focus so the system will be supported by the world knowledge to identify NE for Person or Organization. From *figure 1*,we can classify verbs on "Who" question into four groups and each group as show below

Group1 = {*/pen/* is a }
Group2= {*/sang/* built,*/kit-khon/* invent , */patana/* develop,*/tum/* do }
Group3 = {*/khue/* is a}
Group4 = {*/khao//kai/* be qualify,*/me/* has }

Verb "*/me/has,have*" in group4 must follow by a constraint word such as */me//sit-ti/* has right , */me/aum-nat /* has power or authority .

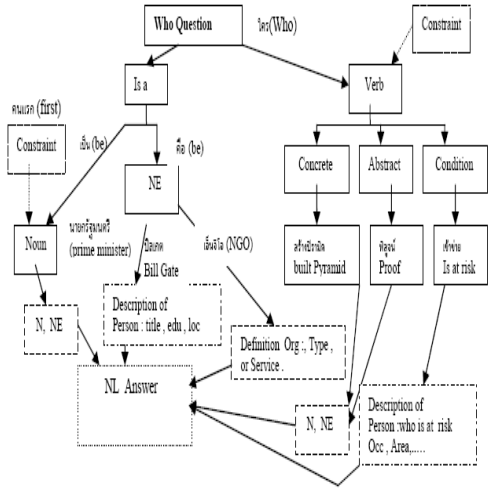


Figure 1 The patterns of Who question

From each verb group we can conclude to four rules.

Rule1 : *If the verb in group1 , then focus is "Person" or "Organization" and the answer is NE.*

If <IP */khrai/*><is a */pen/* > <NP> Then <Focus is NP> and <Answer = NE> where IP=interrogative pronoun

Rule2: *If the verb in group2+NP , then focus is NP and the answer is NE.*

If <IP */khrai/*><VP=V/*verb group3/+NP*> Then <Focus is NP> and <Answer = NE>

Rule3: *If the verb in group3 , then focus is "Person" or "Organization" and the answer is Description or Definition.*

If <IP */khrai/*><is a */khue/* > <NP> Then <Focus is NP> and <Answer = Description or Definition>

Rule 4 *If the verb in group4 , then focus is VP and the answer is list of properties.*

If <IP */khrai/*><VP=V/*verb group4/* + NP >Then <Focus is NP> and <Answer = list of properties>

To enhance the answering system for a precise and concise answer, we mine extended features to detect *lexical terms* such as *description* for Who (person) and *definition* for Who(organization). We examine the feature space of description properties as Gender, Spouse ,Location, Nationality, Occupation, Award , Education ,Position ,Expertise) from the sample of personal profile .To simplify , these features are represented by a set of feature = $(x_1,x_2,\ldots, x_9)$ where i=1,2,…,9 and x_1 is any feature on the feature space. The mining features are based on the proportion test with threshold 0.05.

Table 2 Show the experiment result

Feature	X1	X2	X3	X4	X5	X6	X7	X8	X9
Freq Occ	200	128	119	218	119	155	317	146	119
P	1	0.6	.05	0.1	.05	0.75	0.15	0.7	.05

Table2 shows the proportions of the 9 features. The results of mining features are Position, *Nationality*, *Occupation*, *Location*, *Expertise and Award*. These extended features are storing in the knowledge base to access from system. Figure 2 is the question analysis system. .

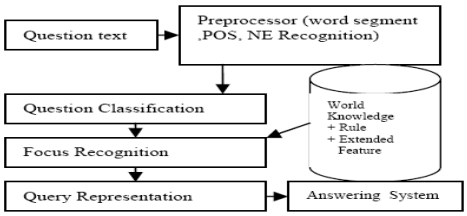


Figure 2 The Question Analysis System

The procedure of question analysis are

1.Input text" /*sun//thon//phu//khue//khrai/*"

2.Preprocessing

3. Identify question type and pruning

4. Identify focus by semantic analysis, world knowledge and Rule3,

5.Query representation = question type +focus+ list of extended features.

Figure 3 is a prototype of query representation

```
( Q focus NE   Answer => Description
      NE  [ Person : Description of NE]
            Class  X (Person) : Evaluation criteria
          {
            { subclass {Person :(Description :
                Name
                Title […] <Rank>
              - Location ,Experience,  Achievement,..
Publication  - Books . Research . Innovation }}}
```

5. Evaluation

The precision of our experiment to classify the question type by using question word with posterior Bayer's technique is 80 % and the recall is 78% .So far we use the process to solve the problems as the application of appropriate rules to individual problem .

The accuracy of question recognizer is 75 % by comparing QA pair (examine by expert).

6 Future Works

We would be collecting the features of other Wh question. We would also append the synset and update the Question Ontology, enhance question analysis by combining reasoning, constraint and suggestion for optimum size for answering system.

References

1. Alfonseca. Enrique ., Marco De Boni, José-Luis Jara-Valencia, Suresh Manandhar.,2001. A prototype Question Answering system using syntactic and semantic information for answer retrieval. *Proceedings of the TREC-9 Conference 2001*.

2. E.Hovy, L.Gerber, U. Hernjakob, C. Lin . 2004. Question Answering in Webclopedia . *Proceedings of the TREC-10 Conference 2002.*

3. Hacioglu, Kadri and Ward, Wayne., 2003. Question Classification with Support Vector Machines and Error Correcting Codes ., in the *Proceedings of NAACL/HLT-2003.*

4. Luc Plamondon and Leila Kosseim. , 2002. QUANTUM: *A Function-Based Question Answering System*. In Robin Cohen and Bruce Spencer (editors), Advances in Artificial Intelligence, 15th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2002, Calgary, Canada.

5.T.Solorio1,ManuelP´erez-Couti˜no1, Manuel Montes-y-G´omez1,2.,LuisVillase˜nor-Pineda1 and Aurelio L´opez-L´opez1.,2005.A Language Independent Method for Question Classification., .page 291.,*Computational Linguistics And Intelligent Text Processing: 6th International Conference. CICLing 2005, held in Mexico City.*

Mind Your Language: Some Information Retrieval and Natural Language Processing Issues in the Development of an Indonesian Digital Library

Stéphane Bressan
National University of Singapore
steph@nus.edu.sg

Mirna Adriani
Zainal A. Hasibuan
Bobby Nazief
University of Indonesia
{mirna,zhasibua,nazief}@cs.ui.ac.id

1. Introduction

In 1928, the vernacular Malay language was proclaimed by the Youth Congress, an Indonesian nationalist movement, the national language of Indonesia and renamed *"Bahasa Indonesia"*, or the Indonesian language. The Indonesian language is now the official language of the republic of Indonesia, the fourth most populated country in the world. Although several hundreds regional languages and dialects are used in the Republic, the Indonesian language is spoken by an estimated 228 million people, not counting an additional 20 million Malay speakers who can understand it. For a nation composed of several thousands islands and for its diasporas of students and professionals, the Internet and the applications it supports such as the World Wide Web, email, discussion groups, and digital libraries are essential media for cultural, economical, and social development. At the same time the development of the Internet and its applications can be either a threat to the survival of indigenous languages or an opportunity for their development. The choice between cultural diversity and linguistic uniformity is in our hands and the outcome depends on our capability to devise, design, and use tools and techniques for the processing of natural languages. Unfortunately natural language processing requires extensive expertise and large collections of reference data.

Linguistic is everything but a prescriptive science. The rules underlying a language and its usages come from observation. Furthermore the speakers continuously modify existing rules and internalize new rules under the influence of the socio-linguistic factors, the least one of which is not the penetration of foreign words. The Indonesian language is a particularly vivid example a living language in constant evolution. It includes vocabulary and constructions from a variety of other languages from Javanese to Arabic, English, and Dutch. It comprises an unusual variety of idioms ranging from a respected literary style to numerous regional dialects (e.g. Betawi) and slangs (e.g. Bahasa Gaul). Linguistic rules and data collections (dictionaries, grammars, etc.) are the foundation of computational linguistic and information retrieval. But their acquisition requires the convergence of significant amounts of effort and competence that smaller or economically challenged communities cannot afford. This compels semi-automatic or automatic and adaptive methods.

The project we are presenting in this paper is a collaboration between the National University of Singapore and the University of Indonesia. The research conducted in this project is concerned with the economical and therefore semi-automatic or automatic acquisition and processing of such linguistic information necessary for the development of other-than-English indigenous and multilingual information systems. The practical objective of is to provide a better access to the wealth of information and documents in the Indonesian language available on the World Wide Web and to technically sustain the development of an Indonesian digital library [25].

In this paper we present an overview of the issues we have met and addressed in the design and development of tools and techniques for the retrieval of information and the processing of text in the Indonesian language. We illustrate the need for adaptive methods by reporting the main results of four experiments in the identification of Indonesian documents, the stemming of Indonesian words, the tagging of part of speech, and the extraction of name-entities, respectively.

2. Identifying Indonesian Documents

The Indonesian Web, or the part of the World Wide Web containing documents primarily in the Indonesian language, is not an easily identifiable component. By the very nature of the Web itself, it is dynamic. Formally, using methods such as the one described in [14], or informally, one can safely estimate the size of the Indonesian Web to be several millions of documents. Web pages in Indonesian link to documents in English, Dutch, Arabic, or any other language. As we only wish to index Indonesian web pages, a language identification system that can tell whether a given document is written in Indonesian or not is needed.

Methods available for language identification [15] yield near perfect performance. However these methods require a training set of documents in the languages to be discriminated. This setting is unrealistic in the context of the web as one can neither know in advance nor predict the languages to be discriminated. We devised a method

[24] that can learn from a training set of documents in the language to be distinguished only. To put it in the Machine Learning terms, we devised an algorithm that learns from positive examples only. As its predecessor, our method is based on trigrams. The effectiveness of our method relies on the specificity of the trigram frequencies for a given language. The comparative performance evaluation shows a precision of 92% and for a recall close to 100%. Figure 2.1 illustrate the performance of the initial method after learning from iteratively larger sets of positive examples.

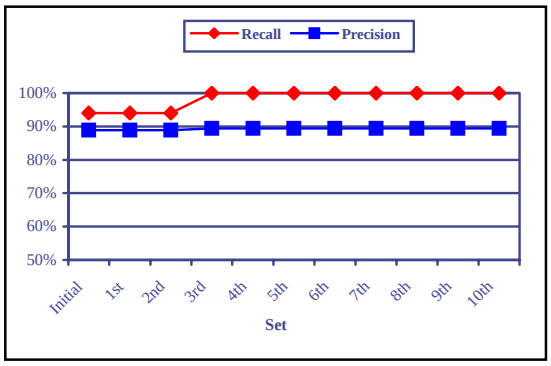


Figure 2.1: Language Identification Performance

Yet this performance is still lower than the one of the algorithms based on discriminating corpuses. To improve the initial performance and to make the solution adaptive to changes in the language and usage, we devised a continuously learning method that uses the documents labeled as Indonesian by the algorithm to further train the algorithm itself. The performance evaluation of this Continuous-Learning Language Distinction quickly converged toward total recall and precision for random samples from the Web. The method even performs well under harsh conditions: it has for instance been able to distinguish Indonesian documents from documents in morphologically similar languages such a Tagalog and even Malay at very respectable levels of precision.

3. Stemming

One of the basic tools for textual information indexing and retrieval is word stemmer. Yet effective stemming algorithms are difficult to devise as they require a sound and complete knowledge of the morphology of the language.

The Indonesian language is a morphologically rich language. There are around 35 standard affixes (prefixes, suffixes, circumfixes, and some infixes inherited from the Javanese language) (see [6]). Affixes can virtually be attached to any word and they can be iteratively combined. The wide use of affixes seems to have created a trend among Indonesian speakers to invent new affixes and affixation rules. This trend is discussed and documented in [23]. We refer to this set of affixes, which includes the standard set, as extended.

In [18], we proposed a morphology-based stemmer for the Indonesian language. An evaluation using inflectional words from an Indonesian Dictionary [23] has shown that the algorithm achieved over 90% correctness in identifying root words [7, 18, 21]. The use of the algorithm improved the retrieval effectiveness of Indonesian documents [18]. In comparison with this morphology-based stemmer, a Porter stemmer and a corpus-based stemmer have been developed for Bahasa Indonesia [7]. However, the evaluation using inflectional words from an Indonesian Dictionary [23] showed that the morphology-based algorithm performed better in identifying root words [7, 18, 21]. Applying a root-word dictionary to all of the stemmer algorithms improved the identification of root words further [7].

In evaluating the effectiveness of the stemmer algorithms, we applied the stemmers to the retrieval of Indonesian documents using an information retrieval system. The result shows that the performance of Porter and corpus-based stemmer algorithms for Bahasa Indonesia is comparable to that of the morphology-based algorithm.

In the field of information retrieval [25], stemming is used to abstract keywords from the morphological idiosyncrasies. Hopefully, improvement in retrieval performance is resulted. We noticed however a lower than expected retrieval performance after stemming (independently of the stemming algorithm). We explained this phenomenon by the fact that Indonesian morphology is essentially derivational (conceptual variations) as opposed to the morphologies of languages such as French or Slovene [19], which are primarily derivational (grammatical variations). This result refines the conclusion of [19] that the effectiveness of stemming is commensurate to the degree of morphological complexity in that we showed that it also depends on the nature of the morphology.

In a recent development of our research we have devised and evaluated a method for the mining of stemming rules from a training corpus of documents [11, 12]. The method induces prefix and suffix rules (and possibly infix rules although this feature is computationally intensive). The method achieves from 80% to 90% accuracy (i.e. is able to induce rules 80% to 90% of which are correct stemming) from corpuses as short as 10000 words. In the experiments above, we have successfully applied the method to the Indonesian and Italian languages as well as to Tagalog.

4. Part of Speech Tagging

Part-of-speech tagging is the task of assigning the correct class (part-of-speech) to each word in a sentence. A part-of-speech can be a noun, verb, adjective, adverb etc. Different word classes may occupy the same position, and similarly, a part-of-speech can take on different roles in a sentence. Automatic part-of-speech tagging is therefore the assignment of a part-of-speech class (or tag) to terms in a document.

In [11] and [12] we present several methods for the fully automatic acquisition of the knowledge necessary for part-of-speech tagging. The methods follow and extend the ideas in [21]. In particular they use various clustering algorithms. The methods we have devised neither use a tagged training corpus such as the method in [3] nor consider a predefined set of tags such as the method in [13]. Our evaluation of the effectiveness of the proposed methods using the Brown corpus [5] tagged by the Penn Treebank Project [16] shows that the best of our methods achieves a consistent improvement over all other methods to which we compared with more than 80% of the words in the tested corpus being correctly tagged.
The detailed results are illustrated in table 4.1. The table reports the average precision, recall and percentage of correctly tagged words for several methods based on trigrams (trigram 1,2 and 3), the state of the art methods (Schutze 1 and 2), and our proposed method (Extended Schutze).

Table 4.1: Part of Speech Tagging Performance

Method	Average Precision	Average Recall	% Correct
Trigram 1	0.70	0.60	64%
Trigram 2	0.74	0.62	66%
Trigram 3	0.76	0.62	67%
Extended Schutze's	0.90	0.72	81%
Schutze1	0.53	0.52	65%
Schutze2	0.78	0.71	80%

A particularly striking result is the appearance of finer granularity clusters of words which are not only of the same part of speech but also share the same affixes (e.g. "menangani", "mengatasi", "mengulangi" share the circumfix "me-i"), the same semantic category (e.g. "Indonesia", "Jepang", "Eropa", "Australia" are names of geo-political entities) or both (e.g. "mengatakan", "menyatakan", "mengungkapkan", "menegaskan" are synonyms meaning "to say"). Indeed, the Indonesian language has not only a derivational morphology, but also, as most languages, a concordance of the paradigmatic and the syntagmatic components.

5. Named Entity Extraction

The last remark suggests that a similar approach to the one we have used for part-speech tagging can be applied for a mainly paradigmatic tagging and therefore for the extraction of information.

To illustrate our objective, let us consider the motivating example from which we wish to extract an XML document describing the meeting taking place:
"British Foreign Office Minister O'Brien (right) and President Megawati pose for photographers at the State Palace."
Figure 5.1 contains the manually constructed XML we hope to obtain in fine. In italic are highlighted the components that require global, ancillary, or external knowledge. Indeed, although, we expect similar methods (association rules, maximum entropy) can be used to learn the model of combination of elementary entities into complex elements, we also expect that global, ancillary, and external knowledge will be necessary such as lists of names of personalities (Mike O'Brien, Megawati Sukarnoputri), gazetteers (Jakarta is in Indonesia), document temporal and geographical context (Jakarta, 05/06/2003), etc.
In [8, 9, 10] we present our preliminary results in an effort to extract structured information in the form of an XML document from texts. We believe this is possible under some ontological hypothesis for a given and well identified application domain. Our preliminary results are only concerned with the individual tagging of named entities such as locations, person names, and organizations. Table 5.1 illustrates the performance of an association rule based technique with a corpus of 1.258 articles from the online versions of two mainstream Indonesian newspaper Kompas (kompas.com) and Republika (republika.co.id).

Table 5.1 Named Entity Recognition Performance

Recall	Precision	F-Measure
60.16%	58.86%	59.45%

On the corpuses to which we have applied it, our method outperforms state of the art techniques such as [4].

```
<meeting>
<date>05/06/2003</date format=europe>
<location>
<name>State Palace</name> <city>Jakarta</city>
<country>Indonesia</country>
</location>
<participants>
<person>
<name>Megawati Soekarnoputri</name>
<quality>President </quality>
<country>Indonesia</country>
</person>
<person>
<name>Mike O'Brien</name>
<quality>Foreign Office
Minister</quality> <country>Britain</country>
</person>
</participants>
</meeting>
```

Figure 5.1: Sample XML extracted from a text

We applied the named entity tagger that identifies persons, organizations, and locations [10] to an information retrieval task, a question answering task for Indonesian documents [17]. The limited success of the experiment compels further research in this domain.

6. Conclusion

While attempting to design and implement tools and techniques for the processing of documents in the Indonesian language on the Web and for the construction of an Indonesian digital library, we were faced with the unavailability of linguistic data and knowledge as well as

with the prohibitive cost of data and knowledge collection.

This situation compelled the design and development of semi-automatic or automatic techniques for or ancillary to tasks as varied as language identification, stemming, part-of-speech tagging and information extraction. The dynamic nature of languages in general and of the Indonesian language in particular also compelled adaptive methods. We have summarized in this paper the main results we have obtained so far.

Our work continues in the same philosophy while addressing new tasks such as spelling error correction, structured information extraction as mentioned above, or phonology for text-to-speech and speech-to-text conversion.

References

[1] Adriani, Mirna and Rinawati. Finding Answers to Indonesian Questions from English Documents. In Working Notes of the Workshop in Cross-Language Evaluation Forum (CLEF), Vienna, September 2005.

[2] Bressan, S., and Indradjaja, L., Part-of-Speech Tagging without Training Intelligence in Proc. of Communication Systems, IFIP International Conference, INTELLCOMM (2004).

[3] Brill, E. Automatic Grammar Induction and Parsing Free Text: A Transformation-based Approach. In Proceedings of ACL 31. Columbus OH (1993).

[4] Chieu, H.L., and Hwee Tou Ng, "Named Entity Recognition: A Maximum Entropy Approach Using Global Information", Proceedings of the 19th International Conference on Computational Linguistics, (2002).

[5] Francis, W.N. and Kucera, F. Frequency Analysis of English Usage. Houghton Mifflin, Boston (1982).

[6] Harimurti Kridalaksana, Pembentukan Kata Dalam Bahasa Indonesia. P.T. Gramedia, Jakarta 1989.

[7] Ichsan, Muhammad. Pemotong Imbuhan Berdasarkan Korpus Untuk Kata Bahasa Indonesia. Tugas Akhir S-1, Fakultas Ilmu Komputer, Universitas Indonesia, 2005.

[8] Indra, Budi., Bressan, S., and Hasibuan, Z., Pencarian Association Rules untuk Pengenalan Entitas Nama. In Proc. of the Seminar on Bringing Indonesian Language toward Globalization through Language, Information and Communication Technology (2003). (in Indonesian)

[9] Indra, Budi. and Bressan, S., Association Rules Mining for Name Entity Recognition. In Proc. of Conference on Web Information Systems Engineering (WISE) (2003).

[10] Indra Budi, Stéphane Bressan, Gatot Wahyudi, Zainal A. Hasibuan, Bobby Nazief: Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach. Discovery Science (2005)

[11] Indradjaja, L. and Bressan, S., Automatic Learning of Stemming Rules for the Indonesian Language, In Proc. of the The 17th Pacific Asia Conference on Language, Information (2003).

[12] Indradjaja, L., and Bressan, S., Penemuan Aturan Pengakaran Kata secara Otomatis. In Proc. of the Seminar on Bringing Indonesian Language toward Globalization through Language, Information and Communication Technology (2003). (in Indonesian)

[13] Jelinek, F. Robust Part-of-speech Tagging using a Hidden Markov Model. Technical Report. IBM, T.J. Watson Research Center (1985).

[14] Lawrence, Steve and Giles, C. Lee. Searching the World Wide Web Science V 280, 1998.

[15] Lazzari, G., et all. Speaker-language identification and speech translation. Part of Multilingual Information Management: Current Levels and Future Abilities, delivered to US Defense ARPA, April 1999.

[16] Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. The Penn Treebank: Annotating Predicate Argument Structure. In ARPA Human Language Technology Workshop (1994).

[17] Natalia, Dessy. Penemuan Jawaban Pada Dokumen Berbahasa Indonesia. Tugas Akhir S-1. Fakultas Ilmu Komputer, Universitas Indonesia, 2006.

[18] Nazief, Bobby and Adriani, Mirna. A Morphology-Based Stemming Algorithm for Bahasa Indonesia. Technical Report. Faculty of Computer Science, 1996.

[19] Popovic, Mirko., and Willett, Peter., The effectiveness of stemming for natural-language access to Slovene textual data. Journal of the American Society for Information Science, Vo. 43, June 1992, pp 384-390.

[20] Schutze, Hinrich (1999). Distributional Part-of-speech Tagging. In EACL7, pages 141-148.

[21] Siregar, Neil Edwin F. Pencarian Kata Berimbuhan Pada Kamus Besar Bahasa Indonesia dengan menggunakan algoritma stemming. Tugas Akhir S-1, Fakultas Ilmu Komputer, Universitas Indonesia, 1995.

[22] Tim Penyusun Kamus, Kamus Besar Bahasa Indonesia.2ed. Balai Pustaka, 1999.

[23] Vinsensius, V., and Bressan, S., Continuous-Learning Weighted-Trigram Approach for Indonesian Language Distinction: A Preliminary Study. In Proceedings of 19th International Conference on Computer Processing of Oriental Languages, 2001.

[24] Vinsensius, V., and Bressan, S., "Temu-Kembali Informasi untuk Dokumen-dokumen dalam Bahasa Indonesia", In Electronic proceedings of Indonesia DLN Seminar, 2001. (In Indonesian).

[25] Yates, R. B. and Neto, B R., Modern Information Retrieval. ACM Press New York, 1999.

Searching Method for English-Malay Translation Memory Based on Combination and Reusing Word Alignment Information

Suhaimi Ab. Rahman, Normaziah Abdul Aziz, Abdul Wahab Dahalan
Knowledge Technology Lab
MIMOS,
Technology Park Malaysia,
57000 Kuala Lumpur, Malaysia.
smie@mimos.my, naa@mimos.my, wahab@mimos.my

Abstract

This paper describes the searching method used in a Translation Memory (TM) for translating English to Malay language. It applies **phrase look-up matching** techniques. In *phrase look-up matching*, the system locates the translation fragments in several examples. The longer the length of each fragment, the better the matching is. The system then generates the translation suggestion by combining these translation fragments. This technique generates a translation suggestion with the assistance of word alignment information.

1 Introduction

The purpose of Translation Memory system (TM) is to assist human translation by re-using pre-translated examples. Several work have been done in this area such as (Hua et al., 2005; Simard and Langlais, 2001; Macklovitch and Russell, 2000) among others. A TM system has three parts: a) Translation memory itself, which records example translation pairs (together with word alignment information); b) Search engine, which retrieves related examples from the translation memory and c) On-line learning mechanism, which learns newly translated translation pairs. When translating a sentence, the TM provides the translation of the best-matched pre-translated example as the translation suggestion.

We developed a TM as an additional tool on top of our existing Machine Translation system[1] to translate documents from English to Malay.

2 English-Malay TM System - Basic Principle

Zerfass (2002) described the text to be translated consists of smaller units like headings, sentences, list items, index entries and so on. These text components are called segments. Figure 1 shows an overall process of lookup segment using phrase look-up matching.

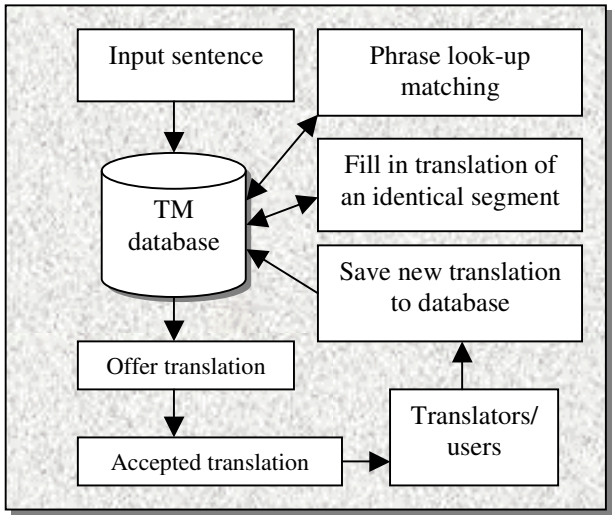


Figure 1. An overall process of lookup segment using phrase look-up matching

[1] The present MT is an EBMT, a project we embarked with Universiti Sains Malaysia and available for usage at www.terjemah.net.my

3 Phrase Look-up Matching

The phrase look-up matching is used to find suggested meaning for a phrase by parsing the phrase into sub-phrases and finding a meaning to these sub-phrases and combine the results to get the final output. Below is a figure showing an example of the phrase look-up matching process.

input sentence:
Selected planting materials are picked when they are 30-60 cm high at about 4 months before harvesting

split into:
Selected planting materials are picked when they are 30-60 cm high at about 4 months before | harvesting
process the left part, no result found, split further
:
:
Selected planting materials are picked when they are 30-60 cm high | at about 4 months before harvesting
result found, applying algorithm, add result to the output

basic output:
"Bahan-bahan tanaman terpilih dipetik apabila ianya mencapai ketinggian 30-60 sm"
process the right side
at about 4 months before harvesting
result found, applying algorithm,
add result to previous output

basic output:
"*Bahan-bahan tanaman terpilih dipetik apabila ianya mencapai ketinggian 30-60 sm" + "pada kira-kira 4 bulan sebelum penuaian"*

Figure 2. Example of The Phrase Look-Up Matching using a Bi-Section Algorithm

3.1 Repetition Avoidance

The basic output has no problems in structure, but it may contain repeated words. These repeated words are generated because of the way we deal with the source target pairs. We implement the repetition avoidance algorithm to solve this problem. Let us consider an example shows in Figure 3.

The target word ***"anda"*** for the source *you* and *your* is repeated 3 times in the output although *your* is mentioned once in the source sentence.

Input Sentence:
If you choose a non-clinical program.

Example retrieved from Sentence Alignment Table (SAT):
Source sentence (E): If ***you*** choose a non-clinical program ***you*** have a greater responsibility for monitoring ***your*** own health
Target sentence (M): *Jika* ***anda*** *memilih suatu program bukan klinikal* ***anda*** *mempunyai tanggungjawab yang lebih besar untuk memantau kesihatan* ***anda***

Basic Output:
"*Jika* ***anda*** *memilih suatu program bukan klinikal* ***anda anda***"

Figure 3. An Example of Basic Output With a Repetition of Word

We might notice that the word *you* and *your* are two different words, treated as two repeated similar words by the program, this is because the repetition avoidance algorithm consider ***"anda"*** in the target sentence as a repeated word, and since *you* and *your* both means ***"anda"*** in Malay, the repetition avoidance algorithm will consider 3 repeated ***"anda"*** in the output.

To determine which are the words ***"anda"*** need to be select from the above basic output, we used a mathematical model, named as the inter-phrase word-to-word distance summation.

3.2 Inter-Phrase Word-to-Word Distance Summation

This algorithm uses mathematical calculation to calculate a summation value that will give a clue on which word is repeated and needs to be omitted and which one is not. Each word that belongs to the basic output will have one summation value calculated. We call that summation value as d_j. The word with a big d_j value is more likely to be a repeated word. In order to know the summation value d_j of that word, we have to sum the word-to-word distance value between that word and the rest of the word in that basic output. Word to word distance is the number of words that separate one word to another in the original SAT entry. The selection combination of each word/s from SAT is based on the aligned word retrieved from Word Alignment Table (WAT).

In order to determine the value for each of the distance word between word in basic output ($\mathbf{loc_i}$) and word in SAT's target sentence ($\mathbf{loc_j}$) we will use this formula: $\mathbf{d_i = |loc_j - loc_i|}$

where :

$\mathbf{i,j} = 0,1,\ldots,n$

$\mathbf{loc_i}$: location value for basic output

$\mathbf{loc_j}$: location value for SAT's target sentence

By applying this formula, the distance value for $\mathbf{d_i}$ will be getting by doing subtraction of the two values - $\mathbf{loc_j}$ and $\mathbf{loc_i}$. This process will continue until all location of the words has been subtracted properly.

Table 1 shows the value of loc_i and loc_j for each of word in the basic output and target sentence from SAT, while Table 2 shows the matrix table for all distance values $\mathbf{d_i}$.

Running the Inter-Phrase Word-to-Word Distance Summation algorithm will first give us the total summation of all distance calculation result.

The $\mathbf{d_j}$ is a summation for all the distance word values generated from $\mathbf{d_i}$. Below is an equation formula on how we can get the summation value for $\mathbf{d_j}$:

$$\forall \mathbf{j} \rightarrow \quad \mathbf{d_j} = \sum_{i=0} \mathbf{| loc_j - loc_i |}$$

It is the sums of the absolute value of the difference between the locations of word $\mathbf{loc_j}$ and each other single entry in the basic output word $\mathbf{loc_i}$. The value of this summation will be the main source of judgment for the choice between repeated words.

Table 2 describes the details for the summation value of word-to-word distances. The d_j summation values will be used as judgment value to omit the extra ***"anda".*** We have crossed the row values belong to the conflicted words, for example word ***"anda".*** Then we sum the rest to get the $\mathbf{d_j}$ of each word.

Since we have two words *you* in the source sentence from SAT, but there is only one in the input, so one of them must be omitted.

Figure 4 depicts the plot of the summations values ($\mathbf{d_j}$) vs the basic output (loc_i).

idx	Basic output	$\mathbf{loc_i}$	SAT	$\mathbf{loc_j}$
0	jika	0	Jika	0
1	anda	1	anda	1
2	memilih	2	memilih	2
3	suatu	3	suatu	3
4	program	4	program	4
5	bukan klinikal	5	bukan klinikal	5
6	anda	6	anda	6
7	anda	7	mempunyai	7
			:	
			lebih besar	10
			:	
			anda	14

Table 1. The Location of word in the basic output and SAT.

The thick dotted line represents the margin between acceptance and non-acceptance words, i.e. everything left hand side the thick dotted line is acceptance and the rest is not.

After removing the dropped words from the basic output, we will have the final output as follows: ***"Jika anda memilih suatu program bukan klinikal".***

4 Result

We have tested this technique with our 7,000 English-Malay bilingual sentences and found that phrase lookup matching with Inter-Phrase Distance Summation technique can reduce some important error for the repetition of the same word meaning from the target sentence.

5 Conclusion

This paper describes a translation memory system using a look-up matching techniques. This technique generates translation suggestions through word alignment information in the pre-translated examples. We have also implemented a mathematical model that is used to make logical judgments that helps in maintaining the accuracy of the output sentence structure. The accuracy and the quality of the translation is dependent on the number of the examples in our TM database i.e. we can improve the quality by increasing the number of examples in the translation memory and word alignment information database.

WAT's Target Sentence / Basic Output			loc_j													
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			Jika	anda	memilih	suatu	program	bukan klinikal	anda	mempunyai	tanggungjawab	yang	lebih besar	untuk	memantau	kesihatan	anda
loc_i	0	Jika	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	1	anda	~~1~~	~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~
	2	memilih	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12
	3	suatu	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11
	4	program	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10
	5	bukan klinikal	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9
	6	anda	~~6~~	~~5~~	~~4~~	~~3~~	~~2~~	~~1~~	~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~
	7	anda	~~7~~	~~6~~	~~5~~	4	3	2	1	0	1	2	3	4	5	6	7
d_j			14	11	8	7	8	11	16	21	26	31	36	41	46	51	56

Table 2. Summation of word-to-word Distances

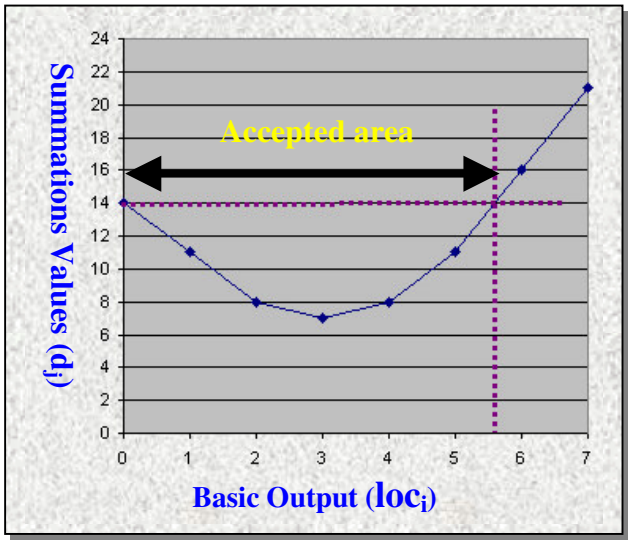


Figure 4. Relation Between the Index and the Inter-Phrase Distance Summation

Acknowledgement

We would like to acknowledge our research assistant, Ahmed M.Mahmood from the International Islamic University Malaysia, who had also contributed his ideas in this work.

References

Angelika Zerfasss. 2002. Evaluating Translation Memory Systems. *In Proc. of the workshop on "Annotation Standards for Temporal Information in Natural Language" (LREC 2002),* Las Palmas, Canary Islands – Spain.

Atril – Déjà Vu. http://www.atril.com

Elliott Macklovitch and Graham Russell. 2000. What's been Forgotten in Translation Memory. *In Proc. of the 4th Conference of the Association for Machine Translation in the Americas (AMTA-2000),* pages 137-146. Mexico.

Michel Simard and Philippe Langlais. 2001. Sub-Sentential Exploitation of Translation Memories. *In Proc. of the 8th Machine Translation Summit (MT Summit VIII),* pages 331-339, Santiago de Compostela, Galicia, Spain.

Trados – Translator's Workbench. http://www.trados.com/

WU Hua, WANG Haifeng, LIU Zhanyi, TANG Kai. 2005. Improving Translation Memory with Word Alignment Information. *In Proc. of the 10th Machine Translation Summit (MT Summit X),* pages 364-371, Phuket, Thailand.

A Phrasal EBMT System for Translating English to Bengali

Sudip Kumar Naskar
Comp. Sc. & Engg. Dept.
Jadavpur University
Kolkata, India 700032
sudip.naskar@gmail.com

Abstract

The present work describes a hybrid MT system from English to Bengali that uses the TnT tagger to assign POS category to the tokens, identifies the phrases through a shallow analysis, retrieves the target phrases using a Phrasal Example Base and then finally assigns a meaning to the sentence as a whole by combining the target language translations of the constituent phrases.

1 Introduction

Bengali is the fifth language in the world in terms of the number of native speakers and is an important language in India. But till date there is no English- Bengali machine translation system available (Naskar and Bandyopadhyay, 2005b).

2 Translation Strategy

In order to translate from English to Bengali (Naskar and Bandyopadhyay, 2005a), the tokens identified from the input sentence are POS tagged using the hugely popular TnT tagger (Brants, 2000). The TnT tagger identifies the syntactic category the tokens belong to in the particular context. The output of the TnT tagger is filtered to identify multiword expressions (MWEs) and the basic POS of each word / term alongwith additional information from WordNet (Fellbaum, 1998). During morphological analysis, the root words / terms (including idioms, named entities, abbreviations, acronyms), along with associated syntactico-semantic information are extracted. Based on the POS tags assigned to the words / terms, a rule-based chunker (shallow parser) identifies the constituent chunks (basic non-recursive phrase units) of the source language sentence and tags them to encode all relevant information that might be needed to translate this phrase and perhaps resolve ambiguities in other phrases. A DFA has been written for identifying each type of chunk: NP, VP, PP, ADJP and ADVP. Verb phrase (VP) translation scheme is *rule based* and uses Morphological Paradigm Suffix Tables. Rest of the phrases (NP, PP, ADJP and ADVP) are translated using Example bases of syntactic transfer rules. A phrasal Example Base is used to retrieve the target language phrase structure corresponding to each input phrase. Each phrase is translated individually to the target language (Bengali) using Bengali synthesis rules (Naskar and Bandyopadhyay, 2005c). Finally, those target language phrases are arranged using some heuristics, based on the word ordering rules of Bengali, to form the target language representation of the source language sentence. Named Entities are transliterated using a modified joint source-channel model (Ekbal et al., 2006).

The structures of NP, ADJP and ADVP are somewhat similar in both English and Bengali. But the VP and PP constructions differ markedly in English and Bengali. First of all, in Bengali, there is no concept of preposition. English prepositions are handled in Bengali using inflexions to the reference objects (i.e., the noun that follows a preposition in a PP), and / or post-positional words after them (Naskar and Bandyopadhyay, 2006). Moreover, inflexions in Bengali get attached to the reference objects and relate it with the main verb of the sentence in c*ase* or k*araka* relations. An inflexion has no existence of its own in Bengali, and it does not have any meaning as well, but in English prepositions have their own existence, i.e., they are separate words. Verb phrases in both English and Bengali depend on the person, number information of the subject and tense and aspect information of the verb. But for any particular root

verb, there are only a few verb forms in English, whereas in Bengali it shows a lot of variations.

3 POS Tagging and Morphological Analysis

The input text is first segmented into sentences. And each sentence is tokenized into words. The tokens identified at this stage are then subjected to the TnT tagger that assigns a POS tag to every word. The HMM based TnT tagger (Brants, 2000) is at per with other state-of-the-art POS taggers.

The output of the TnT tagger is filtered to identify MWEs using WordNet and additional resources like list of acronyms, abbreviations, named entities, idioms, figure of speech, phrasal adjectives, phrase prepositions.

Although, the freely available WordNet (version 2.0) package provides with it the set of programs for accessing and integrating WordNet, we have developed our own interface to integrate WordNet into our system for implementing the particular set of functionalities required for our system.

In addition to the existing eight noun suffixes in the WordNet, we have added three more noun suffixes – " 's ", " ' ", " s' " in the noun suffix set. Multiword expressions or terms are identified in this phase and are treated as a single token. These include multi-word nouns, verbs, adjectives, adverbs, phrase prepositions, phrase adjectives, idioms etc. Sequences of digits and certain types of numerical expressions, such as dates and times, monetary expressions, and percents are also treated as a single token. They can also appear in different forms as any number of variations.

4 Syntax Analysis

In this module, a rule-based (chunker) shallow parser has been developed that identifies and extracts the various chunks (basic non-recursive phrase units) from a sentence and tags them.

A sentence can have different types of phrases - NP, VP, PP, ADJP and ADVP. We have defined a formal grammar for each of them that identify the phrase structure based on the POS information of the tokens (words / terms).

For example, the system chunks the sentence "*Teaching history gave him a special point of view toward current events*" as given below:

```
[NP Teaching history] [VP gave] [NP
him] [NP a special point of view]
[PP toward current events].
```

5 Parallel Example Base

The tables containing the proper nouns, acronyms, abbreviations, and figure of speech in English and the corresponding Bengali translation are the literal example bases. The phrasal templates for the NPs, PPs, ADJPs, and ADVPs store the part of speech of the constituent words along with necessary semantic information. The source and the target phrasal templates are stored in example bases, expressed as context-sensitive rewrite rules, using semantic features. These translation rules are effectively transfer rules. Some of the MWEs in the WordNet represent pattern examples (e.g., `make up one's mind`, `cool one's heels`, `get under one's skin`; one's representing a possessive pronoun).

6 Translating NPs and PPs

NPs and PPs are translated using phrasal example base and bilingual dictionaries. Some examples of transfer rules are given below for NPs:

```
<det & a> <n & singular, human,
nom> → <ekjon> <n′>
<det & a> <adj> <n & singular,
inanimate> → <ekti> <adj′> <n′>
<prn & genitive> <n & plural,
human, nom> → <prn′> <n′> <-era/ra>
```

Below are some examples of transfer rules for PPs.

```
<prep & with/by> <n & singular,
instrument > → <n′> <diye>
<prep & with> <n & singular, person
> → <n′> <-yer/er/r> <songe>
<prep & before> <n & artifact> →
<n′> <-yer/er/r> <samne>
<prep & before> <n & !artifact> →
<n′> <-yer/er/r> <age>
<prep & till> <n & time/place> →
<n′> <porjonto>
<prep & in/on/at> <n & singular,
place> ↔ <n′> <-e/te/y>
```

Using the transfer rules, we can translate the following NPs as:

<det & a> <n & man (sng, human, nom)> ←→ <ekjon> <chele>
<det & a > <n & book (sng, inanimate, acc)>←→<ekti> <boi>
<prn & my (gen)> <n & friends (plr, human, nom)> ←→ <amar> <bondhura>
<n & Ram's (sng, gen)> <n & friends (plr, human, dat)> ←→ <ramer> <bondhuderke>

Similarly, below are some candidate PP translations.

<prep & with> <prn & his (gen)> <n & friends (plr, human, nom)> ←→ <tar> <bondhuder> <sathe>
<prep & in> <n & school (sng, inanimate, loc)> ←→ <bidyalaye>

7 Translating VPs

Bengali verbs have to agree with the subject in person and formality. Bengali verb phrases are formed by appending appropriate suffixes to the root verb. Some verbs in English are translated in Bengali using a combination of a semantically 'light' verb and another meaning unit (a noun, generally) to convey the appropriate meaning. In English to Bengali context this phenomenon is very common, e.g., to swim – *santar* (swimming) *kata* (cut), to try – *chesta* (try) *kara* (do).

Bengali verbs are morphologically very rich. A single verb root has many morphological variants. The Bengali representation of the 'be' verb is formed by suffixing to the present root *ach*, past root *chil* and the future root *thakb* for appropriate tense and person information. The negative form of the 'be' verb in present tense is *nei* for any person information. And in past and future tense, it is formed by simply adding the word *na* postpositionally after their corresponding assertive form.

Root verbs in Bengali can be classified into different groups according to the spelling pattern. All the verbs belonging to the same spelling pattern category, take the same suffix for same person, tense, aspect information. These suffixes also change from the Classical to Colloquial form of Bengali. There are separate morphological paradigm suffix tables for the verb stems that have the same spelling pattern. There are some exceptions to these rules.

The negative forms are formed by adding *na* or *ni* postpositionally. Other verb forms (gerund-participle, dependent gerund, conjunctive participle, infinitive-participle etc.) are also taken care of in the same way by adding appropriate suffixes from a suffix table. Further details can be seen in (Naskar and Bandyopadhyay, 2004).

8 Word Sense Disambiguation

The word sense disambiguation algorithm is based on *eXtended WordNet* (version 2.0-1.1) (Harabagiu et al, 1999). The algorithm takes a *global* approach where all the words in the context window are simultaneously disambiguated in a bid to get the best combination of senses for all the words in the window instead of only a single word. The context window is made up of the all WordNet word tokens present in the current sentence under consideration. A word bag is constructed for each sense of every content word. The word bag for a word-sense combination contains synonyms and content words from the associated tagged glosses of the *synsets* that are related to the word-sense through various WordNet relationships for different parts of speech. Each word (say W_i) in the context is compared with every word in the gloss-bag for every sense (say S_j) of every other word (say W_k) in the context. If a match is found, they are checked further for part-of-speech match. If the words match in part-of-speech as well, a score is assigned to both the words: the word being matched (W_i) and the word whose gloss-bag contains the match (W_j). This matching event indicates mutual confidence towards each other, so both words are rewarded by scoring for this event. A word-sense pair thus gets scores from two different sources, when disambiguating the word itself and when disambiguating neighboring words. Finally, these two scores are combined to arrive at the combination score for a word-sense pair. The sense of a word for which maximum overlap is obtained between the context and the word bag is identified as the disambiguated sense of the word. The baseline algorithm is modified to include more contexts. Increase in the context size, by adding the previous and next sentence in the context, resulted in much better performance. It resulted in 61.77% precision and 85.9% recall, tested on the first 10 Semcor 2.0 files.

9 Resources

WordNet (version 2.0) is the main lexical resource used by the system. We have a separate non-content word dictionary. An English-Bengali dictionary has been developed which maps WordNet English synsets to its Bengali synsets. For the actual translation purpose, the first Bengali word (synonym) in the synset is always taken by the system. So, there is no scope for lexical choice in this work. But, during the dictionary development, the Bengali word that is mostly used by the native speakers is kept at the beginning of the Bengali synset. So effectively, the most frequently used Bengali synonyms are picked up by the system during the dictionary look up.

Figure of Speech expressions in English have been paired with their corresponding counterparts in the target language and these pairs have been stored in a separate Figure of Speech Dictionary. Idioms also are translated using a direct example base. The morphological suffix paradigm tables are maintained for all verb groups. They help in translating VPs.

Named Entities are transliterated. If there is any acronym or abbreviation within the named entity, it is translated. For this translation purpose, the system uses an acronym / abbreviation dictionary that includes the different acronyms/abbreviations occurring in the news domain and their corresponding representation in the Bengali. The transliteration scheme is knowledge-based. It has been trained on a bilingual proper name example-base containing more than 6000 parallel names of Indian origin. The transliteration process uses a modified joint source-channel approach. The transliteration mechanism (especially the chunking of transliteration units) is linguistically motivated and makes use of a linguistic knowledge base.

For sense disambiguation purpose we make use of *eXtended WordNet* (version 2.0-1.1).

10 Conclusion

Anaphora resolution has not been considered by the system, sine this is required only for proper translation of personal pronouns. Only second and third person personal pronouns have honorific variants in Bengali. Pronouns can be translated assuming a default highest honor.

The system has not been evaluated as some parts (specially the dictionary creation) are under development. We intend to evaluate the MT system using the BLEU metric (Papineni et al., 2002).

References

Asif Ekbal, Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2006. *A Modified Joint Source-Channel Model for Transliteration*; In the proceedings of COLING-ACL 2006, Sydney, Australia.

Fellbaum, C. ed., *WordNet – An Electronic Lexical Database*, MIT Press, 1998.

Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. IBM Research Division Technical Report, RC22176 (W0190-022), Yorktown Heights, N.Y.

S. Harabagiu, G. Miller, D. Moldovan, WordNet2 - a Morphologically and Semantically Enhanced Resource. In Proceedings of SIGLEX-99, pages 1-8, University of Mariland, 1999.

Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2006. *Handling of Prepositions in English to Bengali Machine Translation*. In the proceedings of Third ACL-SIGSEM Workshop on Prepositions, EACL 2006. Trento, Italy.

Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2005a. *A Phrasal EBMT System for Translating English to Bengali*. In the proceedings of MT SUMMIT X. Phuket, Thailand.

Sudip Naskar and Sivaji Bandyopadhyay. 2005b. *Use of Machine Translation in India: Current Status*. Proceedings of MT SUMMIT X. Phuket, Thailand.

Sudip Naskar, Sivaji Bandyopadhyay. 2005c. *Using Bengali Synthesis Rules in an English-Bengali Machine Translation System*. In the Proceedings of Workshop on Morphology-2005, 31st March, 2005. IIT Bombay.

Sudip Kumar Naskar, Sivaji Bandyopadhyay. 2005d. *Transliteration of Indian Names for English to Bengali*. In the Proceedings of Platinum Jubilee International Conference of Linguistic Society of India, University of Hyderabad. December 6-8, 2005.

Sudip Naskar and Sivaji Bandyopadhyay. 2004. *Translation of Verb Phrases from English to Bengali*, In the Proceedings of the CODIS 2004. Kolkata, India.

Thorsten Brants. *TnT a statistical part-of-speech tagger*. In Proceedings of the 6th Applied NLP Conference, pages 224--231, 2000.

PREPOSITIONS IN MALAY: Instrumentality

Zahrah Abd Ghafur
Universiti Kebangsaan Malaysia, Kuala Lumpur

Abstract

This paper examines how Malay language manifests instrumentality. Two ways were shown to be the case: by introducing the notion with the preposition **dengan,** and by verbalisation of the instruments itself. The same preposition seems to have carried other notions too if translated to English. But, in thinking in Malay, there is only one underlying notion i.e. the prepositional phrase occurs at the same time as the action verb.

Instrumentality

Language has peculiar ways in introducing instruments. Malay has at least two ways in expressing this notion: one is by using preposition and the other is by verbalising the instruments themselves.

A. Preposition that introduced objects as instruments to perform the actions in the main sentence:

1. dengan

STRUCTURE 1: The instrument is introduced by preposition: *dengan*

X	**actions**	**Y**	**dengan**	**Objects (instruments)**
Dia	*memukul*	*anjing itu*	*dengan*	*sebatang kayu.*
He	hit	the dog	with	a stick
Dia	*menghiasi*	*rumahnya*	*dengan*	*peralatan moden.*
She	ornamented	her house	with	modern equipment.

In the above examples, the word *menggunakan* 'use' can be inserted between the prepositions and the instruments. Together the group *'dengan menggunakan'* can be translated as 'using' or 'with the use of'.

X	**actions**	**Y**	**dengan + menggunakan**	**Objects (instruments)**
Dia	*membuka*	*sampul surat itu*	*dengan menggunakan*	*pembuka surat.*
He	opened	the envelope	Using/ with the use of	a letter opener
Dia	*memukul*	*anjing itu*	*dengan menggunakan*	*sebatang kayu.*
He	hit	the dog	Using/ with the use of	a stick
Dia	*menghiasi*	*rumahnya*	*dengan menggunakan*	*peralatan moden.*
She	decorated	her house	Using/ with the use of	modern equipment.

These examples show that it is possible to delete 'menggunakan' from the group 'dengan menggunakan' without losing their instrumental meaning.

STRUCTURE 2: The instrument is one of the arguments of the verb *menggunakan 'use'* and the action is explicitly expressed after the preposition *untuk 'for ing/to'.*

X	**menggunakan**	**Objects (instruments)**	**untuk**	**actions**	**Y**
Dia	*menggunakan*	*baji*	*untuk*	*membelah*	*kayu itu.*
He	used	(a) wedge	to	split	the wood

All the structures in 1 can be paraphrased into structure 2 and vice versa:

X + actions + Y+ dengan + menggunakan + objects (instruments) ⟷ X + menggunakan + objects (instruments) + untuk + actions + Y

e.g.

X	**menggunakan**	**Objects (instruments)**	**untuk**	**actions**	**Y**
Dia	*menggunakan*	*pembuka surat*	*untuk*	*membuka*	*sampul surat itu*
He	used	a letter opener	to	open	the envelope
Dia	*menggunakan*	*sebatang kayu.*	*untuk*	*memukul*	*anjing itu*
He	used	a stick	to	hit	the dog
Dia	*menggunakan*	*peralatan moden.*	*untuk*	*menghiasi*	*rumahnya*
She	used	modern equipment.	to	decorate	her house

STRUCTURE 3: Verbalisation of a Noun_Instrument: meN- + Noun_Instrument
In Malay most noun instruments can be verbalised by prefixing with the prefix meN-.

Noun_Instrument	**Verb**	**e.g.**
tenggala 'a plough'	*menenggala* 'to plough'	*Ali menenggala tanah sawahnya.* 'Ali ploughed his padi field.'
gunting 'scissors'	*menggunting* 'to cut'	*Dia menggunting rambutnya.* 'He has his hair cut'
komputer 'computer'	*mengkomputerkan* 'to computerise'	*Dia mengkomputerkan sistem pentadiran.* 'He computerised the administrative system'.

Menenggala **'to plough using a *tenggala*.'**
Menggunting **'to cut using a *gunting'***
Mengkomputerkan **'to *computer*ise'**

In these cases, the derived verbs will adopt the default usage of the instruments. Probably that accounts for
i. not all instruments can be verbalised in this way.
ii. instruments which can be verbalised in this way are instrument with specific use.

Thus:
1. *memisau is never derived from pisau 'knife' (*pisau* has many uses)
2. Menggunting will always mean 'cutting with a pair of scissors'.
   Membunuh seseorang dengan menggunakan gunting (killed someone with a pair of scissors) can never be alternated with *menggunting seseorang*.

Other examples:
- *Dia membelah kayu **dengan kapak**. (He split the piece of wood with an axe)*
  = Dia **mengapak** kayu.
- *Mereka menusuk kadbod itu dengan gunting. (They pierced the cardboard with scissors)*
  * Mereka **menggunting** kadbod.
- *Dia menggunting berita itu dari surat khabar semalam. (He clipped the news item from yesterday's papers)*
  = Dia memotong berita itu **dengan gunting** …
- *Orang-orang itu **memecahkan** lantai dengan menggunakan tukul besi. (The men were breaking up the floor with hammers).*
  ***menukul** lantai*
  √***menukul** paku (to hammer nails)*
- *Pada masa dahulu tanaman dituai dengan menggunakan sabit. (In those days the crops were cut by hand with a sickle.*
  √***menyabit** tanaman*

Other than introducing instruments**, dengan** also introduces something else**.**

a. Accompaniment:
STRUCTURE 4: Verb + dengan + NP entity

Verb (intransitive)	**dengan**	**NP Entity**
berjalan 'to walk'	*dengan* 'with'	*Ali*. 'Ali'
bercakap 'talk'	*dengan* 'to'	*dia* 'him'
bersaing *'to compete'*	*dengan* *'with'*	*seseorang* 'someone'.

Verb (transitive)	Object	dengan	NP Entity
membuat ‘to do’	*Sesuatu* *'something'*	*dengan* ‘with’	*Ali*. ‘Ali’
menyanyi ‘to sing’	*lagu-lagu asli* 'traditional songs'	*dengan* ‘with’	*Kawan-kawannya* ‘his friends’
membina 'to start'	*Sebuah keluarga* 'a family'	*dengan* *'with'*	*seseorang* 'someone'.

The use of **Dengan** in both these structures may be alternated with bersama-sama 'together with'. In all the cases examined, in these structures the prepositional phrase (PP) accompanies the subject of the sentence and the action verb is capable of having multiple subjects at the same time.

If a verb has a default of having a single subject, the dengan PP will accompany the object of the verb:

Verb (transitive)	Object	dengan	NP Entity
menggoreng ‘to fry’	*ikan* *'fish'*	*dengan* ‘with’	*kunyit*. ‘turmeric'
ternampak ‘saw’	*seseorang* 'someone'	*dengan* ‘with’	*kawan-kawannya* ‘his friends’
membeli 'to buy	*Sebuah rumah* 'a house'	*dengan* *'(together) with'*	*Tanahnya sekali.* 'the land'.

b. Quality:

STRUCTURE 5: Verb (intransitive) + dengan + NP Quality

Verb (intransitive)	dengan	Quality (adj/adv)
berjalan ‘to progress’	*dengan*	*lancar*. ‘smoothly’
bercakap ‘talk’	*dengan*	*kuat* ‘loudly’
bersaing *'to compete'*	*dengan*	*adil* 'fairly'
bekerja *'to work'*	*dengan*	*keras* *'hard'*

Verb (transitive) + dengan + NP Quality

Verb (transitive)	Object	dengan	Quality (adj/adv)
menyepak ‘to kick’	*bola* *'the ball'*	*dengan*	*cantik.* ‘beautifully'
mengikut ‘folow'	*peraturan* 'the rule'	*dengan*	*Berhati-hati* ‘faithfully’
menutup 'to close'	*pintu* 'the door'	*dengan*	*kuat.* 'forcefully'.

In these cases the PP will modify the transitive as well as the intransitive verb forming an adverbial phrase describing quality (stative description).

If these qualities are substituted by verb phrases (VPs), the group will refer to manner.

c. Manner:

STRUCTURE 6: Verb (intransitive) + dengan + VP

Verb (intransitive)	dengan	VP
berjalan ‘to walk'	*dengan* by	*Mengangkat kaki tinggi-tinggi.* ‘lifting the feet high’
menyanyi ‘to sing’	*dengan* in	*Menggunakan suara tinggi.* ‘in a loud voice’
melawan *'to compete'*	*dengan* *'by'*	*Menunjukkan kekuatannya.* 'exhibiting his strength'

Verb (transitive) + dengan + VP

Verb (transitive)	Object	dengan	VP
mempelajari ‘learning'	*sesuatu* *'something'*	*dengan* by	*Membaca buku.* 'reading (books)'

mengikut ‘folow'	*peraturan* 'the rule'	*dengan* by	*Membeli barang-barang tempatan.* ‘buying local products'
menutup 'to cover'	*makanan* 'the food'	*dengan* *by*	*Meletakkan daun pisang di atasnya.* 'putting banana leaves over it'.
membeli 'to buy'	*Sebuah rumah* 'a house'	*dengan* 'on'	*berhutang* 'loan'.

Stative verbs can be modified by an NP introduced by **dengan**.

d. Modifier to stative verbs:

STRUCTURE 7: Verb (intransitive) + dengan + VP

Stative Verb (adj?)	**dengan**	**NP**
taat ‘faithful'	*dengan* to	*Perintah agama.* ‘religious teachings’
tahan ‘stand’	*dengan*	*kritikan.* ‘the criticism’
meluat *'pissed off'*	*dengan* *'by'*	*Masalah dalaman.* 'the internal problems'
senang *'comfortable'*	*dengan* *'with'*	*Dasar itu.* *'the policy'*

e. complement to certain verbs:

STRUCTURE 8: Verb + dengan + NP

Verb	**dengan**	**NP**
berhubung ‘connected'	*dengan* to	*sesuatu* ‘something’
berseronok ‘enjoying’	*dengan*	*Keadaan itu.* ‘the event’
Tak hadir *'absent'*	*dengan* *with*	*kebenaran.* 'permission'
Ditambah 'Adding'	*dengan*	*Vitamin A*
Diperkuatkan 'reenforced'	*Dengan* 'with'	*Kalsium* calsium

f. to link comparative NPS:

STRUCTURE 9: Comp Prep + NP + dengan + NP

Comp PREP	**NP**	**dengan**	**NP**
Di antara ‘between'	*A*	*dengan* and	*B*
bagaikan ‘as'	*Langit* 'the sky'	*dengan* 'and'	*Bumi..* ‘the earth'
Tak hadir *'absent'*		*dengan* *with*	*kebenaran.* 'permission'
Ditambah 'Adding'		*dengan*	*Vitamin A*
Diperkuatkan 'reenforced'		*Dengan* 'with'	*Kalsium* 'calcium'

Conclusion

The use of the same form of the preposition may point to one direction. It’s a manifestation of the same idea in the language. It suggests that the prepositional phrase occurs at the same time as the verb it qualifies.