COLING 2010

23rd International Conference on Computational Linguistics

Proceedings of the

8th Workshop on Asian Language Resources

21-22 August 2010
Beijing, China

Produced by
Chinese Information Processing Society of China
All rights reserved for Coling 2010 CD production.

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Preface

Language resources play a central role in statistical and learning-based approaches to natural language processing. Thus, recent research has put great emphasis in building these resources for target languages. Parallel resources across various languages are also being developed for multilingual processing. These include lexica and corpora with multiple levels of annotations. Though significant progress has been achieved in modeling few of the Asian languages, with the wider spread of ICT use across the region, there is a growing interest in this field from other linguistic communities. As research in the field matures across Asia, there is a growing need for developing language resources. However the region is not only short in the linguistic resources for more than 2200 language spoken in the region, there is also lack of experience in the researchers to develop these resources. As the efforts to develop the linguistic resources increases, there is also need to coordinate the efforts to develop common frameworks and processes so that these resources can be used by various groups of researchers equally effectively.

The workshop is organised by the Asian Language Resources Committee (ALRC) of Asian Federation for Natural Language Processing (AFNLP). The aim are to chart and catalogue the status of Asian Language Resources, to investigate and discuss the problems related to the standards and specification on creating and sharing various levels of language resources, to promote a dialogue between developers and users of various language resources in order to address any gaps in language resources and practical applications, and to nurture collaboration in their development and use, to provide opportunity for researchers from Asia to collaborate with researchers in other regions.

This is the eighth workshop in the series, and has been representative, with 35 submissions for Asian languages, including Bahasa Indonesia, Chinese, Dzongkha, Hindi, Japanese, Khmer, Sindhi, Sinhala, Thai, Turkish and Urdu, of which 22 have been finalized for presentation. We would like to thank the authors for their submissions and the Program Committee for their timely reviews. We hope that ALR workshops will continue to encourage researchers to focus on developing and sharing resources for Asian languages, an essential requirement for research in NLP.

ALR8 Workshop Organizers

Organizers:

Sarmad Hussain, CLE-KICS, UET, Pakistan
Virach Sornlertlamvanich, NECTEC, Thailand
Hammam Riza, BPPT, Indonesia
(on behalf of ALRC, AFNLP)

Program Committee:

Mirna Adriani
Pushpak Bhatacharyya
Francis Bond
Miriam Butt
Thatsanee Charoenporn
Key-Sun Choi
Ananlada Chotimongkol
Jennifer Cole
Li Haizhou
Choochart Haruechaiyasak
Hitoshi Isahara
Alisa Kongthon
Krit Kosawat
Yoshiki Mikami
Cholwich Nattee
Rachel Roxas
Dipti Sharma
Kiyoaki Shirai
Thepchai Supnithi
Thanaruk Theeramunkong
Takenobu Tokunaga
Ruvan Weerasinghe
Chai Wutiwiwatchai
Yogendra Yadava

Table of Contents

Conference Program

Saturday August 21, 2010

Semantics

9:00–9:25 *A Thesaurus of Predicate-Argument Structure for Japanese Verbs to Deal with Granularity of Verb Meanings*
Koichi Takeuchi, Kentaro Inui, Nao Takeuchi and Atsushi Fujita

9:25–9:50 *Collaborative Work on Indonesian WordNet through Asian WordNet (AWN)*
Chairil Hakim, Budiono Budiono and Hammam Riza

9:50–10:15 *Considerations on Automatic Mapping Large-Scale Heterogeneous Language Resources: Sejong Semantic Classes and KorLex*
Heum Park, Ae sun Yoon, Woo Chul Park and Hyuk-Chul Kwon

10:15–10:40 *Sequential Tagging of Semantic Roles on Chinese FrameNet*
Jihong LI, Ruibo WANG and Yahui GAO

10:40–11:00 Coffee Break

Semantics, Sentiment and Opinion

11:00–11:25 *Augmenting a Bilingual Lexicon with Information for Word Translation Disambiguation*
Takashi Tsunakawa and Hiroyuki Kaji

11:25–11:50 *Construction of bilingual multimodal corpora of referring expressions in collaborative problem solving*
Takenobu Tokunaga, Ryu Iida, Masaaki Yasuhara, Asuka Terai, David Morris and Anja Belz

11:50–12:15 *Labeling Emotion in Bengali Blog Corpus A Fine Grained Tagging at Sentence Level*
Dipankar Das and Sivaji Bandyopadhyay

12:15–12:40 *SentiWordNet for Indian Languages*
Amitava Das and Sivaji Bandyopadhyay

12:40–14:10 Lunch Break

Saturday August 21, 2010 (continued)

Opinion and Information Retrival

14:10–14:35 *Constructing Thai Opinion Mining Resource: A Case Study on Hotel Reviews*
Choochart Haruechaiyasak, Alisa Kongthon, Pornpimon Palingoon and Chatchawal Sangkeettrakarn

14:35–15:00 *The Annotation of Event Schema in Chinese*
Hongjian Zou, Erhong Yang, Yan Gao and Qingqing Zeng

15:00–15:25 *Query Expansion for Khmer Information Retrieval*
Channa Van and Wataru Kameyama

15:25–16:00 Coffee Break

Text Corpus

16:00–16:25 *Word Segmentation for Urdu OCR System*
Misbah Akram and Sarmad Hussain

16:25–16:50 *Dzongkha Word Segmentation*
Sithar Norbu, Pema Choejey, Tenzin Dendup, Sarmad Hussain and Ahmed Muaz

16:50–17:15 *Building NLP resources for Dzongkha: A Tagset and A Tagged Corpus*
Chungku Chungku, Jurmey Rabgay and Gertrud Faaß

17:15–17:40 *Unaccusative/Unergative Distinction in Turkish: A Connectionist Approach*
Cengiz Acarturk and Deniz Zeyrek

Sunday August 22, 2010

Grammars and Parsing

9:00–9:25 *A Preliminary Work on Hindi Causatives*
Rafiya Begum and Dipti Misra Sharma

9:25–9:50 *A Supervised Learning based Chunking in Thai using Categorial Grammar*
Thepchai Supnithi, Chanon Onman, Peerachet Porkaew, Taneth Ruangrajitpakorn, Kanokorn Trakultaweekoon and Asanee Kawtrakul

9:50–10:15 *A hybrid approach to Urdu verb phrase chunking*
Wajid Ali and Sarmad Hussain

10:15–10:40 *Development of the Korean Resource Grammar: Towards Grammar Customization*
Sanghoun Song, Jong-Bok Kim, Francis Bond and Jaehyung Yang

10:40–11:00 Coffee Break

Grammars and Applications

11:00–11:25 *An Open Source Urdu Resource Grammar*
Shafqat Mumtaz Virk, Muhammad Humayoun and Aarne Ranta

11:25–11:50 *A Current Status of Thai Categorial Grammars and Their Applications*
Taneth Ruangrajitpakorn and Thepchai Supnithi

11:50–12:15 *Chained Machine Translation Using Morphemes as Pivot Language*
Li Wen, Chen Lei, Han Wudabala and Li Miao

12:15 Concluding Remarks and Discussion

A Thesaurus of Predicate-Argument Structure for Japanese Verbs to Deal with Granularity of Verb Meanings

Koichi Takeuchi
Okayama University /
koichi@cl.cs.
okayama-u.ac.jp

Kentaro Inui
Tohoku University /
inui@ecei.
tohoku.ac.jp

Nao Takeuchi
Free Language
Analyst

Atsushi Fujita
Future University Hakodate /
fujita@fun.ac.jp

Abstract

In this paper we propose a framework of verb semantic description in order to organize different granularity of similarity between verbs. Since verb meanings highly depend on their arguments we propose a verb thesaurus on the basis of possible shared meanings with predicate-argument structure. Motivations of this work are to (1) construct a practical lexicon for dealing with alternations, paraphrases and entailment relations between predicates, and (2) provide a basic database for statistical learning system as well as a theoretical lexicon study such as Generative Lexicon and Lexical Conceptual Structure. One of the characteristics of our description is that we assume several granularities of semantic classes to characterize verb meanings. The thesaurus form allows us to provide several granularities of shared meanings; thus, this gives us a further revision for applying more detailed analyses of verb meanings.

1 Introduction

In natural language processing, to deal with similarities/differences between verbs is essential not only for paraphrase but also textual entailment and QA system which are expected to extract more valuable facts from massively large texts such as the Web. For example, in the QA system, assuming that the body text says “He lent her a bicycle”, the answer of the question “He gave her a bicycle?” should be “No”, however the answer of “She rented the bicycle?” should be “Yes”. Thus constructing database of verb similarities/differences enables us to deal with detailed paraphrase/non-paraphrase relations in NLP.

From the view of the current language resource, how the shared/different meanings of “He lent her a bicycle” and “He gave her a bicycle” can be described? The shared meaning of *lend* and *give* in the above sentences is that they are categorized to *Giving Verbs*, as in Levin’s English Verb Classes and Alternations (EVCA) (Levin, 1993), while the different meaning will be that *lend* does not imply ownership of the theme, i.e., *a bicycle*. One of the problematic issues with describing shared meaning among verbs is that semantic classes such as *Giving Verbs* should be dependent on the granularity of meanings we assumed. For example, the meaning of *lend* and *give* in the above sentences is not categorized into the same Frame in FrameNet (Baker et al., 1998). The reason for this different categorization can be considered to be that the granularity of the semantic class of *Giving Verbs* is larger than that of the *Giving* Frame in FrameNet[1]. From the view of natural language processing, especially dealing the with propositional meaning of verbs, all of the above classes, i.e., the wider class of *Giving Verbs* containing *lend* and *give* as well as the narrower class of *Giving* Frame containing *give* and *donate*, are needed. Therefore, in this work, in order to describe verb meanings with several granularities of semantic classes, a thesaurus form is adopted for our verb dictionary.

Based on the background, this paper presents a thesaurus of predicate-argument structure for verbs on the basis of a lexical decompositional framework such as Lexical Conceptual Structure (Jackendoff, 1990); thus our

[1] We agree with the concept of Frame and FrameElements in FrameNet but what we propose in this paper is the necessity for granularities of Frames and FrameElements.

Proceedings of the 8th Workshop on Asian Language Resources, pages 1–8,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

proposed thesaurus can deal with argument structure level alternations such as causative, transitive/intransitive, stative. Besides, taking a thesaurus form enables us to deal with shared/differenciate meaning of verbs with consistency, e.g., a verb class node of “lend” and “rent” can be described in the detailed layer of the node “give”.

We constructed this thesaurus on Japanese verbs and the current status of the verb thesaurus is this: we have analyzed 7,473 verb meanings (4,425 verbs) and organized the semantic classes in a five-layer thesaurus with 71 semantic roles types. Below, we describe background issues, basic design issues, what kind of problems remain, limitations and perspectives of applications.

2 Existing Lexical Resources and Drawbacks

2.1 Lexical Resources in English

From the view of previous lexical databases In English, several well-considered lexical databases are available, e.g., EVCA, Dorr’s LCS (Dorr, 1997), FrameNet, WordNet (Fellbaum, 1998), VerbNet (Kipper-Schuler, 2005) and PropBank (Palmer et al., 2005). Besides there is the research project (Pustejovsky and Meyers, 2005) to find general descriptional framework of predicate argument structure by merging several lexical databases such as PropBank, NomBank, TimeBank and PennDiscouse Treebank.

Our approach corresponds partly to each lexical database, (i.e., FrameNet’s Frame and FrameElements correspond to our verb class and semantic role labels, and the way to organize verb similarity classes with thesaurus corresponds with WordNet’s synset), but is not exactly the same; namely, there is no lexical database describing several granularities of semantic classes between verbs with arguments. Of course, since the above English lexical databases have links with each other, it is possible to produce a verb dictionary with several granularities of semantic classes with arguments. However, the basic categories of classifying verbs would be little different due to the different background theory of each English lexical database; it must be not easy to add another level of semantic granularity with keeping consistency for all the lexical databases; thus, thesaurus form is needed to be a core form for describing verb meanings[2].

2.2 Lexical Resources in Japanese

In previous studies, several Japanese lexicons were published: IPAL (IPA, 1986) focuses on morpho-syntactic classes but IPAL is small[3]. EDR (Jap, 1995) consists of a large-scale lexicon and corpus (See Section 3.4). EDR is a well-considered and wide coverage dictionary focusing on translation between Japanese and English, but EDR’s semantic classes were not designed with linguistically-motivated lexical relations between verbs, e.g., alternations, causative, transitive, and detransitive relations between verbs. We believe these relations must be key for dealing with paraphrase in NLP.

Recently Japanese FrameNet (Ohara et al., 2006) and Japanese WordNet (Bond et al., 2008) are proposed. Japanese FrameNet currently published only less than 100 verbs[4]. Besides Japanese WordNet contains 87000 words and 46000 synsets, however, there are three major difficulty of dealing with paraphrase relations between verbs: (1) there is no argument information; (2) existing many similar synsets force us to solve fine disambiguation between verbs when we map a verb in a sentence to WordNet; (3) the basic verbs of Japanese (i.e., highly ambiguous verbs) are wrongly assigned to unrelated synsets because they are constructed by translation from English to Japanese.

[2] As Kipper (Kipper-Schuler, 2005) showed in their examples mapping between VerbNet and WordNet verb senses, most of the mappings are many-to-many relations; this indicates that some two verbs grouped in a same semantic type in VerbNet can be categorized into different synsets in WordNet. Since WordNet does not have argument structure nor syntactic information, we cannot purchase what is the different features for between the synsets.

[3] It contains 861 verbs and 136 adjectives.

[4] We are supplying our database to Japanese FrameNet project.

3 Thesaurus of Predicate-Argument Structure

The proposed thesaurus of predicate-argument structure can deal with several levels of verb classes on the basis of granularity of defined verb meaning. In the thesaurus we incorporate LCS-based semantic description for each verb class that can provide several argument structure such as construction grammar (Goldberg, 1995). This must be high advantage to describe the different factors from the view of not only syntactic functions but also internal semantic relations. Thus this characteristics of the proposed thesaurus can be powerful framework for calculating similarity and difference between verb senses. In the following sections we explain the total design of thesaurus and the details.

3.1 Design of Thesaurus

The proposed thesaurus consists of hierarchy of verb classes we assumed. A verb class, which is a conceptual class, has verbs with a shared meaning. A parent verb class includes concepts of subordinate verb class; thus a subordinate verb class is a concretization of the parent verb class. A verb class has a semantic description that is a kind of semantic skeleton inspired from lexical conceptual structure (Jackendoff, 1990; Kageyama, 1996; Dorr, 1997). Thus a semantic description in a verb class describes core semantic relations between arguments and shadow arguments of a shared meaning of the verb class. Since verb can be polysemous, each verb sense is designated with example sentences. Verb senses with a shared meaning are assigned to a verb class. Every example sentence is analyzed into their arguments and semantic role types; and then their arguments are linked to variables in semantic description of verb class. This indicates that one semantic description in a verb class can provide several argument structure on the basis of syntactic structure. This architecture is related to construction grammar.

Here we explain this structure using verbs such as *rent, lend, give, hire, borrow, lease*. We assume that each verb sense we focus on here is designated by example sentences, e.g., "Mother gives a book to her child", "Kazuko rents a bicycle from her friend", and "Taro lend a car to his friend". As Figure 1 shows that all of the above verb senses are involved in the verb class *Moving of One's Possession* [5]. The semantic description, which expresses core meaning of the verb class *Moving of One's Possession* is

```
([Agent] CAUSE)
BECOME [Theme] BE AT [Goal].
```

Where the brackets [] denote variables that can be filled with arguments in example sentences. Likewise parentheses () denote occasional factor. "Agent" and "Theme" are semantic role labels that can be annotated to all example sentences. Figure 1 shows that the children of the verb class *Moving of One's Possession* are the two verb classes *Moving of One's Possession/Renting* and *Moving of One's Possession/Lending*. In the *Renting* class, *rent, hire and borrow* are there, while in the *Lending* class, *lend and lease* exist. Both of the semantic descriptions in the children verb classes are more detailed ones than the parent's description.

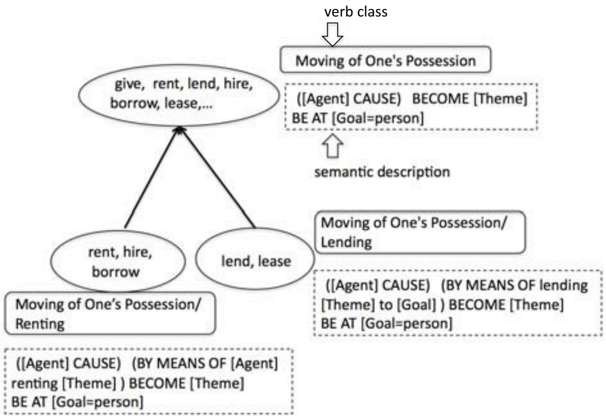

verb class
Moving of One's Possession
give, rent, lend, hire, borrow, lease,...
([Agent] CAUSE) BECOME [Theme] BE AT [Goal=person]
semantic description
Moving of One's Possession/ Lending
rent, hire, borrow
lend, lease
([Agent] CAUSE) (BY MEANS OF lending [Theme] to [Goal]) BECOME [Theme] BE AT [Goal=person]
Moving of One's Possession/ Renting
([Agent] CAUSE) (BY MEANS OF [Agent] renting [Theme]) BECOME [Theme] BE AT [Goal=person]

Figure 1: Example of verb classes and their semantic descriptions in parent-children.

A semantic description in the *Renting* class, i.e.,

```
([Agent] CAUSE)
```

[5] The name of a verb class consists of hierarchy of thesaurus; and Figure 1 shows abbreviated verb class name. Full length of the verb class name is *Change of State/Change of Position (Physical)/Moving of One's Possession*.

```
(BY MEANS OF [Agent]  renting [Theme])
BECOME [Theme]  BE AT [Agent],
```

describes semantic relations between “Agent” and “Theme”. Since semantic role labels are annotated to all of the example sentences, the variables in the semantic description can be linked to practical arguments in example sentences via semantic role labels (See Figure 2).

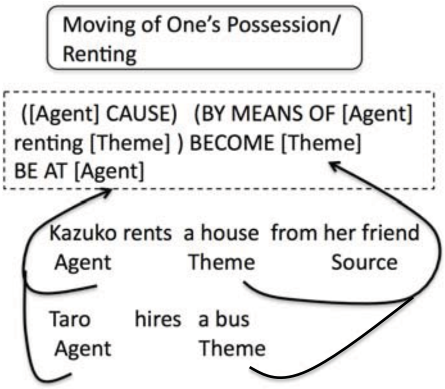

Moving of One's Possession/ Renting
([Agent] CAUSE) (BY MEANS OF [Agent] renting [Theme]) BECOME [Theme] BE AT [Agent]
Kazuko rents a house from her friend
Agent Theme Source
Taro hires a bus
Agent Theme

Figure 2: Linking between semantic description and example sentences.

3.2 Construction of Verb Class Hierarchy

To organize hierarchical semantic verb class, we take a top down and a bottom up approaches. As for a bottom up approach, we use verb senses defined by a dictionary as the most fine-grained meaning; and then we group verbs that can be considered to share some meaning. As for a dictionary, we use the Lexeed database (Fujita et al., 2006), which consists of more than 20,000 verbs with explanations of word sense and example sentences.

As a top down approach, we take three semantic classes: *State*, *Change of State*, and *Activity* as top level semantic classes of the thesaurus according to Vendler’s aspectual analysis (Vendler, 1967) (See Figure 4). This is because the above three classes can be useful for dealing with the propositional, especially, resultative aspect of verbs. For example “He threw a ball” can be an *Activity* and have no special result; but “He broke the door” can be a *Change of State* and then we can imagine a result, i.e., *broken door*. When other verb senses can express the same results, e.g., “He destroyed the door,” we would like to regard them as having the same meaning.

We define verb classes in intermediate hierarchy by grouping verb sense on the basis of aspectual category (i.e., action, state, change of state), argument type (i.e., physical, mental, information), and more detailed aspects depending on aspectual category. For example, *walk the country*, *travel all over Europe* and *get up the stairs* can be considered to be in the *Move on Path* class.

Verb class is essential for dealing with verb meanings as synsets in WordNet. Even if we had given an incorrect class name, the thesaurus will work well if the whole hierarchy keeps is-a relation, namely, the hierarchy does not contain any multiple inheritance.

The most fine-grained verb class before individual verb sense is a little wider than alternations. Currently, for the fine-grained verb class, we are organizing what kind of differentiated classes can be assumed (e.g., manner, background, presupposition, and etc.).

3.3 Semantic Role Labels

The aim of describing arguments of a target verb sense is (1) to link the same role arguments in a related verb sense and (2) to provide disambiguated information for mapping a surface expression to a verb sense. The Lexeed database provides a representative sentence for each word sense. The sentence is simple, without adjunctive elements such as unessential time, location or method. Thus, a sentence is broken down into subject and object, and semantic role labels are annotated to them (Figure 3).

ex.:	nihon-ga	shigen-wo	yunyuu-suru
trans.:	Japan (NOM)	resouces (ACC)	import
AS:	Agent	Theme	

Figure 3: An example of semantic role label.

Of course, only one representative sentence would miss some essential arguments; also, we

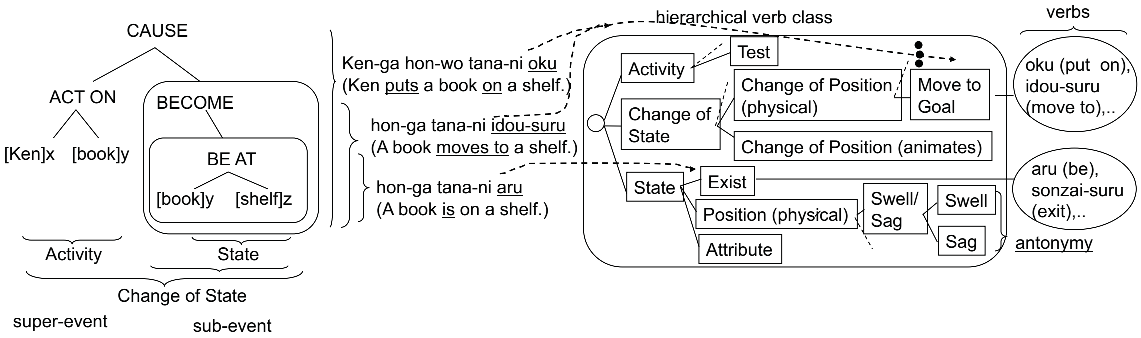


Figure 4: Thesaurus and corresponding lexical decomposition.

do not know how many arguments are enough. This can be solved by adding examples[6]; however, we consider the semantic role labels of each representative sentence in a verb class as an example of assumed argument structure to a verb class. That is to say, we regard a verb class as a concept of event and suppose it to be a fixed argument frame for each verb class. The argument frame is described as compositional relations.

The principal function of the semantic role label name is to link arguments in a verb class. One exception is the *Agent* label. This can be a marker discriminating transitive and intransitive verbs. Since the semantic class of the thesaurus focuses on *Change of State*, transitive alternation cases such as “The president expands the business” and “ The business expands” can be categorized into the same verb class. Then, these two examples are differentiated by the *Agent* label.

3.4 Compositional Semantic Description

As described in Section 3.1, we incorporate compositional semantic structure to each verb class to describe syntactically motivated lexical semantic relations and entailment meanings that will expand the thesaurus. The benefit of compositional style is to link entailed meanings by means of compositional manner. As an example of entailment, Figure 5 shows that a verb class *Move to Goal* entails *Theme* to be *Goal*, and this corresponds to a verb class *Exist*.

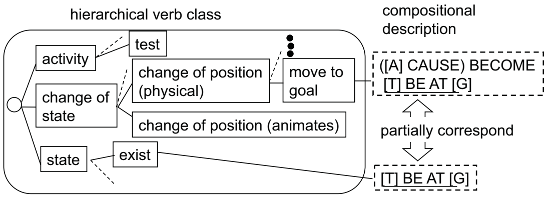

hierarchical verb class
activity
test
change of state
change of position (physical)
move to goal
change of position (animates)
state
exist
compositional description
([A] CAUSE) BECOME [T] BE AT [G]
partially correspond
[T] BE AT [G]

Figure 5: Compositional semantic description.

In this verb thesaurus, being different from previous LCS studies, we try to ensure the compositional semantic description as much as possible by means of linking each sub-event structure to both a semantic class and example sentences. Therefore, we believe that our verb thesaurus can provide a basic example data base for LCS study.

3.5 Intrinsic Evaluation on Coverage

We did manual evaluation that how the proposed verb thesaurus covers verb meanings in news articles. The results on Japanese new corpus show that the coverage of verbs is 84.32% (1825/2195) in 1000 sentences randomly sampled from Japanese news articles[7]. Besides we take 200 sentences and check whether the verb meanings in the sentences can correspond to verb meaning in our thesaurus. The result shows that our thesaurus meaning covers 99.5% (199 verb meanings/200 verb meanings) of 200

[6] We are currently constructing an SRL annotated corpus.

[7] Mainichi news article in 2003.

verbs[8].

4 Discussions

4.1 Comparison with Existing Resources

Table 1 and Table 2 show a comparison of statistical characteristics with existing resources. In the tables, WN and Exp denote the number of word meanings and example sentences, respectively. Also, SRL denotes the number corresponding to semantic role label.

Looking at number of concepts, our Thesaurus has 709 types of concepts (verb classes) which is similar to FrameNet and more than VerbNet. This seems to be natural because FrameNet is also language resource constructed on argument structure. Thanks to our thesaurus format, if we need more fine grained concepts, we can expand our thesaurus by adding concepts as new nodes at the lowest layer in the hierarchy. While at the number of SRL, FrameNet has much more types than our thesaurus, and in the other resources VerbNet and EDR the number of SRL is less than our thesaurus. This comes from the different design issue of semantic role labels. In FrameNet they try to differentiate argument types on the basis of the assumed concept, i.e., Frame. In contrast with FrameNet we try to merge the same type of meaning in arguments. VerbNet and EDR also defined abstracted SRL; The difference between their resources and our thesaurus is that our SRLs are defined taking into account what kind of roles in the core concept i.e., verb class; while SRLs in VerbNet and EDR are not dependent on verb's class.

Table 2 shows that our thesaurus does not have large number in registered words and examples comparing to EDR and JWordNet. As we stated in Section 3.5, the coverage of our verb class to newspaper articles are high, but we try to add examples by constructing annotated Japanese corpus of SRL and verb class.

Table 1: Comparing to English resources

	FrameNet	WordNet	VerbNet
Concepts	825 (Frame) (Ver 1.3) 2007	N/A (Synset) (Ver 3.0) 2006	237 (class) (Ver 2.2) 2006
Words	6100	155287	3819
WM	N/A	117659	5257
Exp	13500	48349	N/A
SRL	746	N/A	23
POS	V,N,A,Ad	V,N,A,Ad	V
Lang	E,O	E,O	E

Table 2: Comparing to Japanese resources

	EDR	JWordNet	Our Thesaurus
Concepts	430000 (class) (Ver 3.0) 2003	N/A (Ver 0.92) 2009	709 2008
Words	410000	92241	4425
WM	270000	56741	7473
Exp	200000	48276	7473
SRL	28	N/A	71
POS	all	V,N,A,Ad	V
Lang	EJ	E,J	J

4.2 Limitations of Developed Thesaurus

One of the difficulties of annotating the semantic class of word sense is that a word sense can be considered as several semantic classes. The proposed verb thesaurus can deal with multiple semantic classes for a verb sense by adding them into several nodes in the thesaurus. However, this does not seem to be the correct approach. For example, what kind of *Change of State* semantic class can be considered in the following sentence?

a. *He took on a passenger*.

Assuming that *passenger* is *Theme*, *Move to goal* could be possible when we regard the vehicle[9] as *Goal*. In another semantic class, *Change State of Container* could be possible when we regard the vehicle as a container. Currently, all of the verb senses are linked to only one semantic class that can be considered as the most related semantic class.

[8] This evaluation is done by one person. Of course we need to check this by several persons and take inter-annotator agreement.

[9] *Vehicle* does not appear in the surface expression but *vehicle* can exist. We currently describe the shadow argument in the compositional description, but it would be hard to prove the existence of a shadow argument.

From the user side, i.e., dealing with the propositional meaning of the sentence (**a.**), various meanings should be estimated. Consider the following sentence:

b. *Thus, we were packed.*

As the semantic class of the sentence (**a.**) *Change State of Container* could better explain why they are packed in the sentence (**b.**)

The other related issue is how we describe the scope, e.g.,

c. *He is hospitalized.*

If we take the meaning as a simple manner, *Move to Goal* can be a semantic class. This can be correct from the view of annotation, but we can guess *he cannot work* or *he will have a tough time* as following events. FrameNet seems to be able to deal with this by means of a special type of linking between Frames.

Consequently, we think the above issues of semantic class should depend on the application side's demands. Since we do not know all of the requirements of NLP applications currently, then it must be sufficient to provide an expandable descriptional framework of linguistically motivated lexical semantics.

4.3 Remaining and Conceivable Ways of Extension

One of the aims of the proposed dictionary is to identify the sentences that have the same meanings among different expressions. One of the challenging paraphrase relations is that the sentences expressed from the different view points. Given the buying and selling in Figure 6, a human can understand that both sentences denote almost the same event from different points of view. This indicates that the sentences made by humans usually contain the point of view of the speaker. This is similar to a camera, and we need to normalize the expressions as to their original meaning.

We consider that NLP application researchers need to relate these expressions. Logically, if we know "buy" and "sell" have shared meanings of *giving and taking things*, we can describe their relations with "or" in logical form. Therefore, finding and describing these verb relations will be essential for dealing with propositional meanings of a sentence.

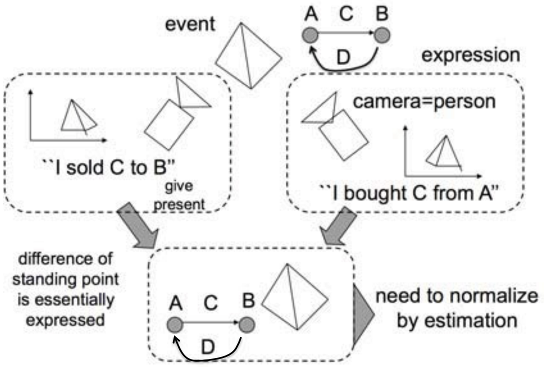


Figure 6: Requirement of normalization to deal with different expressions from the different views.

For further view of application, event matching to find a similar situation in Web documents is supposed to be a practical and useful application. Assuming that a user is confronted with the fact that wireless LAN in the user's PC does not work, and the user wants to search for documents that provide a solution, the problem is that expressions of situations must be different from the views of individual writers, e.g., "wireless LAN did not work" or "wireless LAN was disconnected". How can we find the same meaning in these expressions, and how can we extract the answers by finding the same situation from FAQ documents? To solve this, a lexical database describing verb relations between "go wrong" and "disconnect" must be the base for estimating how the expressions can be similar. Therefore, constructing a lexicon can be worthwhile for developing NLP applications.

5 Conclusion

In this paper, we presented a framework of a verb dictionary in order to describe shared meaning as well as to differentiate meaning between verbs from the viewpoint of relating eventual expressions of NLP. One of the characteristics is that we describe verb relations on the basis of several semantic granularities using a thesaurus form with argument structure. Semantic granularity is the basis for how we categorize (or recognize which semantic class relates to a verb meaning). Also, we ensure functions and limitations of semantic classes and argument structure from the viewpoint of dealing with paraphrases. That is, required semantic classes will be highly dependent on applications; thus, the framework of the verb-sense dictionary should have expandability. The proposed verb thesaurus can take several semantic granularities; therefore, we hope the verb thesaurus will be applicable to NLP's task[10].

In future work, we will continue to organize differentiated semantic classes between verbs and develop a system to identify the same event descriptions.

Acknowledgments

We would like to thank NTT Communication Research Laboratory for allowing us to use the Lexeed database, and Professor Koyama for many useful comments.

References

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 86–90.

Bond, Francis, Hitoshi Isahara, Kyoko Kanzaki, and Kiyotaka Uchimoto. 2008. Construction of Japanese WordNet from Multi-lingual WordNet. In *Proceedings of the 14th Annual Meeting of Japanese Natural Language Processing*, pages 853–856.

Dorr, Bonnie J. 1997. Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation. *Machine Translation*, 12(4):271–325.

Fellbaum, Chistiane. 1998. *WordNet an Electronic Lexical Database*. MIT Press.

Fujita, Sanae, Takaaki Tanaka, Fransis Bond, and Hiromi Nakaiwa. 2006. An implemented description of japanese: The lexeed dictionary and the hinoki treebank. In *COLING/ACL06 Interactive Presentation Sessions*, pages 65–68.

Goldberg, Adele E. 1995. *Constructions*. The University of Chicago Press.

IPA: Information-Technology Promotion Agency, Japan, 1986. *IPA Lexicon of the Japanese Language for Computers*.

Jackendoff, Ray. 1990. *Semantic Structures*. MIT Press.

Japan Electronic Dictionary Research Institute, Ltd, 1995. *EDR: Electric Dictionary the Second Edition*.

Kageyama, Taro. 1996. *Verb Semantics*. Kurosio Publishers. (In Japanese).

Kipper-Schuler, K. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, PhD Thesis, University of Pennsylvania.

Levin, Beth. 1993. *English Verb Classes and Alternations*. University of Chicago Press.

Ohara, Kyoko Hirose, Seiko Fujii, Toshio Ohori, Ryoko Suzuki, Hiroaki Saito, and Shun Ishizaki. 2006. Frame-based contrastive lexical semantics and japanese framenet: The case of risk and kakeru. In *Proceeding of the Fourth International Conference on Construction Grammar*. http://jfn.st.hc.keio.ac.jp/ja/publications.html.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Pustejovsky, J. and Martha P.and A. Meyers. 2005. Merging propbank, nombank, timebank, penn discourse treebank and coreference. In *Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*, pages 5–12.

Vendler, Zeno. 1967. *Linguistics in Philosophy*. Cornell University Press.

[10] The proposed verb thesaurus is available at: http://cl.cs.okayama-u.ac.jp/rsc/data/. (in Japanese).

Collaborative Work on Indonesian WordNet through Asian WordNet (AWN)

Hammam Riza

Agency for the Assessment and Application of Technology (BPPT), Indonesia
hammam@iptek.net.id

Budiono

Agency for the Assessment and Application of Technology (BPPT), Indonesia
budi@iptek.net.id

Chairil Hakim

Agency for the Assessment and Application of Technology (BPPT), Indonesia
chairil@iptek.net.id

Abstract

This paper describes collaborative work on developing Indonesian WordNet in the AsianWordNet (AWN). We will describe the method to develop for collaborative editing to review and complete the translation of synset. This paper aims to create linkage among Asian languages by adopting the concept of semantic relations and synset expressed in WordNet.

1 Introduction

Multilingual lexicons is of foremost importance for intercultural collaboration to take place, as multilingual lexicons are several multilingual application such as Machine Translation, terminology, multilingual computing.

WordNet is the resource used to identify shallow semantic features that can be attached to lexical units. The original WordNet is English WordNet proposed and developed at Princeton University WordNet (PWN) by using bilingual dictionary.

In the era of globalization, communication among languages becomes much more important. People has been hoping that natural language processing and speech processing. We can assist in smoothening the communication among people with different languages. However, especially for Indonesian language, there were only few researches in the past.

The Princeton WordNet is one of the semantically English lexical banks containing semantic relationships between words. Concept mapping is a process of organizing to forming meaningful relationships between them.

The goal of Indonesian AWN database management system is to share a multilingual lexical database of Indonesian language which are structured along the same lines as the AWN.

AWN is the result of the collaborative effort in creating an interconnected Wordnet for Asian languages. AWN provides a free and public platform for building and sharing among AWN. The distributed database system and user-friendly tools have been developed for user. AWN is easy to build and share.

This paper describes manual interpretation method of Indonesian for AWN. Based on web services architecture focusing on the particular cross-lingual distributed. We use collective intelligence approach to build this English equivalent. In this sequel, in section 2 the collaborations builders works on web interface at www.asianwordnet.org. In section 3, Interpretation of Indonesian AWN, short description of progress of English – Indonesian translation and the obstacle of translation.

2 Collaborative AWN

WordNet covers the vast majority of nouns, verbs, adjectives and adverbs from English language. The words are organized in synonym sets called synset. Each synset represents a concept includes an impressive number of semantic relations defined across concepts.

The information encoded in WordNet is used in several stages in the parsing process. For instance, attribute relations, adjective/adverb classifications, and others are semantic features extracted from WordNet and stored together with the words, so that they can be directly used by the semantic parser.

Proceedings of the 8th Workshop on Asian Language Resources, pages 9–13,
Beijing, China, 21-22 August 2010.

To build language WordNet there are two main of discussion; the merge approach and the expand approach. The merge approach is to build the taxonomies of the language (synset) using English equivalent words from bilingual dictionaries. The expand approach is to map translate local words the bilingual dictionaries. This approach show the relation between senses. The system manages the synset assignment according to the preferred score obtained from the revision process. For the result, the community will be accomplish into original form of WordNet database. The synset can generate a cross language result.

AWN also introduce a web-based collaborative workbench, for revising the result of synset assignment and provide a framework to create AWN via linkage through PWN synset. AWN enables to connect and collaborate among individual intelligence in order accomplish a text files.

At present, there are ten Asian language in the community. The amount of the translated synsets had been increased. Many language have collaboration in AWN.

- Agency for the Assessment and Application of Technology (BPPT), Indonesia
- National Institute of Information and Communications Technology (NICT), Japan
- Thai Computational Linguistics Laboratory (TCL), Thailand
- National Electronics and Computer Technology Center (NECTEC), Thailand
- National University of Mongolia (NUM), Mongolia
- Myanmar Computer Federation (MCF), Myanmar
- National Authority of Science and Technology (NAST), Lao PDR
- Madan Puraskar Pustakalaya (MPP), Nepal
- University of Colombo School of Computing (UCSC), SriLanka
- Vietnamese Academy of Science and Technology (VAST), Vietnam

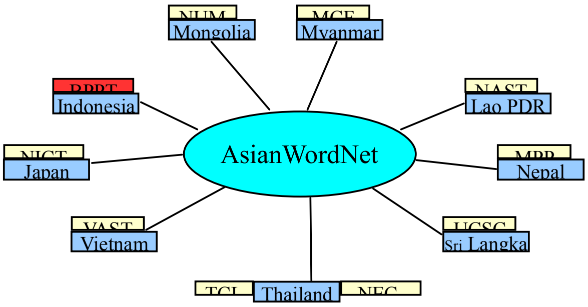


Fig 1. Collaboration on Asian WordNet

3 Interpretation of Indonesian AWN

Indonesian WordNet have been used as a general-purpose translation. Our approach was to generate the query for the web services engine in English and then to translate every key element of the query (topic, focus, keywords) into Indonesian without modifying the query. The dictionary is distinguished by set of entry word characteristic, clear definitions, its guidance on usage. All dictionary information for entries is structured such as entry word, multiple word entries, notes, contemporary definitions, derivations, example sentence, idioms, etc. All dictionary are implemented as text-files and as linguistic databases connected to Indonesian AWN. The set of language tags consists of part of speech, case, gender, number, tense, person, voice, aspect, mood, form, type, reflexive, animation.

3.1 Progress English – Indonesian

Indonesian WordNet is used Word Net Management System (WNMS) tools developed by AsianWordNet to create web services among Asia languages based on Princeton WordNet® version 3.0, Co-operation by TCL and BPPT establish on October 2007.

As presented above, we follow the merge to create and share the Indonesian WordNet by translating the each synonym translation. We expand an appropriate synset to a lexical entry by considering its English equivalent.

We plan to have reliable process to create and share Indonesian WordNet in AWN. We classify this work into four person AWN translators to participate in the project of Indonesian AWN.

Each person was given a target translator in a month should reach at least 3000 sense so that the total achievement 12000 senses in a month. From 117.659 senses that there is expected to be completed within 10 months. On the process of mapping, a unique word will be generated for every lexical entry which contain. The grammatical dictionaries contain normalized entry word with hyphenation paradigm plus grammatical tags.

	Assignment			TOTAL
	March	April	May	sense
Noun	10560	14199	16832	82115
verb	6444	6444	6499	13767
Adjective	1392	1392	1936	18156
Adverb	481	481	488	3621
Total	18877	22516	25755	117659

Table 1. Statistic of synsets

In the evaluation of our approach for synset assignment, we selected randomly sense from the the result of synset assignment to English – Indonesian dictionary for manually checking. The random set cover all types of part-of-speech. With the best information of English equivalents marked with CS=5. The word entry must be translated into the appropriate words by meaning explanation.

Table 1. presents total assignment translated words into Indonesian for the second third month. Following the manual to translate the English AWN to Indonesian, we resulted the progress of AWN at this time.

We start to translate or edit from some group of base type in "By Category". These base types are based on categories from PWN. There is only 21.89% (approved 25,755 of 117,659 senses) of the total number of the synsets that were able to be assigned to lexical entry in the Indonesian – English Dictionary.

3.2 Obstacle of Indonesian Translation

Wordnet has unique meaning of word which is presented in synonym set. Each synset has glossary which defines concept its representation. For examples word car, auto, automobile, and motorcar has one synset.

An automatic compilation of dictionary in AWN have a translational issues. There are many cases in explanation sense. One word in English will be translated into a lot of Indonesian words, glossary can be express more than one Indonesian word (Ex. 1).

One of the main obstacles in studying the absorption of English words in Indonesian words, is the fact that the original form of some words that have been removed due to Indonesian reform process, in which some words have been through an artificial process. There is no special character in Indonesian word, especially in technical word, so that means essentially the same as the English word (Ex. 2).

Ex. 1. **time frame**

POS	noun time
synset	time_frame
gloss	a time period during which something occurs or is expected to occur; an agreement can be reached in a reasonably short time frame"
Indonesian	jangka waktu, selang waktu

Ex. 2. **resolution**

POS	noun phenomenon
synset	resolution
gloss	(computer science) the number of pixels per square inch on a computer generated display; the greater the resolution, the better the picture
Indonesian	resolusi

Using definitions from the WordNet electronic lexical database. A major problem in natural language processing is that of lexical ambiguity, be it syntactic. Each single words must be container for some part of the linguistic knowledge need to ambiguous wordnet sense. Therefore, not only a single heuristic translate Indonesian words. The WordNet defined in some semantic relations, this categories using lexicographer file and glossary definitions relations are assigned to weight in the range. WordNet hierarchy for the first sense of the word "*empty*" there are 10 synset words (I take three of ten) that are related to the meaning are the following in (Ex. 3.)

Three concepts recur in WordNet literature that entail a certain amount of ambiguity : terminological distance, semantic distance and conceptual distance. Terminological distance, by contrast, often appears to refer to the suitability of the word selected to express a given concept. Semantic distance is understood to mean the contextual factor of precision in meaning. And the conceptual distance between words, in which have relations proved.

Ex. 3.	**empty**	
	POS	noun art
	synset	empty
	gloss	a container that has been emptied; "return all empties to the store"
	Indonesian	hampa
	empty	
	POS	verb change
	synset	empty, discharge
	gloss	become empty or void of its content; "The room emptied"
	Indonesian	mengosongkan
	empty	
	POS	adjectives all
	synset	empty
	gloss	emptied of emotion; "after the violent argument he felt empty"
	Indonesian	kosong, penat

Disambiguation is unquestionably the most abundant and varied application. It precision and relevance in response to a query inconsistencies. Schematically the semantic disambiguation are selected in the glossaries of each noun, verb, and adjectives and its subordinates.

WordNet information, whose objective is to build designs for the association between sentences and coherence relations as well as to find lexical characteristics in coherence categories. WordNet became an ancillary tool for semantic ontology design geared to high quality information extraction from the web services.

A comparative analysis of trends in wordnet use :

1. Support for the design of grammatical categories designed to classify information by aspects and traits, but in particular to design and classify semantic ontologies.
2. Basis for the development of audio-visual and multi-media information retrieval systems.

4 Internet Viewer

The pilot internet service based on Wordnet 3.0 is published at http://id.asianwordnet.org.

5 Discussion and Conclusion

Any multilingual process such as cross-lingual information must involve resources and language pair. Language specific can be applied in parallel to achieve best result.

In this paper we describe manually sharing of Indonesian in the AWN by using dictionaries. AWN provides a free and public platform for building and sharing among AWN. We want continue the work defined learning the service matching system. Our future work on AWN will focuses in development platform WordNet and language technology web services.

Although AWN application are going steadily, the limitations are:

1. AWN designed for manual so authenticity can not be a reference.

2. Classification was performed manually, which means that the reasons and depth of classification may not be consistent.

References

Valenina Balkova, Andrey Suhonogov, Sergey Yablonsky. 2004. Rusian WordNet: From UML-notation to Internet/Infranet Database Implementation. In Porceedings of the Second International WordNet Conference (GWC 2004),

Riza, H., Budiono, Adiansya P., Henky M., (2008). I/ETS: Indonesian-English Machine Translation System using Collaborative P2P Corpus, Agency for the Assessment and Application of Technology (BPPT), Indonesia, University of North Texas.

Shi, Lei., Rada Mehalcea, (2005), Putting Pieces Together : Combining FrameNet, VerbNet, and WordNet for Robust Semantic Parsing

Thoongsup, S., Kergrit Robkop, Chumpol Mokarat, Tan Sinthurahat, (2009). Thai WordNet Construction. Thai Computational Linguistics Lab., Thailand

Virach Sornlertlamvanich., The 5th International Conference of the Global WordNet Association (GWC-2010), Mumbai, India , 31st Jan. - 4th Feb., 2010.

Fragos, Kostas, Yannis Maistros, Christos Skourlas, (2004). Word Sense Disambiguation using WORDNET relations. Dep. Of Computer Engineering NTUA, Greece.

www.asianwordnet.org

Considerations on Automatic Mapping Large-Scale Heterogeneous Language Resources: *Sejong Semantic Classes* and *KorLex*

Heum Park
Center for U-Port IT
Research and Education
Pusan National University
parheum2@empal.com

Aesun Yoon
LI Lab. Dept. of French
Pusan National University
asyoon@pusan.ac.kr

Woo Chul Park and Hyuk-Chul Kwon*
AI Lab Dept. of Computer Science
Pusan National University
hckwon@pusan.ac.kr

Abstract

This paper presents an automatic mapping method among large-scale heterogeneous language resources: *Sejong Semantic Classes* (SJSC) and *KorLex*. KorLex is a large-scale Korean WordNet, but it lacks specific syntactic & semantic information. *Sejong Electronic Dictionary* (SJD), of which semantic segmentation depends on SJSC, has much lower lexical coverage than KorLex, but shows refined syntactic & semantic information. The goal of this study is to build a rich language resource for improving Korean semantico-syntactic parsing technology. Therefore, we consider integration of them and propose automatic mapping method with three approaches: 1) Information of Monosemy/Polysemy of Word senses (IMPW), 2) Instances between Nouns of SJD and Word senses of KorLex (INW), and 3) Semantically Related words between Nouns of SJD and Synsets of KorLex (SRNS). We obtain good performance using combined three approaches: recall 0.837, precision 0.717, and F1 0.773.

1 Introduction

While remarkable progress has been made in Korean language engineering on morphological level during last two decades, syntactic and semantic processing has progressed more slowly. The syntactic and semantic processing requires 1) linguistically and formally well defined argument structures with the selectional restrictions of each argument, 2) large and semantically well segmented lexica, 3) most importantly, interrelationship between the argument structures and lexica. A couple of language resources have been developed or can be used for this end. *Sejong Electronic Dictionaries* (SJD) for nouns and predicates (verbs and adjectives) along with semantic classes (Hong 2007) were developed for syntactic and semantic analysis, but the current versions do not contain enough entries for concrete applications, and they show inconsistency problem. A Korean WordNet, named *KorLex* (Yoon & al, 2009), which was built on Princeton WordNet 2.0 (PWN) as its reference model, can provide means for shallow semantic processing but does not contain refined syntactic and semantic information specific to Korean language. *Korean Standard Dictionary* (STD) provides a large number of entries but it lacks systematic description and formal representation of word senses, like other traditional dictionaries for humans. Given these resources which were developed through long-term projects (5 – 10 years), integrating them should result in significant benefits to Korean syntactic and semantic processing.

The primary goal of our recent work including the work reported in this paper is to build a language resource, which will improve Korean semantico-syntactic parsing technology. We proceed by integrating the argument structures as provided by SJD, and the lexical-semantic hierarchy as provided by KorLex. SJD is a language resource, of which all word senses are labeled according to *Sejong semantic classes* (SJSC), and in which selectional re-

* Corresponding Author

Proceedings of the 8th Workshop on Asian Language Resources, pages 14–21,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

strictions are represented in SJSC as for the argument structures of predicates. KorLex is a large scale language resource, of which the lexical-semantic hierarchies and other language–independent semantic relations between synsets (synonym sets) share with those of PWN, and of which Korean language specific information comes from STD. The secondary goal is the improvement of three resources as a result of comparing and integrating them.

In this paper, we report on one of the operating steps toward to our goals. We linked each word sense of KorLex to that of STD by hand, when the former was built in our previous work (Yoon & al. 2009). All predicates in SJD were mapped to those of STD on word sense level by semi-automatic mapping (Yoon, 2010). Thus KorLexVerb and KorLexAdj have syntactico-semantic information on argument structures via this SJD - STD mapping. However, the selectional restrictions provided by SJD are not useful, if SJSC which represents the selectional restrictions in SJD is not linked to KorLex. We thus conduct two mapping methods between SJSC and upper nodes of KorLexNoun: 1) manual mapping by a PH.D in computational semantics (Bae & al. 2010), and 2) automatic mapping. This paper reports the latter. Reliable automatic mapping methods among heterogeneous language resources should be considered, since the manual mapping among large-scale resources is a very time and labor consuming job, and might lack consistency. Less clean resources are, much harder and more confusing manual mapping is.

In this paper, we propose an automatic mapping method of those two resources with three approaches to determine mapping candidate synsets of KorLex to a terminal node of SJSC: 1) using information of monosemy/polysemy of word senses, 2) using instances between nouns of SJD and word senses of KorLex, and 3) using semantically related words between nouns of SJD and word senses of KorLex. We compared the results of automatic mapping method with three approaches with those of manual mapping aforementioned.

In the following Section 2, we discuss related studies concerning language resources and automatic mapping methods of heterogeneous language resources. In Section 3, we introduce KorLex and SJD. In Section 4, we propose an automatic mapping method with three approaches from semantic classes of SJD to synsets of KorLex. In Section 5, we compare the results of automatic mapping with those of manual mapping. In Section 6, we draw conclusions and future works.

2 Related Works

Most existing mappings of heterogeneous language resources were conducted manually by language experts. The Suggested Upper Merged Ontology (SUMO) had been fully linked to PWN. For manual mapping of between PWN and SUMO, it was considered synonymy, hypernymy and instantiation between synsets of PWN and concepts of SUMO, and found the nearest instances of SUMO for synsets of PWN. Because the concept items of SUMO are much larger than those of PWN, it could be mapped between high level concepts of PWN and synonymy concepts of SUMO easily. (Ian Niles et al 2003). Dennis Spohr (2008) presented a general methodology to mapping EuroWordNet to the SUMO for extraction of selectional preferences for French. Jan Scheffczyk et al. (2006) introduced the connection of FrameNet to SUMO. They presented general-domain links between FrameNet Semantic Types and SUMO classes in SUOKIF and developed a semi-automatic, domain-specific approach for linking FrameNet Frame Elements to SUMO classes (Scheffczyk & al. 2006). Sara Tonelli et al. (2009) presented a supervised learning framework for the mapping of FrameNet lexical units onto PWN synsets to solve limited coverage of semantic phenomena for NLP applications. Their best results were recall 0.613, precision 0.761 and F1 measure 0.679.

Considerations on automatic mapping methods among language resources were always attempted for the sake of efficiency, using similarity measuring and evaluating methods. Typical traditional evaluating methods between concepts of heterogeneous language resources were the dictionary-based approaches (Kozima & al 1993), the semantic distance algorithm using PWN (Hirst & al 1998), the scaling method by semantic distance between concepts (Sussna 1997), conceptual similarity between concepts (Wu & al 1994), the scaled semantic similarity between concepts (Leacock 1998), the semantic similarity between concepts using IS-A relation (Resnik 1995), the measure of similarity between concepts (Lin 1998), Jiang and Conrath's (1997) similarity computations to synthesize edge and node based techniques, etc.

Satanjeev et al. (2003) presented a new measure of semantic relatedness between concepts that was based on the number of shared words (overlaps) in their definitions (glosses) for word sense disambiguation. The performances of their extended gloss overlap measure with 3-word window were recall 0.342, precision 0.351 and F1 0.346. Siddharth et al. (2003) presented the Adapted Lesk Algorithm to a method of word sense disambiguation based on semantic relatedness. In addition, Alexander et al (2006) introduced the 5 existing evaluating methods for PWN-based measures of lexical semantic relatedness and compared the performance of typical five measures of semantic relatedness for NLP applications and information retrieval. Among them, Jiang-Conrath's method showed the best performances: precision 0.247, recall 0.231 and F1 0.211 for Detection.

In many studies, it was presented a variety of the adapted evaluating algorithms. Among them, Jiang-Conrath's method, Lin's the measure of similarity and Resnik's the semantic similarity show good performances (Alexander & al 2006, Daniele 2009).

3 Language resources to be mapped

3.1 KorLex 1.5

KorLex 1.5 was constructed from 2004 to 2007. Different from its previous version (KorLex 1.0) which preserves all semantic relations among synsets of PWN, KorLex 1.5 modifies them by deletion/correction of existing synsets, addition of new synsets and conversion of hierarchical structure. Currently, KorLex includes nouns, verbs, adjectives, adverbs and classifiers: KorLexNoun, KorLexVerb, KorLexAdj, KorLexAdv and KorLexClas, respectively. Table 1 shows the size of KorLex 1.5, in which 'Trans' means the number of synsets translated from PWN 2.0 and 'Total' is the number of manually added synsets including translated ones.

	Word Forms	**Synsets**		**Word Senses**
		Trans	**Total**	
KorLexNoun	89,125	79,689	90,134	102,358
KorLexVerb	17,956	13,508	16,923	20,133
KorLexAdj	19,698	18,563	18,563	20,905
KorLexAdv	3,032	3,664	3,664	3,123
KorLexClas	1,181	-	1,377	1,377
Total	**130,992**	**115,424**	**130,661**	**147,896**

Table 1. Product of KorLex 1.5

KorLexNoun includes 25 semantic domains with 11 unique beginners with maximum 17 levels in depth and KorLexVerb includes 15 semantic domains with 11 unique beginners with maximum 12 levels in depth. Basically, KorLex synsets inherit the semantic information of PWN synsets mapped to them. The synset information of PWN consists of synset ID, semantic domain, POS, word senses, semantic relations, frame information, and so on.

We linked each word sense of KorLex 1.5 to that of STD by hand, when the former was built in our previous work (Yoon & al. 2009). STD includes 509,076 word entries with about 590,000 word senses. It contains a wide coverage for general words and a variety of example sentences for each meaning. More than 60% of word senses in KorLex 1.5 are linked to those of STD. KorLex 1.5, thus, inherits lexical relations described in STD, but both resources lack refined semantic-syntactic information.

3.2 Sejong Electronic Dictionary

SJD was developed during 1998-2007 manually by linguists for a variety of Korean NLP application as a general-purpose machine readable dictionary. Based on *Sejong semantic classes* (SJSC), approximately 25,000 nouns and 20,000 predicates (verbs and adjectives, SJPD) contain refined syntactic and semantic information.

SJSC is a set of hierarchical meta-languages classifying word senses and it includes 474 terminal nodes and 139 non-terminal nodes, and 6 unique beginners. Each unique beginner has levels from minimum 2 to maximum 7 levels in depth. Sejong Noun Dictionary (SJND) contains 25,458 entries and 35,854 word senses having lexical information for each entry: semantic classes of SJSC, argument structures, selectional restrictions, semantically related words, derivatioinal relations/words et al.

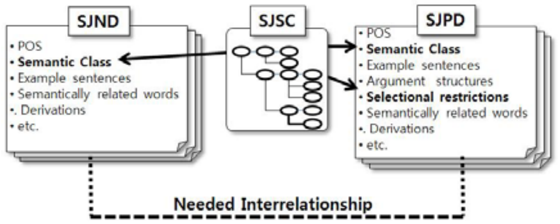


Figure 1. Correlation of lexical information among SJND, SJPD and SJSC

Figure 1 shows the correlation of lexical information among SJND, SJPD and SJSC. Certainly, that information of SJD should be applied to a variety of NLP applications: information retrieval, text analysis/generation, machine translations, and various studies and educations. However, SJD has much lower lexical coverage than KorLex. More serious problem is that SJND and SJPD are still noisy: internal consistency inside each dictionary and external interrelationship between SJND, SJPD, and SJSC need to be ameliorated, as indicated by dot line in Fig. 1.

4 Automatic Mapping from Semantic Class of SJSC to Synsets of KorLex

KorLex and SJSC have different hierarchical structures, grain sizes, and lexical information as aforementioned. For example, the semantic classes of SJSC are much bigger concepts in grain size than the synsets of KorLex: 623 concepts in SJSC vs.130,000 synsets in KorLex. Determining their semantic equivalence thus needs to be firmly based on linguistic clues.

Using following 3 linguistic clues that we found, we propose an automatic mapping method from semantic classes of SJSC to synsets of KorLex with three approaches to determine mapping candidate synsets: 1) Information of Monosemy/Polysemy of Word senses (IMPW), 2) Instances between Nouns of SJD and Word senses of KorLex (INW), and 3) Semantically Related words between Nouns of SJD and Synsets of KorLex (SRNS).

For automatic mapping method, following processes were conducted. First, to find word senses of synsets that matched to nouns of SJND for each semantic class. Second, to select mapping candidate synsets among them with three approaches aforementioned. Third, to determine the least upper bound (LUB) synsets and mapping synsets among candidates. Finally, to link each semantic class of SJSC to all lower-level synsets of LUB synsets.

4.1 Finding matched word senses between synsets and nouns of SJND

For a semantic class of SJSC, we first find word senses and synsets from KorLex that matched with nouns of SJND classified to that semantic class. Figure 2 shows the matched word senses and synsets between nouns of SJND, then synsets of KorLex for a semantic class. The left side of Figure 2 shows nodes of semantic classes with hierarchical structure and the center box shows the matched words (bold ones) among nouns of SJND with word senses of synsets in KorLex, and the right side shows matched word senses and synsets in KorLex' hierarchical structure.

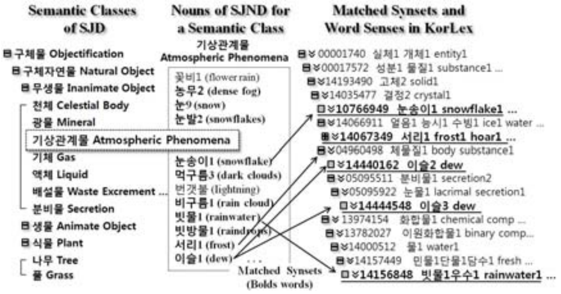

Semantic Classes of SJD
Nouns of SJND for a Semantic Class
Matched Synsets and Word Senses in KorLex
구체물 Objectification
구체자연물 Natural Object
무생물 Inanimate Object
천체 Celestial Body
광물 Mineral
기상관계물 Atmospheric Phenomena
기체 Gas
액체 Liquid
배설물 Waste Excrement ...
분비물 Secretion
생물 Animate Object
식물 Plant
나무 Tree
풀 Grass
기상관계물 Atmospheric Phenomena
꽃비1 (flower rain)
농무2 (dense fog)
눈9 (snow)
눈발2 (snowflakes)
눈송이1 (snowflake)
먹구름3 (dark clouds)
번갯불 (lightning)
비구름1 (rain cloud)
빗물1 (rainwater)
빗방울1 (raindrops)
서리1 (frost)
이슬1 (dew)
...
Matched Synsets (Bolds words)
00001740 실체1 개체1 entity1
00017572 성분1 물질1 substance1 ...
14193490 고체2 solid1
14035477 결정2 crystal1
10766949 눈송이1 snowflake1 ...
14066911 얼음1 농시1 수빙1 ice1 water ...
14067349 서리1 frost1 hoar1 ...
04960498 체물질1 body substance1
14440162 이슬2 dew
05095511 분비물1 secretion2
05095922 눈물1 lacrimal secretion1
14444548 이슬3 dew
13974154 화합물1 chemical comp ...
13782027 이원화합물1 binary comp...
14000512 물1 water1
14157449 민물1단물1담수1 fresh ...
14156848 빗물1우수1 rainwater1 ...

Figure 2. Matched word senses and synsets with nouns of SJND for a semantic class

For example, a semantic class 기상관련물 'Atmospheric Phenomena' (rectangle in the left) has nouns of SJND (words in the center), the bold words are the matched words with word senses of synsets from KorLex, and the underlined synsets of the right side are the matched ones and synset IDs in KorLex. The notations for automatic mapping process between semantic classes of SJSC and synsets of KorLex are as follows: noun of SJND is *ns*, matched noun ns_m, un-matched ns_u , semantic class of SJSC is *sc*, synset is *ss* and word sense of a synset is *ws* in KorLex, and monosemy word is w_{mono} and polysemy word is w_{poly}.

A semantic class *sc* has nouns *ns* of SJND having matched noun ns_m and un-matched ns_u by comparing with word senses *ws* of a synset *ss* in KorLex. Thus a synset has word senses as $ss_1=\{ws_1, ws_2, \ldots, ws_n\}=\{ns_{m1}, ns_{m2}, \ldots, ns_{mk}, ns_{u\,k+1}, ns_{u\,k+2}, \ldots\}$. And nouns of SJND for a semantic class sc_1 is presented $ns(sc_1)=\{ns_{m1},$

ns_{m2}, …, ns_{mk}, $ns_{u\ k}$, $ns_{u\ k+1}$, …}. Therefore, we can find the matched word senses ns_{m1} ~ ns_{mk} for a semantic class *sc* from nouns of SJND and word senses of a synset *ss* in KorLex.

4.2 Selecting Mapping Candidate Synsets

Using those matched synsets and word senses, we select mapping candidate synsets with three different approaches.

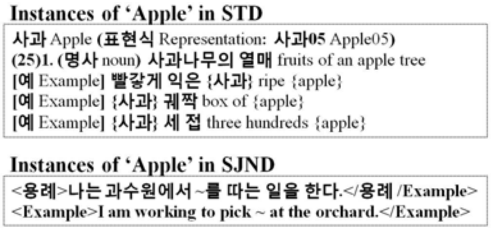
4.2.1 Using Information of Monosemy and Polysemy of KorLex

Using information of monosemy/polysemy of word senses of a synset, the first approach evaluates mapping candidate synsets. The candidate synsets are evaluated into three categories: *mc(A)* is a most relevant candidate synset, *mc(B)* is a relevant candidate synset and *mc(C)* is a reserved synset. Evaluation begins from lowest level synsets to top-level beginner. The process of first approach is as follows.

1) For a synset which contains a single word sense, *ss*={ws_1}, if the word sense is a monosemy, it is categorized as a a candidate synset *mc(A)*. If it is a polysemy, categorization is postponed for evaluating relatedness among siblings: candidate *mc(C)*.
2) In the case of a synset having more than one word sense, *ss*={ws_1, ws_2, …}, if the matched words ns_m among word senses of a synset are over 60%: $P_{ss}(ws)$=(count(ns_m)/ count(*ws*)) ≥ 0.6, we evaluate whether that synset is mapping candidate in the next step.
3) If all matched words ns_m of a synset are monosemic, we categorize it as a candidate synset *mc(A)*. If monosemic words among matched words are over 50%: $P_{ss}(w_{mono}|ns_m) \geq 0.5$, it is evaluated as a *mc(B)*. A synset containing polysemies over 50%: $P_{ss}(w_{poly}|ns_m) \geq 0.5$, categorization is postponed for evaluating relatedness among siblings: candidate synset *mc(C)*.
4) To repeat from step 1) to 3) for all of synsets, in order to evaluate mapping candidate synsets. And then, to construct hierarchical structure for all those synsets.

4.2.2 Using Instances between Nouns of SJND and Word senses of KorLex

The second approach is to evaluate mapping candidate synsets using comparison of instances between nouns of SJND and word senses of a synset. As for KorLex, we used the examples of STD linked to word senses of KorLex. Figure 3 shows instances of STD and SJND for a word sense 'Apple'.



Instances of 'Apple' in STD

사과 Apple (표현식 Representation: 사과05 Apple05)
(25)1. (명사 noun) 사과나무의 열매 fruits of an apple tree
[예 Example] 빨갛게 익은 {사과} ripe {apple}
[예 Example] {사과} 궤짝 box of {apple}
[예 Example] {사과} 세 접 three hundreds {apple}

Instances of 'Apple' in SJND

<용례>나는 과수원에서 ~를 따는 일을 한다.</용례 /Example>
<Example>I am working to pick ~ at the orchard.</Example>

Figure 3. Instances of STD and SJND for word sense 'Apple' 사과

We reformulated the Lesk algorithm (Lesk 1987, Banerjee and Pedersen 2002) for comparing instances and evaluating mapping candidate synsets. The process of evaluating mapping candidate synsets is as follows.

1) To compare instances of a noun *ns* of SJND with examples of a word of STD linked to word sense *ws* of a synset *ss*, and to compute the Relatedness-A(*ns*, *ws*) = score(instance(*ns*), example(*ws*)).
2) To compare all nouns *ns* of SJND for a semantic class with all nouns in instances of STD linked to word senses *ws*, and to compute the Relatedness-B(*ns*, *ws*) = score(∀*ns*, nouns(example(*ws*))).
3) If Relatedness-A(*ns*, *ws*) ≥ λ_1 and Relatedness-B(*ns*, *ws*) ≥ λ_2, a synset is evaluated as a candidate synset *mc(A)*. If either Relatedness-A(*ns*, *ws*) ≥ λ_1 or Relatedness-B(*ns*, *ws*) ≥ λ_2, evaluated as a candidate synset *mc(B)*. When threshold λ_1 and λ_2 were 1~4, we had good performances.
4) To repeat from step 1) to 3) for all of synsets, in order to determine mapping candidate synsets. And then, to construct hierarchical structure for all those synsets.

4.2.3 Using Semantically Relatedness between Nouns of SJND and Synsets of KorLex

The third approach is to evaluate mapping candidate synsets using comparison of semantic relations and their semantically related words between a noun of SJND and word senses of a synset. To compute the relatedness between them, we reformulated the computational formula of relatedness based on the Lesk algorithm (Lesk 1987, Banerjee & al 2002). The process of evaluating mapping candidate synsets is as follows.

1) To compare semantically related words: between synonyms, hypernyms, hyponyms and antonyms of a noun of SJND and those of a synset of KorLex. To compute the Relatedness-C(*ns*, *ss*) = score (relations(*ns*), relations(*ss*)).
2) To compare all nouns *ns* of SJND for a semantic class with synonyms, hypernyms and hyponyms of a synset of KorLex, and compute the Relatedness-D(*ns*, *ss*) = score (∀*ns*, relations(*ss*)).
3) If Relatedness-C(*ns*, *ss*) $\geq \lambda_3$ and Relatedness-D(*ns*, *ss*) $\geq \lambda_4$, a synset is evaluated as a candidate synset *mc(A)*. If either Relatedness-C(*ns*, *ss*) $\geq \lambda_3$ or Relatedness-D(*ns*, *ss*) $\geq \lambda_4$, evaluated as a candidate synset *mc(B)*. When threshold λ_3 and λ_4 were 1~4, we have good performances.
4) To repeat from step 1) to 3) for all of synsets, in order to determine mapping synsets. And then, to construct hierarchical structure for all those synsets.

4.3 Determining Least Upper Bound (LUB) Synsets and Mapping Synsets

Next, we determine the LUB synsets using mapping candidate synsets and hierarchical structure having semantic relations: parent, child and sibling. In order to determine LUB and mapping synsets, we begin evaluation with bottom-up direction. Using relatedness among child-sibling candidate synsets, we evaluated whether their parent synset is a LUB synset or not. If the parent is a LUB synset, we evaluate its parent (grand-parent of the candidate) synset using relatedness among its sibling synsets. If the parent is not a LUB, the candidate synsets *mc(A)* or *mc(B)* are determined as mapping synsets (or LUB) and stop finding LUB. For all semantic classes, we determine LUB and mapping synsets. Finally, we link the LUB and mapping synsets to each semantic class of SJSC. The process of determining of LUB and mapping synsets is as follows.

1) Using candidate synsets and their sibling, for all candidate synsets *mc(A)*, *mc(B)* or *mc(C)* selected from the processes of "4.2 Select Mapping Candidate Synsets", to determine whether it is a LUB or not and final mapping synsets.
2) Among sibling synsets, if the ratio of count(*mc(A)*) to count(*mc(A)+mc(B)+ mc(C)*) is over 60%, the parent synset of siblings is evaluated as a candidate synset *mc(A)* and as a LUB.
3) If the ratio of count(*mc(A)+mc(B)*) to count(*mc(A)+mc(B)+mc(C)*) is over 70%, the parent of siblings is evaluated as a candidate synset *mc(A)* and as a LUB. If the ratio of count(*mc(A)+mc(B)*) to count (*mc(A) +mc(B)+mc(C)*) is between 50% and 69%, the parent of siblings is evaluated as a candidate synset *mc(B)* and as a LUB.
4) And if the others, to stop finding LUB for that synset and to determine final mapping synsets with its own level of candidate.
5) To repeat from step 1) to 4) until finding LUB synsets and final mapping synsets.

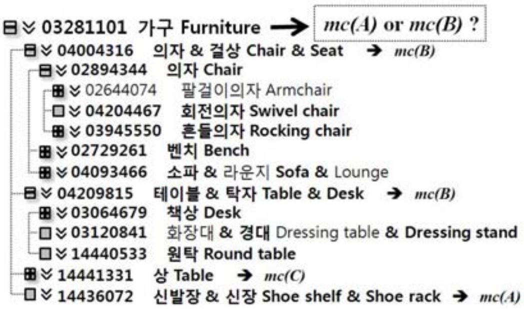

⊟ ≚ 03281101 가구 Furniture → mc(A) or mc(B) ?
⊟ ≚ 04004316 의자 & 걸상 Chair & Seat → mc(B)
⊟ ≚ 02894344 의자 Chair
⊞ ≚ 02644074 팔걸이의자 Armchair
□ ≚ 04204467 회전의자 Swivel chair
⊞ ≚ 03945550 흔들의자 Rocking chair
⊞ ≚ 02729261 벤치 Bench
⊞ ≚ 04093466 소파 & 라운지 Sofa & Lounge
⊟ ≚ 04209815 테이블 & 탁자 Table & Desk → mc(B)
⊞ ≚ 03064679 책상 Desk
□ ≚ 03120841 화장대 & 경대 Dressing table & Dressing stand
□ ≚ 14440533 원탁 Round table
⊞ ≚ 14441331 상 Table → mc(C)
□ ≚ 14436072 신발장 & 신장 Shoe shelf & Shoe rack → mc(A)

Figure 4. Hierarchical structure of mapping candidate synsets for a semantic class

Figure 4 shows hierarchical structure of mapping candidate synsets for a semantic class 'Furniture' and when candidate synsets' ID are '04004316' (Chair & Seat): *mc(B)*, '04209815' (Table & Desk): *mc(B)*, '14441331' (Table): *mc(C)*, and '14436072' (Shoe shelf & Shoe rack): *mc(A)*, we determine whether their parent synset '03281101' (Furniture) is a LUB or not, and evaluate it as a candidate synset *mc(A)* or *mc(B)*. In this case, synset '03281101' (Furniture) is a candidate *mc(A)* and a LUB synset.

For all semantic classes, we find their mapping LUB and mapping synsets using information of hierarchical structure and candidate synsets. Finally, we link each semantic class of SJSC to all lower level synsets of matched LUB synsets.

5 Experiments and Results

We experimented automatic mapping between 623 semantic classes of SJSC and 90,134 noun synsets of KorLex using the proposed automatic mapping method with three approaches. To evaluate the performances, we used the results of manual mapping as correct answers, that was mapped 474 semantic classes (terminal nodes) of SJSC to 65,820 synsets (73%) (include 6,487 LUB) among total 90,134 noun synsets of KorLex. We compared the results of automatic mapping with those of manual mapping. For evaluation of performances, we employed Recall, Precision and the F1 measure: F1 = (2*Recall*Precision)/(Recall+ Precision).

Approaches	Recall	Precision	F1
1)	0.904	0.502	0.645
2)	0.774	0.732	0.752
3)	0.670	0.802	0.730
1)+2)	**0.805**	**0.731**	**0.766**
1)+3)	**0.761**	**0.758**	**0.759**
2)+3)	0.636	0.823	0.718
1)+2)+3)	**0.838**	**0.718**	**0.774**

Table 2. Performances of automatic mapping with three approaches

Table 2 shows the performances of automatic mapping with three approaches: 1) IMPW, 2) INW, and 3) SRNS. The '1)', '2)' or '3) in the Table present the results using for each approach method and '1)+2)', '1)+3)' or '2)+3)' present those of combining two approaches. The '1)+2)+3)' presents those of the combining three approaches and we can see the best performances using the last approach among results: recall 0.837, precision 0.717 and F1 0.773. The first approach '1)' method shows high recall, but low precision and the third approach '3)' method present low recall and high precision. '1)+3)' and '2)+3)' shows good performances overall. Thus, we could see good performances using the combined approach methods.

Second, we compared the numbers of semantic classes, nouns entries of SJND, noun synsets and word senses of KorLex for each approach, after mapping processes.

As shown in Table 3, we can see the most numbers of mapping synsets using the '1)' approach. The '1)+2)+3)' shows the results similar to '1)', but has the best performances (see Table 2). The percentages in the round bracket present the ratio of the results of automatic mapping to original lexical data of *Sejong* and KorLex: 474 semantic classes of SJSC, 25,245 nouns of SJND and 90,134 noun synsets and 147,896 word senses in KorLex.

	SJD		KorLex	
Approaches	SC (SJSC)	Nouns (SJND)	Synsets	Word Senses
1)	**473**	**18,575**	**54,943**	**69,970**
2)	445	18,402	52,109	66,936
3)	413	18,047	49,768	64,003
1)+2)	463	18,521	52,563	67,109
1)+3)	457	18,460	51,786	66,157
2)+3)	383	17,651	48,398	62,063
1)+2)+3)	**466 (98.3%)**	**18,542 (72.8%)**	**54,083 (60%)**	**69,259 (46.8%)**

Table 3. Numbers of semantic class, noun of SJD, synset and word sense of KorLex

In manual mapping, we mapped 73% (65,820) synsets of KorLex for 474 semantic classes of SJSC. The 24,314 synsets was excluded in manual mapping among 90,134 total nouns synsets. The reasons of excluded synsets in manual mapping were 1) inconsistency of inheritance for lexical relations of parent-child in SJSC or KorLex, 2) inconsistency between criteria for SJSC and candidate synsets, 3) candidate synsets belonging to more than two semantic classes, 4) specific proper nouns (chemical compound names), and 5) polysemic abstract synsets (Bae & al. 2010).

In automatic mapping, we could map 60% (54,083) synsets among total nouns synsets (90,134) of KorLex, and it is 82.2% of the results of manual mapping. The 11,737 synsets was excluded in automatic mapping by comparing with manual mapping. Most of them were 1) tiny-grained synsets found in the lowest levels, 2) synsets having no matched word senses with those of SJND, 3) synsets with polysemic word senses, 4) word senses having poor instances in KorLex and in SJND, 5) word senses in SJND having poor semantic relations.

Level	LUB	Ratio	Level	LUB	Ratio
1	18	0.6%	9	230	7.3%
2	18	0.6%	10	98	3.1%
3	174	5.5%	11	32	1.0%
4	**452**	**14.3%**	12	20	0.6%
5	**616**	**19.5%**	13	4	0.1%
6	**570**	**18.0%**	14	4	0.1%
7	**486**	**15.4%**	15	2	0.1%
8	**442**	**14.0%**	16-17	0	0%

Table 4. Numbers and Ratio of LUB synsets excluded in automatic mapping

Table 4 shows the numbers and ratio of the LUB synsets excluded in automatic mapping for each level in depth. Most synsets are 4-8 levels synsets among 17 levels in depth.

6 Conclusions

We proposed a novel automatic mapping method with three approaches to link Sejong Semantic Classes and KorLex using 1) information of monosemy/polysemy of word senses, 2) instances of nouns of SJD and word senses of KorLex, 3) semantically related words of nouns of SJD and synsets of KorLex. To find common clues from lexical information among those language resources is important process in automatic mapping method. Our proposed automatic mapping method with three approaches shows notable performances by comparing with other studies on automatic mapping among language resources: recall 0.837, precision 0.717 and F1 0.773. Therefore, from those studies, we can improve Korean semantico-syntactic parsing technology by integrating the argument structures as provided by SJD, and the lexical-semantic hierarchy as provided by KorLex. In addition, we can enrich three resources: KorLex, SJD and STD as results of comparing and integrating them. We expect to improve automatic mapping technology among other Korean language resources through this study.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MEST) (No. 2007-0054887).

References

Jan Scheffczyk, Adam Pease, Michael Ellsworth. 2006. *Linking FrameNet to the Suggested Upper Merged Ontology*. Proc of the 2006 conference on Formal Ontology in Information Systems (FOIS 2006): 289-300.

Ian Niles and Adam Pease. 2003. *Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology*. In Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03).

KorLex, 2007. *Korean WordNet*, Korean Language processing Lab, Pusan National University. Available at http://korlex.cs.pusan.ac.kr

C. Hong. 2007. *The Research Report of Development 21th century Sejong Dictionary*, Ministry of Culture, Sports and Tourism, The National Institute of the Korean Language.

Dennis Spohr. 2008. *A General Methodology for Mapping EuroWordNets to the Suggested Upper Merged Ontology*, Proceedings of the 6th LREC 2008:1-5.

Alexander Budanitsky and Graeme Hirst. 2006. *Evaluating WordNet-based Measures of Lexical Semantic Relatedness*, Computational Linguistics,Vol 32: Issue 1:13- 47.

Siddharth Patwardhan, Satanjeev Banerjee and Ted Pedersen. 2003. *Using Measures of Semantic Relatedness for Word Sense Disambiguation*, CICLing 2003, LNCS(vol 2588):241-257.

Satanjeev Banerjee and Ted Pedersen. 2002. *An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet*, Proceedings of CICLing 2002, LNCS 2276:136-145

Sara Tonelli and Daniele Pighin. 2009. *New Features for FrameNet -WordNet Mapping*, Proceedings of the 13th Conference on Computational Natural Language Learning: 219-227.

Aesun Yoon, Soonhee Hwang, E. Lee, Hyuk-Chul Kwon. 2009. *Consruction of Korean WordNet 'KorLex 1.5'*, JourNal of KIISE: Sortware and Applications, Vol 36: Issue 1:92-108.

Soonhee Hwang, A. Yoon, H. Kwon. 2010. *KorLex 1.5: A Lexical Sematic Network for Korean Numeral Classifiers*, JourNal of KIISE: Sortware and Applications, Vol 37: Issue 1:60-73.

Sun-Mee Bae, Kyoungup Im, Aesun Yoon. 2010. *Mapping Heterogeneous Ontologies for the HLT Applications: Sejong Semantic Classes and KorLexNoun 1.5*, Korean Journal of Cognitive Science. Vol. 21: Issue 1: 95-126.

Aesun Yoon. 2010. *Mapping Word Senses of Korean Predicates Between STD(STandard Dictionary) and SJD(SeJong Electronic Dictionary) for the HLT Applications*, Journal of the Linguistic Society of Korea. No 56: 197-235.

Hyopil Shin. 2010. *KOLON: Mapping Korean Words onto the Microkosmos Ontology and Combining Lexical Resources*. Journal of the Linguistic Society of Korea. No 56: 159-196.

Sequential Tagging of Semantic Roles on Chinese FrameNet

Jihong LI
Computer Center
Shanxi University
lijh@sxu.edu.cn

Ruibo WANG, Yahui GAO
Computer Center
Shanxi University
{wangruibo,gaoyahui}@sxu.edu.cn

Abstract

In this paper, semantic role labeling(SRL) on Chinese FrameNet is divided into the subtasks of boundary identification(BI) and semantic role classification(SRC). These subtasks are regarded as the sequential tagging problem at the word level, respectively. We use the conditional random fields(CRFs) model to train and test on a two-fold cross-validation data set. The extracted features include 11 word-level and 15 shallow syntactic features derived from automatic base chunk parsing. We use the orthogonal array of statistics to arrange the experiment so that the best feature template is selected. The experimental results show that given the target word within a sentence, the best F-measures of SRL can achieve 60.42%. For the BI and SRC subtasks, the best F-measures are 70.55 and 81%, respectively. The statistical t-test shows that the improvement of our SRL model is not significant after appending the base chunk features.

1 Introduction

Semantic parsing is important in natural language processing, and it has attracted an increasing number of studies in recent years. Currently, its most important aspect is the formalization of the proposition meaning of one sentence through the semantic role labeling. Therefore, many large human-annotated corpora have been constructed to support related research, such as FrameNet (Baker et al., 1998), PropBank (Kingsbury and Palmer, 2002), NomBank (Meyers et al., 2004), and so on. On this basis, several international semantic evaluations have been organized, which include Senseval 3 (Litkowski, 2004), SemEval 2007 (Baker,et al., 2007), CoNLL 2008 (Surdeanu et al., 2008), CoNLL 2009 (Hajic et al., 2009), and so on.

The first SRL model on FrameNet was proposed by Gildea and Jurafsky(2002). The model consists of two subtasks of boundary identification(BI) and semantic role classification(SRC). Both subtasks were implemented on the pretreatment results of the full parsing tree. Many lexical and syntactic features were extracted to improve the accuracy of the model. On the test data of FrameNet, the system achieved 65% precision and 61% recall.

Most works on SRL followed Gildea's framework of processing the SRL task on English FrameNet and PropBank. They built their model on the full parse tree and selected features using various machine learning methods to improve the accuracy of SRL models. Many attempts have made significant progress, ssuch as the works of Pradhan et al. (2005), Surdeanu et al. (2007), and so on. Other researchers regarded the task of SRL as a sequential tagging problem and employed the shallow chunking technique to solve it, as described by Marquez at al. (2008).

Although the SRL model based on a full parse tree has good performance in English, this method of processing is not available in other languages, especially in Chinese. A systemic study of Chinese SRL was done by Xue et al. (2008). Like the English SRL procedure, he removed many

Proceedings of the 8th Workshop on Asian Language Resources, pages 22–29,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

uncorrelated constituents of a parse tree and relied on the remainder to identify the semantic roles using the maximum entropy model. When human-corrected parse is used, the F-measures on the PropBank and NomBank achieve 92.0 and 69.6%, respectively. However, when automatic full parse is used, the F-measures only achieve 71.9 and 60.4%, respectively. This significant decrease prompts us to analyze its causes and to find a potential solution.

First, the Chinese human-annotated resources of semantic roles are relatively small. Sun and Gildea only studied the SRL of 10 Chinese verbs and extracted 1,138 sentences in the Chinese Tree Bank. The size of the Chinese PropBank and Chinese NomBank used in the paper of Xue is significantly smaller than the ones used in English language studies. Moreover, more verbs exist in Chinese than in English, which increases the sparsity of Chinese Semantic Role data resources. The same problem also exists in our experiment. The current corpus of Chinese FrameNet includes about 18,322 human-annotated sentences of 1,671 target words. There is only an average of less than 10 sentences for every target word. To reduce the influence of the data sparsity, we adopt a two-fold cross validation technique for train and test labeling.

Second, because of the lack of morphological clues in Chinese, the accuracy of a state-of-the-art parsing system significantly decreases when used for a realistic scenario. In the preliminary stage of building an SRL model of CFN, we employed a Stanford full parser to parse all sentences in the corpus and adopted the traditional SRL technique on our data set. However, the experiment result was insignificant. Only 76.48% of the semantic roles in the data set have a constituent with the same text span in the parse tree, and the F-measure of BI can only achieves 54%. Therefore, we attempted to use another processing technique for SRL on CFN. We formalized SRL on CFN into a sequential tagging problem at the word level. We first extracted 11 word features into the baseline model. Then we added 15 additional base chunk features into the SRL model.

In this paper, the SRL task of CFN comprises two subtasks: BI and SRC. These are regarded as a sequential tagging problem at the word level. Conditional random fields(CRFs) model is employed to train the model and predict the result of the unlabeled sentence. To improve the accuracy of the model, base chunk features are introduced, and the feature selection method involving an orthogonal array is adopted. The experimental results illustrate that the F-measure of our SRL model achieves 60.42%. This is the best SRL result of CFN so far.

The paper is organized as follows. In Section 2, we describe the situation of CFN and introduce SRL on CFN. In Section 3, we propose our SRL model in detail. In Section 4, the candidate feature set is proposed, and the orthogonal-array-based feature selection method is introduced. In Section 5, we describe the experimental setup used throughout this paper. In Section 6, we list our experimental results and provide detailed analysis. The conclusions and several further directions are given at the end of this paper.

2 CFN and Its SRL task

Chinese FrameNet(CFN) (You et al., 2005) is a research project that has been developed by Shanxi University, creating an FN-styled lexicon for Chinese, based on the theory of Frame Semantics (Fillmore, 1982) and supported by corpus evidence. The results of the CFN project include a lexical resource, called the CFN database, and associated software tools. Many natural language processing(NLP) applications, such as Information Retrieval and Machine Translation, will benefit from this resource. In FN, the semantic roles of a predicate are called the frame elements of a frame. A frame has different frame elements. A group of lexical units (LUs) that evokes the same frame share the same names of frame elements.

The CFN project currently contains more than 1,671 LUs, more than 219 semantic frames, and has exemplified more than 18,322 annotated sentences. In addition to correct segmentation and part of speech, every sentence in the database is marked up to exemplify the semantic and syntactic information of the target word. Each annotated sentence contains only one target word.

(a). <medium-np-subj 第 1/m 章/q > <tgt=" 陈述" 介绍/v > <msg-np-obj 算法/n 与/c 数据/n

结构/n > ； /w

The CFN Corpus is currently at an early stage, and the available CFN resource is relatively limited, so the SRL task on CFN is described as follows. Given a Chinese sentence, a target word, and its frame, we identify the boundaries of the frame elements within the sentence and label them with the appropriate frame element name. This is the same as the task in Senseval-3.

3 Shallow SRL Models

This section proposes our SRL model architecture, and describes the stages of our model in detail.

3.1 SRL Model Architecture

A family of SRL models can be constructed using only shallow syntactic information as the input. The main differences of the models in this family mainly focus on the following two aspects.

i) model strategy: whether to combine the subtasks of BI and SRC?

ii) tagging unit: which is used as the tagging unit, word or chunk.

The one-stage and two-stage models are two popular strategies used in SRL tasks, as described by Sui et al. (2009). The word and the chunk are regarded as the two different tagging units of the SRL task.

In our SRL model, we consider BI and SRC as two stages, and the word is always used as the tagging unit. The detailed formalization is addressed in the following subsections.

3.2 BI

The aim of the BI stage is to identify all word spans of the semantic roles in one Chinese sentence. It can be regarded as a sequential tagging problem. Using the IOB2 strategy (Erik et al., 1999), we use the tag set $\{B,I,O\}$ to tag all words, where tag ”B” represents the beginning word of a chunk, ”I” denotes other tokens in the chunk, and ”O” is the tag of all tokens outside any chunks. Therefore, the example sentence (a) can be represented as follows:

(b). 第 1$|_B$ 章 $|_I$ 介绍 $|_O$ 算法 $|_B$ 与 $|_I$ 数据 $|_I$ 结构 $|_I$ ； $|_O$

To avoid the problem of data sparsity, we use all sentences in our train data set to train the model of BI.

3.3 SRC

After predicting the boundaries of semantic role chunks in a sentence, the proper semantic role types should be assigned in the SRC step. Although it can be easily modeled as a classification problem, we regarded it as a sequential tagging problem at the word level. An additional constraint is employed in this step: the boundary tags of the predicting sequence of this stage should be consistent with the the output of the BI stage.

One intuitive reason for this model strategy is that the SRC step can use the same feature set as BI, and it can further prove the rationality of our feature optimization method.

3.4 Postprocessing

Not all predicted IOB2 sequences can be transformed to the original sentence correctly; therefore, they should satisfy the following compulsory constraints.

(1) The tagging sequence should be regular. ”I...”, ”... OI...”, ”I-X...”, ”... O-I-X...”, ”... B-X-I-Y...”, and ”B-I-X-I-X-I-Y...” are not the regular IOB2 sequences.

(2) The tag for the target word must be ”O”.

We use the Algorithm 1 to justify whether the IOB2 sequences are regular.

Moreover, at the SRC stage, the boundary tags of the IOB2 sequence must be consistent with the given boundary tags.

For the BI stage, we firstly add an additional chunk type tag X to all ”B” and ”I” tags in the IOB2 sequences, and then use Algorithm 1 to justify the regularity of the sequences.

In the testing stage of the SRL model, we use the regular sequence with the max probability as the optimal output.

Algorithm 1. justify the regular IOB2 sequence

```
Input: (1) IOB2 sequence:$S = (s_1, .., s_n)$
where $s_i \in \{B - X, I - X, O\}$, and $1 \le i \le n$
(2) The position of target word in sentence $pt$
1, Initialization:
    (1) Current chunk type: $ct = NULL$;
    (2) Regularity of sequence: $state =' REG'$;
2, Check the tag of target word: $s_{pt}$:
    (1) If $s_{pt} ==' O'$: go to Step 3;
    (2) If $s_{pt} <>' O'$: $state =' IRR'$, and go to Step 4;
3,$For(i = 1; i \le n; i++)$
    (1) If $s_i ==' B - X'$: $ct =' X'$;
    (2) If $s_i ==' I - X'$ and $ct <>' X'$: $state =' IRR'$,
    and go to Step 4;
    (3) If $s_i ==' O'$: $ct = NULL$;
4, Stop
Output: Variable $state$;
```

3.5 Why Word-by-word?

We ever tried to use the methods of constituent-by-constituent and chunk-by-chunk to solve our SRL task on CFN, but the experiment results illustrate that they are not suitable to our task.

We use the Stanford Chinese full parser to parse all sentences in the CFN corpus and use the SRL model proposed by Xue et al.(2008) in our task. However, the results is insignificant. Only 66.72% of semantic roles are aligned with the constituents of the full parse tree, and the F-measure of BI only achieves 52.43%. The accuracy of the state-of-the-art Chinese full parser is not high enough, so it is not suitable to our SRL task.

Chunk-by-chunk is another choice for our task. When We use base chunk as the tagging unit of our model, only about 15% of semantic roles did not align very well with the boundary of automatically generated base chunks, and the F-measure is significantly lower than the method of word-by-word, as described by Wang et al.(2009).

Therefore, words are chosen as the tagging unit of our SRL model, which showed significant results from the experiment.

4 Feature Selection and Optimization

Word-level features and base-chunk features are used in our SRL research.

Base chunk is a Chinese shallow parsing scheme proposed by Professor Zhou. He constructed a high accuracy rule-based Chinese base chunk parse (Zhou, 2009), the F-measure of which can achieve 89%. We use this parse to generate all base chunks of the sentences in our corpus and to extract several types of features from them. The automatically generated base chunks of example sentences (a) are given as follows:

(c).[mp-ZX 第 1/m 章/q] [vp-SG 介绍/v] [np-SG 算法/n] 与/c [np-AM 数据/n 结构/n] ； /w

4.1 Candidate Feature Set

Three types of features are given as follows:

Features at the word level:

Word: The current token itself;

Part-of-Speech: The part of speech of the current token;

Position: The position of the current word relative to the target word(before, after, or on);

Target word: The target word in the sentence;

Features at the base chunk level:

Syntactic label: The syntactic label of the current token, such as, *B-np*,*I-vp*, etc;

Structural label: The structural label of the current token, such as, *B-SG*, *I-ZX*, etc;

Head word and its Part of Speech: The head word and its part of speech of the base chunk;

Shallow syntactic path: The combination of the syntactic tags from the source base chunk, which contains the current word, to the target base chunk, which contains the target word of the sentence;

Subcategory: The combination of the syntactic tags of the base chunk around the target word;

Other Features:

Named entity: The three types of named entities are considered: person, location, and time. They can be directly mapped from the part of speech of the current word.

Simplified sentence: A boolean feature. We use the punctuation count of the sentence to estimate whether the sentence is the simplified sentence.

Aside from the basic features described above, we also use combinations of these features, such as *word/POS* combination, etc.

4.2 Feature Optimization Method

In the baseline model, we only introduce the features at the word level. Table 1 shows the candidate features of our baseline model and proposes their optional sizes of sliding windows.

For Table 1, we use the orthogonal array $L_{32}(4^9 \times 2^4)$ to conduct 32 different templates.

The best template is chosen from the highest F-measure for testing the 32 templates. The detailed orthogonal-array-based feature selection method was proposed by Li et al.(2010).

Table 1. Candidate features of baseline models

Feature type	Window size			
word	[0,0]	[-1,1]	[-2,2]	[-3,3]
bigram of word	-	[-1,1]	[-2,2]	[-3,3]
POS	[0,0]	[-1,1]	[-2,2]	[-3,3]
bigram of POS	-	[-1,1]	[-2,2]	[-3,3]
position	[0,0]	[-1,1]	[-2,2]	[-3,3]
bigram of position	-	[-1,1]	[-2,2]	[-3,3]
word/POS	-	[0,0]	[-1,1]	[-2,2]
word/position	-	[0,0]	[-1,1]	[-2,2]
POS/position	-	[0,0]	[-1,1]	[-2,2]
trigram of position	-	[-2,0]	[-1,1]	[0,2]
word/target word	-	[0,0]		
target word	[0,0]			

Compared with the baseline model, the features at the word and base chunk levels are all considered in Table 2.

Table 2. Candidate features of the base chunk-based model

Feature type	Window size		
word	[0,0]	[-1,1]	[-2,2]
bigram of word	-	[-1,1]	[-2,2]
POS	[0,0]	[-1,1]	[-2,2]
bigram of POS	-	[-1,1]	[-2,2]
position	[0,0]	[-1,1]	[-2,2]
bigram of position	-	[-1,1]	[-2,2]
word/POS	-	[0,0]	[-1,1]
word/position	-	[0,0]	[-1,1]
POS/position	-	[0,0]	[-1,1]
trigram of position	-	[-2,0]	[-1,1]
syntactic label	[0,0]	[-1,1]	[-2,2]
syn-bigram	-	[-1,1]	[-2,2]
Syn-trigram	-	[-1,1]	[-2,2]
head word	[0,0]	[-1,1]	[-2,2]
head word-bigram	-	[-1,1]	[-2,2]
POS of Head	[0,0]	[-1,1]	[-2,2]
POS-bigram of head	-	[-1,1]	[-2,2]
syn/head word	[0,0]	[-1,1]	[-2,2]
stru/head word	[0,0]	[-1,1]	[-2,2]
shallow path	-	[0,0]	[-1,1]
subcategory	-	[0,0]	[0,0]
named Entity	-	[0,0]	[0,0]
simplified Sentence	-	[0,0]	[0,0]
target word(compulsory)	[0,0]		

The orthogonal array $L_{54}(2^1 \times 3^{25})$ is employed to select the best feature template from all candidate feature templates in Table 2. To distinguish it from the baseline model, we call the model based on the table 2 as the ”base chunk-based SRL model”.

For both feature sets described above, the target word is the compulsory feature in every template, and the boundary tags are introduced as features during the SRC stage.

The feature templates in Table 2 cannot contain the best feature template selected from Table 1. This is a disadvantage of our feature selection method.

5 Experimental Setup and Evaluation Metrics

5.1 Data Set

The experimental data set consists of all sentences of 25 frames selected in the CFN corpus. These sentences have the correct POS tags and CFN semantic information; they are all auto parsed by the rule-based Chinese base chunk parser. Table 3 shows some statistics on these 25 frames.

Table 3. Summary of the experimental data set

Frame	FEs	Sents	Frame	FEs	Sents
感受	6	569	因果	7	140
知觉特征	5	345	陈述	10	1,603
思想	3	141	拥有	4	170
联想	5	185	适宜性	4	70
自主感知	14	499	发明	12	198
查看	9	320	计划	6	90
思考	8	283	代表	7	80
非自主感知	13	379	范畴化	11	125
获知	9	258	证明	9	101
相信	8	218	鲜明性	9	260
记忆	12	298	外观	10	106
包含	6	126	属于某类	8	74
宗教信仰	5	54	**Totals**	**200**	**6,692**

5.2 Cross-validation technique

In all our experiments, three groups of two-fold cross-validation sets are used to estimate the performance of our SRL model. All sentences in a frame are cut four-fold on average, where every two folder are merged as train data, and the other two folds are used as test data. Therefore, we can obtain three groups of two-fold cross-validation data sets.

Estimating the parameter of fold number is one of the most difficult problems in the cross-validation technique. We believe that in the task of SRL, the two-fold cross validation set is a reasonable choice, especially when the data set is relative small. With a small data set, dividing it in half is split of data set is the best approximation of the real-world data distribution of semantic roles and the sparse word tokens.

5.3 Classifiers

CRFs model is used as the learning algorithm in our experiments. Previous SRL research has demonstrated that CRFs model is one of the best statistical algorithms for SRL, such as the works of Cohn et al. (2005) and Yu et al. (2007).

The crfpp toolkit[1] is a good implementation of the CRF classifier, which contains three different training algorithms: CRFL1, CRFL2, and MIRA. We only use CRFL2 with Gaussian priori regularization and the variance parameter C=1.0.

5.4 Evaluation Metrics

As described in SRL reseach, precision, recall, and F-measure are also used as our evaluation metrics. In addition, the standard deviation of the F-measure is also adopted as an important metric of our SRL model. The computation method of these metrics is given as follows:

Let P_j^i, R_j^i and F_j^i be the precision, recall, and F-measure of the jth group of the ith cross validation set, where $j = 1, 2, 3$ and $i = 1, 2$. The final precision(P), recall(R), and F-measure(F) of our SRL model are the expectation values of the P_j^i, R_j^i, and F_j^i, respectively.

The estimation of the variance of cross-validation is another difficult problem in the cross-validation technique. Although it has been proven that the uniform and unbiased estimation of the variance of cross-validation does not exist (Yoshua et al., 2007), we adopted the method proposed by Nadeau et al. (2007), to estimate the variance of the F-measure of cross-validation sets. This method is proposed hereinafter.

Let F_j be the average F-measure of the j group experiment, that is, $F_j = \frac{1}{2}(F_j^1 + F_j^2)$, where $j = 1, 2, 3$. The proposed estimator of the variance of F_j in the work of Nadeau et al. (2007) is as follows:

$$\begin{aligned}\widehat{Var}(F_j) &= (\frac{1}{K} + \frac{n_2}{n_1})\sum_{i=1}^{2}(F_j^i - F_j) \\ &= (\frac{1}{2} + 1)\sum_{i=1}^{2}(F_j^i - F_j)\end{aligned}$$

where, K is the fold number of cross-validation and n_1 and n_2 are the counts of training examples and testing examples. In our experimental setting, $K = 2$ and $\frac{n_2}{n_1} \approx 1$. Moreover, the estimation of the variance of the total F-measure is as follows:

$$\begin{aligned}Var(F) &= Var(\frac{1}{3}(F_1 + F_2 + F_3)) \\ &= \frac{1}{9}\sum_{j=1}^{3} Var(F_j)\end{aligned}$$

Using $\widehat{Var}(F_j)$ to estimate $Var(F_j)$, we can obtain:

$$\begin{aligned}\widehat{Var}(F) &= \frac{1}{9}\sum_{j=1}^{3}\widehat{Var}(F_j) \\ &= \frac{1}{6}\sum_{j=1}^{3}\sum_{i=1}^{2}(F_j^i - F_j)\end{aligned}$$

Finally, we can derive the standard deviation of the F-measure, that is, $std(F) = \sqrt{\widehat{Var}(F)}$.

5.5 Significance Test of Two SRL Models

To test the significance of SRL models A and B, we use the following statistics S.

$$S = \frac{F(A) - F(B)}{\sqrt{Var(F(A)) + Var(F(B))}} \sim t(n)$$

where $F(A)$ and $F(B)$ are the F-measures of models A and B, and n is the freedom degree of t-distribution, an integer nearest to the n'.

$$n' = \frac{3(Var(F(A)) + Var(F(B)))^2}{(Var(F(A))^2 + Var(F(B))^2)}$$

We use the $p - value(\cdot)$ to test the significance of SRL models A and B, which are given as follows:

$$p - value(F(A), F(B)) = P(S \geq t_{1-\alpha/2}(n))$$

If $p - value(F(A), F(B)) \leq 0.05$, the difference of the F-measures between models A and B is significant at 95% level.

[1] crfpp toolkit: http://crfpp.sourceforge.net/

6 Experimental Results and Discussion

We summarized the experiment results of every stage of our SRL model, that is, BI, SRC and a combination of these two steps (BI+SRC).

6.1 Baseline SRL Model

The results of the baseline model are given in Table 4, which only uses the features in Table 1.

Table 4. Results of the baseline model

	P(%)	R(%)	F(%)	std(F)
BI	74.42	66.80	70.40	0.0031
SRC	-	-	80.32	0.0032
BI+SRC	62.87	56.44	59.48	0.0050

In Table 1, because the results of the SRC stage are based on human-corrected boundary information, the precision, recall, and F-measure of this stage are the same. Therefore, we only give the F-measure and its deviation at the SRC stage.

In the baseline model, the BI stage is the bottleneck of our SRL model. Its F-measure only achieves 70.4%, and the recall is lower than the precision. Moreover, the F-measure of the final model only achieves 59.48%, and its standard deviation is larger than both stages.

6.2 Base chunk-based SRL Model

When base chunk features, proposed in Table 2, are employed in the SRL model, we can obtain the results summarized in Table 5.

Table 5. Results of the base chunk-based model

	P(%)	R(%)	F(%)	std(F)
BI	74.69	66.85	70.55	0.0038
SRC	-	-	81.00	0.0029
BI+SRC	63.97	57.25	60.42	0.0049

A comparison of Table 4 and Table 5 provides the following two conclusions.

(1) When base chunk features are used, all P, R, F at every stage slightly increase ($< 1\%$).

(2) The significance test values between the baseline model and the base chunk-based model are given in Table 6. For every stage, the performance boost after introducing the base chunk features is not significant at 95% level. However, the impact of base chunk features at the SRC stage is larger than that at the BI stage.

Table 6. Test values between two SRL models

	BI	SRC	BI+SRC
$p - value$	0.77	0.166	0.228

7 Conclusions and Further Directions

The SRL of Chinese predicates is a challenging task. In this paper, we studied the task of SRL on the CFN. We proposed a two-stage model and exploited the CRFs classifier to implement the automatic SRL systems. Moreover, we introduced the base chunk features and the OA-based method to improve the performance of our model. Experimental results shows that the F-measure of our best model achieves 60.42%, and the base chunk features cannot improve the SRL model significantly.

In the future, we plan to introduce unlabeled data into the training phase and use the EM-schemed semi-supervised learning algorithms to boost the accuracy of our SRL model.

Acknowledgement

The authors would like to thank Prof. Kaiying LIU for his comments and Prof. Qiang ZHOU for the base-chunk parser.

References

Baker, C., Fillmore, C., and John B. 1998. The Berkeley Framenet project. *In Proceedings of COLING-ACL*, 86-90, Montreal, Canada.

Baker, C., Ellsworth, M., Erk, K. 2007. SemEval'07 Task 19: Frame semantic structure extraction. *Proceedings of the 4th International Workshop on Semantic Evaluations*, 99-104, Prague, Czech Republic.

Cohn, T., Blunsom P. 2005. Semantic role labeling with tree conditional random fields. *Proceedings of CoNLL 2005*, ACL, 169-172.

Erik F., and John V. 1999. Representing text chunks. *In Proceedings of EACL'99*, 173-179.

Fillmore, C. 1982. Frame Semantics. *In The Linguistic Society of Korea*, Seoul: Hanshin.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling for semantic roles. *Computational Linguistics*, 28(3):245-288.

Hajic, J., Ciaramita, M., Johansson, R., Kawahara, D., Marti, M., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Stěpánek, J., Stranak, P., Surdeanu, M., Nianwen X., Zhang, Y. 2009. The CoNLL-2009 shared task: syntactic and semantic dependencies in multiple languages. *In Proceedings of CoNLL 2009*, 1-18, Boulder, CO, USA..

Jihong, L., Ruibo, W., Weilin, W., and Guochen, L. 2010. Automatic Labeling of Semantic Roles on Chinese FrameNet. *Journal of Software*, 2010, 21(4):597-611.

Jiangde, Y., Xiaozhong, F., Wenbo, P., and Zhengtao, Y. 2007. Semantic role labeling based on conditional random fields *Journal of southeast university (English edition)*, 23(2):5361-364.

Liping, Y., and Kaiying, L. 2005. Building Chinese FrameNet database. *In Proceedings of IEEE NLP-KE'05* , 301-306.

Litkowski, K. 2004. Senseval-3 task automatic labeling of semantic roles. *Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 9-12, Barcelona, Spain.

Màrquez, L., Carreras, X., Litkowski, K., Stevenson, S. 2008. Semantic Role Labeling: An Introduction to the Special Issue. *Computational Linguistics*, 34(2):145-159.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. 2004. The NomBank Project: An interim report. *In Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*, 24-31, Boston, MA, USA.

Nadeau, C., and Bengio, Y. 2003. Inference for the generalization error. *Machine Learning*, 52: 239-281.

Nianwen, X. 2008. Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 2008, 34(2): 225-255.

Paul, K., and Martha, P. 2002. From TreeBank to PropBank. *In Proceedings of LREC-2002*, Canary Islands, Spain.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J., Jurafsky, D. 2005. Support vector learning for semantic argument classification. *Machine Learning*, 2005, 60(1):11-39.

Qiang, Z. 2007. A rule-based Chinese base chunk parser. *In Proc. of 7th International Conference of Chinese Computation (ICCC-2007)*, 137-142, Wuhan, China.

Ruibo, W. 2004. Automatic Semantic Role Labeling of Chinese FrameNet Based On Conditional Random Fields Model. *Thesis for the 2009 Master's Degree of Shanxi University*, Taiyuan, Shanxi, China.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., Nivre, J. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. *In Proceedings of CoNLL 2008*, 159-177, Manchester, England, UK.

Surdeanu, M., Màrquez, L., Carreras, X., Comas, P. 2007. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, 29:105-151.

Yoshua, B., and Yves, G. 2004. No unbiased estimator of the variance of K-fold cross-validation *Journal of Machine Learning Research*, 5:1089-1105.

Weiwei, S., Zhifang, S., Meng, W., and Xing, W. 2009. Chinese semantic role labeling with shallow parsing. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing(EMNLP 2009)*, ACL, 1475-1483.

Augmenting a Bilingual Lexicon with Information for Word Translation Disambiguation

Takashi Tsunakawa
Faculty of Informatics
Shizuoka University
tuna@inf.shizuoka.ac.jp

Hiroyuki Kaji
Faculty of Informatics
Shizuoka University
kaji@inf.shizuoka.ac.jp

Abstract

We describe a method for augmenting a bilingual lexicon with additional information for selecting an appropriate translation word. For each word in the source language, we calculate a correlation matrix of its association words versus its translation candidates. We estimate the degree of correlation by using comparable corpora based on these assumptions: "parallel word associations" and "one sense per word association." In our word translation disambiguation experiment, the results show that our method achieved 42% recall and 49% precision for Japanese-English newspaper texts, and 45% recall and 76% precision for Chinese-Japanese technical documents.

1 Introduction

The bilingual lexicon, or bilingual dictionary, is a fundamental linguistic resource for multilingual natural language processing (NLP). For each word, multiword, or expression in the source language, the bilingual lexicon provides translation candidates representing the original meaning in the target language.

Selecting the right words for translation is a serious problem in almost all of multilingual NLP. One word in the source language almost always has two or more translation candidates in the target language by looking up them in the bilingual lexicon. Because each translation candidate has a distinct meaning and property, we must be careful in selecting the appropriate translation candidate that has the same sense as the word inputted. This task is often called word translation disambiguation.

In this paper, we describe a method for adding information for word translation disambiguation into the bilingual lexicon. Comparable corpora can be used to determine which word associations suggest which translations of the word (Kaji and Morimoto, 2002). First, we extract word associations in each language corpus and align them by using a bilingual dictionary. Then, we construct a word correlation matrix for each word in the source language. This correlation matrix works as information for word translation disambiguation.

We carried out word translation experiments on two settings: English-to-Japanese and Chinese-to-Japanese. In the experiments, we tested Dice/Jaccard coefficients, pointwise mutual information, log-likelihood ratio, and Student's *t*-score as the association measures for extracting word associations.

2 Constructing word correlation matrices for word translation disambiguation

2.1 Outline of our method

In this section, we describe the method for calculating a word correlation matrix for each word in the source language. The correlation matrix for a word f consists of its association words and its translation candidates. Among the translation candidates, we choose the most acceptable one that is strongly suggested by its association words occurring around f.

We use two assumptions for this framework:

(i) Parallel word associations:
Translations of words associated with each other in a language are also associated with each other in another language

Proceedings of the 8th Workshop on Asian Language Resources, pages 30–37,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

(Rapp, 1995). For example, two English words “tank” and “soldier” are associated with each other and their Japanese translations “戦車 (*sensha*)” and “兵士 (*heishi*)” are also associated with each other.

(ii) One sense per word association:
A polysemous word exhibits only one sense of a word per word association (Yarowsky, 1993). For example, a polysemous word “tank” exhibits the “military vehicle” sense of a word when it is associated with “soldier,” while it exhibits the “container for liquid or gas” sense when it is associated with “gasoline.”

Under these assumptions, we determine which of the words associated with an input word suggests which of its translations by aligning word associations by using a bilingual dictionary. Consider the associated English words (tank, soldier) and their Japanese translations (戦車 (*sensha*), 兵士 (*heishi*)). When we translate the word “tank” into Japanese, the associated word “soldier” helps us to translate it into “戦車 (*sensha*)”, not to translate it into “タンク (*tanku*)” which means “a storage tank.”

This naive method seems to suffer from the following difficulties:

- A disparity in topical coverage between two corpora in two languages
- A shortage in the bilingual dictionary
- The existence of polysemous associated words that cannot determine the correct sense of the input word

For these difficulties, we use the tendency that the two words associated with a third word are likely to suggest the same sense of the third word when they are also associated with each other. For example, consider an English associated word pair (tank, troop). The word “troop” cannot distinguish the different meanings because it can co-occur with the word “tank” in both senses of the word. The third word “soldier,” which is associated with both “tank” and “troop,” can suggest the translation “戦車 (*sensha*).”

The overview of our method is shown in Figure 1. We first extract associated word pairs in the source and target languages from comparable corpora. Using a bilingual dictionary, we obtain alignments of these word associations. Then, we iteratively calculate a correlation matrix for each word in the source language. Finally, we select the translation with the highest correlation from the translation candidates of the input word and the co-occurring words.

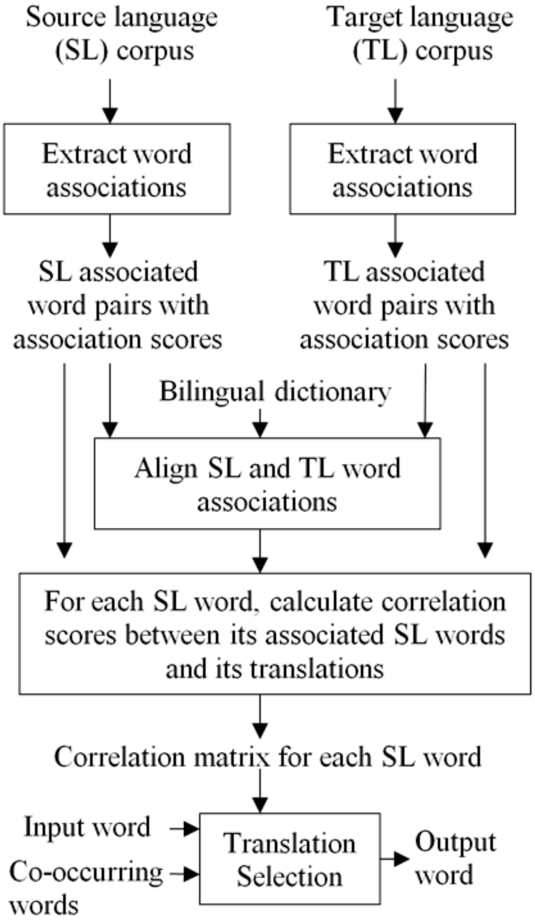

Source language (SL) corpus
Target language (TL) corpus
Extract word associations
Extract word associations
SL associated word pairs with association scores
TL associated word pairs with association scores
Bilingual dictionary
Align SL and TL word associations
For each SL word, calculate correlation scores between its associated SL words and its translations
Correlation matrix for each SL word
Input word
Co-occurring words
Translation Selection
Output word

Figure 1. Overview of our method.

For each input word in the source language, we calculate correlation values between their translation candidates and their association words. The algorithm is shown in Figure 2.

In Algorithm 1, the initialization of correlation values is based on word associations, where D is a set of word pairs in the bilingual dictionary, and A_f and A_e are the sets of associated word pairs. First, we retain associated words $f'(i)$ when its translation e' exists and when e' is associated with e. In the iteration, the correlation values of associated words $f'(i)$ that suggest $e(j)$ increase relatively by using association scores $\alpha(f'(i), f)$ and

$\alpha(f'(i), f'')$. In our experiments, we set the number of iterations N_r to 10.

2.2 Alternative association measures for extracting word associations

We extract co-occurring word pairs and calculate their *association scores*. In this paper, we focus on some frequently used metrics for finding word associations based on their occurrence/co-occurrence frequencies.

Suppose that words x and y frequently co-occur. Let n_1 and n_2 be the occurrence frequencies of x and y respectively, and let m be the frequency that x and y co-occur between w content words. The parameter w is a window size that adjusts the range of co-occurrences.

Let N and M be the sum of occurrences/co-occurrences of all words/word pairs, respectively. The frequencies are summarized in Table 1.

The word association scores $\alpha(x, y)$ are defined as follows:

- Dice coefficient (Smadja, 1993)

$$\mathrm{Dice}(x,y) = \frac{2m}{n_1 + n_2} \qquad (1)$$

- Jaccard coefficient (Smadja et al., 1996)

$$\mathrm{Jaccard}(x,y) = \frac{m}{n_1 + n_2 - m} \qquad (2)$$

- Pointwise mutual information (pMI) (Church and Hanks, 1990)

$$\mathrm{pMI}(x,y) = \log_2 \frac{m/M}{(n_1/N)(n_2/N)} \qquad (3)$$

- Log-likelihood ratio (LLR) (Dunning, 1993)

$$\begin{aligned} &\mathrm{LLR}(x,y) \\ &= -2\big(\mathrm{logL}(m, n_1, r) \\ &+ \mathrm{logL}(n_2 - m, N - n_1, r) \\ &- \mathrm{logL}(m, n_1, r_1) \\ &- \mathrm{logL}(n_2 - m, N - n_1, r_2)\big); \qquad (4) \end{aligned}$$

$$\begin{aligned} &\mathrm{logL}(k, n, r) = \\ &k \log_2 r + (n - k) \log_2 (1 - r), \qquad (5) \end{aligned}$$

$$r_1 = \frac{m}{n_1}, r_2 = \frac{n_2 - m}{N - n_1}, r = \frac{n_2}{N} \qquad (6)$$

- Student's t-score (TScore) (Church et al., 1991)

$$\mathrm{TScore}(x,y) = \frac{m - n_1 n_2 / N}{\sqrt{m}} \qquad (7)$$

We calculate association scores for all pairs of words when their occurrence frequencies are not less than a threshold T_f and when their

```
Algorithm 1:
Input:
    f: an input word
    f'(1), …, f'(I): associated words of f
    e(1), …, e(J): translation candidates of f
    N_r: number of iterations
    A bilingual lexicon
    Word association scores α for both languages
Output:
    C_f = [C_f(f'(i), e(j))]: a correlation matrix for f

if $\sum_k \delta_f(f'(i), e(k)) \neq 0$ then
    $C_f(f'(i), e(j)) \leftarrow \frac{\delta_f(f'(i), e(j))}{\sum_k \delta_f(f'(i), e(k))}$
else
    $C_f(f'(i), e(j)) \leftarrow 0$
end
$i \leftarrow 0$
while $i < N_r$
    $i \leftarrow i + 1$
    $C_f(f'(i), e(j)) \leftarrow \alpha(f'(i), f)$
      $\times \frac{\sum_* \alpha(f'(i), f'') \cdot C_f(f'', e(j))}{\max_k \sum_* \alpha(f'(i), f'') \cdot C_f(f'', e(k))}$
end
$\left( \delta_f(f', e) = \begin{cases} 1 & (\exists e' : (e, e') \in A_e, (f', e') \in D) \\ 0 & \text{(otherwise)} \end{cases} \right)$
$(\Sigma_* := \Sigma_{\{f'' | (f, f'') \in A_f, (f'(i), f'') \in A_f\}})$
```

Figure 2. Algorithm for calculating correlation matrices.

	x occur	x not occur	Total
y occur	m	$n_2 - m$	n_2
y not occur	$n_1 - m$	$M - n_1 - n_2 + m$	$N - n_2$
Total	n_1	$N - n_1$	N

Table 1. Contingency matrix of occurrence frequencies.

<home>	国 (*kuni*) [country]	本塁 (*honrui*) [home base]	家 (*ie*) [house]	自宅 (*jitaku*) [my home]	家庭 (*katei*) [homeplace]	施設 (*shisetsu*) [facilities]
base	0.009907	**0.017649**	0.005495	0.006117	0.005186	0.005597
game	0.043507	**0.048358**	0.025145	0.028208	0.019987	0.023014
Kansas	**0.010514**	0.003786	0.004280	0.007307	0.004320	0.005459
run	0.023468	**0.042035**	0.014430	0.015765	0.012061	0.012986
season	0.044855	**0.050952**	0.025406	0.028506	0.020716	0.023631

Table 2. Word correlation matrix for a word "home,"
as information for word translation disambiguation

threshold T_c.

We handle word pairs whose association scores are not less than a predefined value T_A; some of the thresholds were evaluated in our experiment. The associated word pair sets A_f and A_e in Algorithm 1 includes only word pairs whose scores are not less than T_A in the source and target language, respectively.

3 Word Translation Disambiguation

Consider that a translator changes a word in an input sentence. Usually, two or more translation candidates are enumerated in the bilingual dictionary for a word. The translator should select a translation word that is grammatically/ syntactically correct, semantically equivalent to the input, and pragmatically appropriate.

We assume that the translation word *e* for an input word *f* tends to be selected if words occurring around *f* are strongly correlated with *e*. Using the correlation matrices, we select *e* as a translation if the associated word *f'* occurs around *f* and the score $C_f(f',e)$ is large. In addition, we take distance between *f* and *f'* into account.

We define the score of the translation word $e(f_0)$ for an input word f_0 as follows. Consider an input word f_0 that occurs in the context of "… f_{-2} f_{-1} f_0 f_1 f_2 …." The score for a translation word $e(f_0)$ for an input word f_0 is

$$\mathrm{Score}\big(e(f_0)\big) = \sum_{1\le|p|\le\gamma} \frac{1}{\sqrt{|p|}} C_{f_0}\left(f_p, e(f_0)\right), \quad (8)$$

defined as where p is the relative position of the words surrounding f_0, $C_{f_0}\left(f_p, e(f_0)\right)$ is the value of the correlation matrix for f_0, and γ is the window size for word translation disambiguation.

> A career .284 hitter, Beltran batted .267 in the regular *season*, split between *Kansas* City and Houston, but came alive in the playoffs. He hit .435 in 12 postseason *games*, with six stolen *bases*, eight **home** *runs* and 14 *runs* batted in.

Figure 3. An example of an input word "home"

A simple example is shown in Figure 3. The word "home" in this context means the sense of "home base" used in baseball games, not "house" or "hometown." The surrounding words such as "games," "bases," and "runs" can be clues for indicating the correct sense of the word. By using the correlation matrix (Table 2) and formula (8), we calculate a score for each translation candidate and select the best translation with the largest score. In this case, Score(本塁 (*honrui*)) = 0.1134 was the best score and the correct translation was selected.

4 Experiment

4.1 Experimental settings

We carried out word translation experiments on two different settings. In the first experiment (Experiment A), we used large-scale comparable corpora from English and Japanese newspaper texts. The second experiment (Experiment B) was targeted at translating technical terms by using Chinese and Japanese domain-specific corpora.

We used the following linguistic resources for the experiments:

- Experiment A

    - Training comparable corpora
        - The New York Times texts from English Gigaword Corpus Fourth Edition (LDC2009T13): 1.6 billion words
        - The Mainichi Shimbun Corpus (2000-2005): 195 million words
    - Test corpus
        - A part of The New York Times (January 2005): 157 paragraphs, 1,420 input words
- Experiment B1
    - Training comparable corpus
        - In-house Chinese-Japanese parallel corpus in the environment domain: 53,027 sentence pairs[1]
    - Test corpus
        - A part of the training data: 1,443 sentences, 668 input words
- Experiment B2
    - Training comparable corpus
        - In-house Chinese-Japanese parallel corpus in the medical domain: 123,175 sentence pairs
    - Test corpus
        - A part of the training data: 940 sentences, 3,582 input words
- Dictionaries
    - Japanese-English bilingual dictionaries: Total 333,656 term pairs
        - EDR Electronic Dictionary
        - Eijiro, Third Edition
        - EDICT (Breen, 1995)
    - Chinese-English bilingual dictionary
        - Chinese-English Translation Lexicon Version 3.0 (LDC 2002L27): 54,170 term pairs
        - Wanfang Data Chinese-English Science/Technology Bilingual Dictionary: 525,259 term pairs

For the Chinese-Japanese translation, we generated a Chinese-Japanese bilingual dictionary by merging Chinese-English and Japanese-English dictionaries. The Chinese-Japanese bilingual dictionary includes every Chinese-Japanese term pair (t_C, t_J) when (t_C, t_E) and (t_J, t_E) were present in the dictionaries for one or more English terms t_E. This merged dictionary contains about two million term pairs. While these Chinese-Japanese term pairs include wrong translations, it was not a serious problem in our experiments because wrong translations were excluded in the procedure of our method.

We applied morphological analysis and part-of-speech tagging by using TreeTagger (Schmid, 1994) for English, JUMAN for Japanese, and mma (Kruengkrai et al., 2009) for Chinese, respectively.

In the test corpus, we manually annotated reference translations for each target word.[2] The parameters we used were as follows:

Experiment A:

T_f= 100 (Japanese), T_f= 1000 (English), T_c = 4, w = 30, γ = 30.

Experiment B1/B2:

T_f= 100, T_c = 4, w = 10, γ = 25.

Some of the parameters were empirically adjusted.

In the experiments, the matrices could be obtained for 9103 English words (A), 674 Chinese words (B1) and 1258 Chinese words (B2), respectively. In average one word had 3.24 (A), 1.15 (B1) and 1.51 (B2) translation candidates[3] by using the best setting. Table 2 is the resulted matrix for the word "home" in the Experiment A.

4.2 Results of English-Japanese word translation

Table 3 shows the results of Experiment A. We classified the translation results for 1,420 target English words into four categories: True, False, R, and M. When the translation was output, the result was True if the output is included in the reference translations, and it was False otherwise. The result was R when all the associated words in the correlation matrix

[1] We could prepare only parallel corpora for Chinese-Japanese language pair as training corpora. For our experiments, we assumed them as comparable corpora and did not use the correspondence of sentence pairs.

[2] We prepared multiple references for several target words. The average numbers of reference translations for an input word are 1.84 (A), 1.50 (B1), and 1.48 (B2), respectively.

[3] From each matrix, we cut off the columns with translations that do not have the best scores for any associated words, because such translations are never selected.

did not occur around the input word. The result was M when no correlation matrix existed for the input word. We did not select a translation output in these cases. The recall and precision are shown in the parentheses below.

- Recall = (True) / (Number of input words)
- Precision = (True) / (True + False)

Among the settings, we obtained the best results, 42% recall and 49% precision, when we used the Jaccard coefficient for association scores and T_A = 0, which means all pairs were taken into consideration. Among other settings, the Dice coefficient achieved a comparable performance with Jaccard.

4.3 Results of Chinese-Japanese word translation

Tables 4 and 5 show the results of Experiment B. In each domain, we tested only the settings on Dice, pMI, and LLR with T_A = 0.

In the environmental domain, the pointwise mutual information score achieved the best performance, 45% recall and 76% precision. However, the Dice coefficient gave the best recall (55%) for the medical domain. This result indicates that Experiment B1/B2 had higher precision and more words without the correlation matrix than Experiment A had.

Score	T_A	True	False	R	M
Dice	0	588 (41%/48%)	627	90	115
	0.001	586 (41%/49%)	619	94	121
	0.01	479 (34%/49%)	507	243	191
Jaccard	0	**594** (**42%/49%**)	621	90	115
	0.001	584 (41%/49%)	609	105	122
	0.01	348 (25%/48%)	378	374	320
pMI	0	292 (21%/49%)	309	704	115
	1	293 (21%/49%)	308	703	116
LLR	10	530 (37%/42%)	747	28	115
	100	529 (37%/42%)	744	32	115
T-Score	1	486 (34%/38%)	793	26	115
	4	489 (34%/38%)	787	26	118

Table 3. Results of English-Japanese word translation (A)

Score	T_A	True	False	R	M
Dice	0	**1984** (**55%**/69%)	895	82	621
pMI	0	1886 (53%/**70%**)	804	271	621
LLR	0	1652 (46%/57%)	1246	63	621

Table 4. Results of Chinese-Japanese word translation for environmental domain (B1).

Score	T_A	True	False	R	M
Dice	0	277 (41%/69%)	124	14	253
pMI	0	**303** (**45%/76%**)	95	17	253
LLR	0	269 (40%/67%)	131	15	253

Table 5. Results of Chinese-Japanese word translation for medical domain (B2).

4.4 Discussion

As a result, we could generate bilingual lexicons with word translation disambiguation information for 9103 English words and 1932 Chinese words. Although the number of words might be augmented by changing the settings, the size does not seem to be sufficient as bilingual dictionaries. The availability of larger output should be investigated.

The experimental results show that our method selected correct translations for at least half of the input words if a correlation matrix existed and if the associated words co-occur. Among all input words, at least 40% of the input words can be translated. The bilingual dictionaries included 24.4, 38.6, and 52.0 translation candidates for one input word in Experiment A, B1, and B2, respectively. When we select the most frequent word, the precisions were 7%, 1%, and 1%, respectively. Meanwhile, the average numbers of translation candidates in the correlation matrices for one input word are shown in Table 6. These indicate that our method effectively removed noisy translations from the Chinese-Japanese dictionary merged Japanese-English and Chinese-English dictionaries, and that the association scores contributed word translation disambiguation.

Among the settings, the Jaccard/Dice coefficients were proven to be effective, although pointwise mutual information (pMI) was also effective for technical domains and the Chinese-Japanese language pair. Because the Jaccard/Dice coefficients were originally used for measuring the proximity of sets, these might be effective for collecting related words by using the similarity of kinds of co-occurring words. However, pMI tends to emphasize low-frequency words as associated words. The consequence of this tendency might be that low-frequency associated words do not appear around the input word in the newspaper text.[4]

In most metrics for the association score, the lowest threshold value T_A achieved the best performance. This result indicates that the cut-off of associated words by some thresholds was not effective, although it requires more time and memory space to obtain correlation matrices without cut-off. How to optimize other parameters in our method remains unsolved. More words without the correlation matrix were present in Experiment B1/B2 than in Experiment A because the input word was often a technical term that was not in the bilingual dictionary. The better recall and precision of Experiment B1/B2 came from several reasons, including difference of test sets and language pairs. In addition, it might have an impact on this result that the fact that word translation disambiguation of technical terms is easier than word translation disambiguation of common words.

We handled only nouns as input words and associated words in this study. Considering only the co-occurrence in a fixed window would be insufficient to apply this method to the translation of verbs and other parts of speech. In future work, we will consider syntactic co-occurrence, which is obtained by conducting dependency parsing of the results of a sentence. The correlation between associated words and translation candidates also needs to be re-examined. Similarly, we will handle verbs as associated words to the input nouns by using syntactic co-occurrence.

Score	T_A	Exp.A	Exp.B1	Exp.B2
Dice	0	1.83	0.61	0.91
pMI	0	1.27	0.50	0.79
LLR	10/0	1.34	1.48	2.75

Table 6. Average numbers of translation candidates in the correlation matrices for one input word.

5 Related Work

Statistical machine translation (Brown et al., 1990) automatically acquires knowledge for word translation disambiguation from parallel corpora. Word translation disambiguation is based on probabilities calculated from the word alignment, phrase pair extraction, and the language model. However, much broad context/domain information is not considered. Carpuat and Wu (2007) proposed context-dependent phrasal translation lexicons by introducing context-dependent features into statistical machine translation.

Unsupervised methods using dictionaries and corpora were proposed for monolingual WSD (Ide and Veronis, 1998). They used grammatical information including parts-of-speech, syntactically related words, and co-occurring words as the clues for the WSD. Our method uses a part of the clues for bilingual WSD and word translation disambiguation.

Li and Li (2002) constructed a classifier for word translation disambiguation by using a bilingual dictionary with bootstrapping techniques. We also conducted recursive calculation by dealing with the bilingual dictionary as the seeds of the iteration.

Vickrey et al. (2005) introduced a context as a feature for a statistical MT system and they generated word-level translations. How to introduce the word-level translation disambiguation into sentence-level translation is a considerable problem.

[4] We limited the maximum number of association words for one word to 400 in descending order of their association scores because of restriction of computational resources. In future work, we may alleviate the drawback of pMI by enlarging or deleting this limitation.

6 Conclusion

In this paper, we described a method for adding information for word translation disambiguation into the bilingual lexicon, by considering the associated words that co-occur with the input word. We based our method on the following two assumptions: "parallel word associations" and "one sense per word association." We aligned word associations by using a bilingual dictionary, and constructed a correlation matrix for each word in the source language for word translation disambiguation. Experiments showed that our method was applicable for both common newspaper text and domain-specific text and for two language pairs. The Jaccard/Dice coefficients were proven to be more effective than the other metrics as word association scores. Future work includes extending our method to handle verbs as input words by introducing syntactic co-occurrence. The comparisons with other disambiguation methods and machine translation systems would strengthen the effectiveness of our method. We consider also evaluations on real NLP tasks including machine translation.

Acknowledgments

This work was partially supported by Japanese/Chinese Machine Translation Project in Special Coordination Funds for Promoting Science and Technology (MEXT, Japan).

References

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79-85.

Carpuat, Marine and Dekai Wu. 2007. Context-dependent phrasal translation lexicons for statistical machine translation. In *Proc. of Machine Translation Summit XI*, pages 73-80.

Church, Kenneth W., William Gale, Patrick Hanks and Donald Hindle. 1991. Using statistics in lexical analysis. *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, pages 115-164.

Church, Kenneth W. and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22-29.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61-74.

Ide, Nancy and Jean Veronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1-40.

Kaji, Hiroyuki and Yasutsugu Morimoto. 2002. Unsupervised word sense disambiguation using bilingual comparable corpora. In *Proc. of the 19th International Conference on Computational Linguistics*, pages 411-417.

Kruengkrai, Canasai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 513-521.

Li, Cong and Hang Li. 2002. Word translation disambiguation using bilingual bootstrapping. In *Proc. of the 40th Annual Meeting of Association for Computational Linguistics*, pages 343-351.

Rapp, Reinhard. 1995. Identifying word translations in non-parallel texts. In *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 320-322.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. of the 1st International Conference on New Methods in Natural Language Processing.*

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143-177.

Smadja, Frank, Kathleen R. McKeown and Vasileios Hatzivassiloglou. 1996. Translating collocations or bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):3-38.

Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proc. of the Conference on HLT/EMNLP*, pages 771-778.

Yarowsky, David. 1993. One sense per collocation. In *Proc. of ARPA Human Language Technology Workshop*, pages 266-271.

Construction of bilingual multimodal corpora of referring expressions in collaborative problem solving

TOKUNAGA Takenobu **IIDA Ryu** **YASUHARA Masaaki** **TERAI Asuka**
{take,ryu-i,yasuhara}@cl.cs.titech.ac.jp asuka@nm.hum.titech.ac.jp
Tokyo Institute of Technology
David MORRIS **Anja BELZ**
D.Morris@brighton.ac.uk a.s.belz@itri.brighton.ac.uk
University of Brighton

Abstract

This paper presents on-going work on constructing bilingual multimodal corpora of referring expressions in collaborative problem solving for English and Japanese. The corpora were collected from dialogues in which two participants collaboratively solved Tangram puzzles with a puzzle simulator. Extra-linguistic information such as operations on puzzle pieces, mouse cursor position and piece positions were recorded in synchronisation with utterances. The speech data was transcribed and time-aligned with the extra-linguistic information. Referring expressions in utterances that refer to puzzle pieces were annotated in terms of their spans, their referents and their other attributes. The Japanese corpus has already been completed, but the English counterpart is still undergoing annotation. We have conducted a preliminary comparative analysis of both corpora, mainly with respect to task completion time, task success rates and attributes of referring expressions. These corpora showed significant differences in task completion time and success rate.

1 Introduction

A referring expression (RE) is a linguistic device that refers to a certain object of interest (e.g. used in describing where the object is located in space). REs have attracted a great deal of attention in both language analysis and language generation research. In language analysis research, reference resolution, particularly anaphora resolution (Mitkov, 2002), has a long research history as far back as the mid-1970s (Hobbs, 1978). Much research has been conducted from both theoretical and empirical perspectives, mainly concerning the identification of antecedents or entities mentioned within the same text. This trend, targeting reference resolution in written text, is still dominant in the language analysis, perhaps because such techniques are intended for use in applications such as information extraction.

In contrast, in language generation research interest has recently shifted from the generation of one-off references to entities to generation of REs in discourse context (Belz et al., 2010) and investigating human referential behaviour in real world situations, with the aim of using such techniques in applications like human-robot interaction (Piwek, 2007; Foster et al., 2008; Bard et al., 2009).

In both analysis and generation, machine-learning approaches have come to replace rule-based approaches as the predominant research trend since the 1990s. This trend has made annotated corpora an indispensable component of research for training and evaluating proposed methods. In fact, research on reference resolution has developed significantly as a result of large scale corpora, e.g. those provided by the Message Understanding Conference (MUC)[1] and the Automatic Content Extraction (ACE)[2] project. These corpora were constructed primarily for information extraction research, thus were annotated with co-reference relations within texts. Also in the language generation community, several corpora

[1] http://www.nlpir.nist.gov/related_projects/muc/
[2] http://www.itl.nist.gov/iad/tests/ace/

Proceedings of the 8th Workshop on Asian Language Resources, pages 38–46,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

have been developed (Di Eugenio et al., 2000; Byron, 2005; van Deemter et al., 2006; Foster and Oberlander, 2007; Foster et al., 2008; Stoia et al., 2008; Spanger et al., 2009a; Belz et al., 2010). Unlike the corpora of MUC and ACE, many are collected from situated dialogues, and therefore include multimodal information (e.g. gestures and eye-gaze) other than just transcribed text (Martin et al., 2007). Foster and Oberlander (2007) emphasised that any corpus for language generation should include all possible contextual information at the appropriate granularity. Since constructing a dialogue corpus generally requires experiments for data collection, this kind of corpus tends to be small-scale compared with corpora for reference resolution.

Against this background, we have been developing multimodal corpora of referring expressions in collaborative problem-solving settings. This paper presents on-going work of constructing bilingual (English and Japanese) comparable corpora in this domain. We achieve our goal by replicating, for the English corpus, the same process of data collection and annotation as we used for our existing Japanese corpus (Spanger et al., 2009a). Our aim is to create bilingual multimodal corpora collected from dialogues in dynamic situations. From the point of view of reference analysis, our corpora contribute to augmenting the resources of multimodal dialogue corpora annotated with reference relations which have been minor in number compared to other types of text corpora. From the point of view of reference generation, our corpora contribute to increasing the resources available that can be used to further research of this kind. In addition, our corpora contribute to comparative studies of human referential behaviour in different languages

The structure of the paper is as follows. Section 2 describes the experimental set-up for data collection which was introduced in our previous work (Spanger et al., 2009a). The setting is basically the same for the construction of the English corpus. Section 3 explains the annotation scheme adopted in our corpora, followed by a description of a preliminary analysis of the corpora in section 4. Section 5 briefly mentions related work to highlight the characteristics of our corpora. Finally, Section 6 concludes the paper and looks at possible future directions.

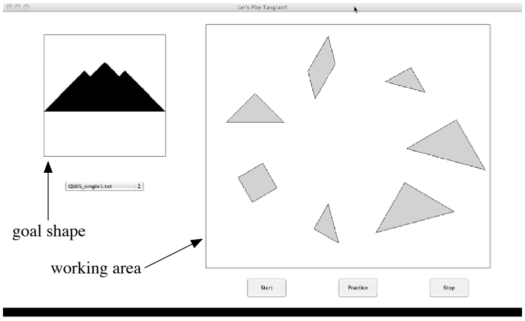


Figure 1: Screenshot of the Tangram simulator

2 Data collection

2.1 Experimental set-up

We recruited subjects in pairs of friends and colleagues. Each pair was instructed to solve Tangram puzzles collaboratively. Tangram puzzles are geometrical puzzles that originated in ancient China. The goal of a Tangram puzzle is to construct a given goal shape by arranging seven simple shapes, as shown in Figure 1. The pieces include two large triangles, a medium-sized triangle, two small triangles, a parallelogram and a square.

With the aim of recording the precise position of every piece and every action the participants made during the solving process, we implemented a Tangram simulator in which the pieces can be moved, rotated and flipped with simple mouse operations on a computer display. The simulator displays two areas: a goal shape area and a working area where the pieces can be manupulated and their movements are shown in real time.

We assigned a different role to each participant of a pair: one acted as the *solver* and the other as the *operator*. The operator has a mouse for manipulating Tangram pieces, but does not have a goal shape on the screen. The solver has a goal shape on the screen but does not have a mouse. This setting naturally leads to a situation where given a certain goal shape, the solver thinks of the necessary arrangement of the pieces and gives instructions to the operator how to move them, while the operator manipulates the pieces with the mouse according to the solver's instructions.

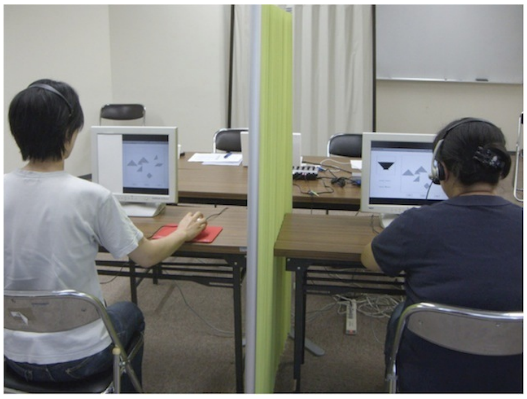


Figure 2: Picture of the experiment setting

As we mentioned in our previous study (Spanger et al., 2009a), this interaction produces frequent use of referring expressions intended to distinguish specific pieces of the puzzle. In our Tangram simulator, all pieces are of the same color, thus color is not useful in identifying a specific piece, i.e. only size and shape are discriminative object-intrinsic attributes. Instead, we can expect other attributes such as spatial relations and deictic reference to be used more often.

Each pair of participants sat side by side as shown in Figure 2. Each participant had his/her own computer display showing the shared working area. A room-divider screen was set between the solver (right side) and operator (left side) to prevent the operator from seeing the goal shape on the solver's screen, and to restrict their interaction to speech only.

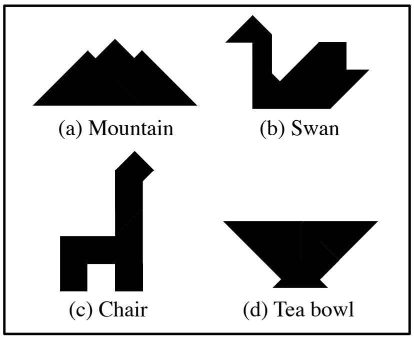


Figure 3: The goal shapes given to the subjects

Each participant pair was assigned 4 trials consisting of two symmetric and two asymmetric goal shapes as shown in Figure 3. In Cognitive Science, a wide variety of different kinds of puzzles have been employed extensively in the field of Insight Problem solving. This has been termed the "puzzle-problem approach" (Sternberg and Davidson, 1996; Suzuki et al., 2001) and in the case of physical puzzles has relatively often involved puzzle tasks of symmetric shapes like the so-called T-puzzle, e.g. (Kiyokawa and Nakazawa, 2006). In more recent work Tangram puzzles have been used as a means to study various new aspects of human problem solving approaches, including collection of of eye-gaze information (Baran et al., 2007). In order to collect data as broadly as possible in this context, we set up puzzle-problems including both symmetrical as well as asymmetrical ones as shown in Figure 3.

The participants exchanged their roles after two trials, i.e. a participant first solves a symmetric and then an asymmetric puzzle as the solver and then does the same as the operator, and vice versa. The order of the puzzle trials is the same for all pairs.

Before starting the first trial as the operator, each participant had a short training exercise in order to learn how to manipulate pieces with the mouse. The initial arrangement of the pieces was randomised every time. We set a time limit of 15 minutes for the completion of each trial (i.e. construction of the goal shape). In order to prevent the solver from getting into deep thought and keeping silent, the simulator is designed to give a hint every five minutes by showing a correct piece position in the goal shape area. After 10 minutes have passed, a second hint is provided, while the previous hint disappears. A trial ends when the goal shape is complete or the time is up. Utterances by the participants are recorded separately in stereo through headset microphones in synchronisation with the position of the pieces and the mouse operations. Piece positions and mouse actions were automatically recorded by the simulator at intervals of 10 msec.

Table 1: The ELAN Tiers of the corpus

Tier	meaning
OP-UT	utterances by the operator
SV-UT	utterances by the solver
OP-REX	referring expressions by the operator
OP-Ref	referents of OP-REX
OP-Attr	attributes of OP-REX
SV-REX	referring expressions by the solver
SV-Ref	referents of SV-REX
SV-Attr	attributes of SV-REX
Action	action on a piece
Target	the target piece of Action
Mouse	the piece on which the mouse is hovering

∗ Indentation of Tier denotes parent-child relations.

Table 2: Attributes of referring expressions

dpr	: demonstrative pronoun, e.g. “the same one”, “this”, “that”, “it”
dad	: demonstrative adjective, e.g. “that triangle”
siz	: size, e.g. “the large triangle”
typ	: type, e.g. “the square”
dir	: direction of a piece, e.g. “the triangle facing the left”.
prj	: projective spatial relation (including directional prepositions or nouns such as “right”, “left”, “above”. . .) e.g. “the triangle to the left of the square”
tpl	: topological spatial relation (including non-directional prepositions or nouns such as “near”, “middle”. . .), e.g. “the triangle near the square”
ovl	: overlap, e.g. “the small triangle under the large one”
act	: action on pieces, e.g “the triangle that you are holding now”, “the triangle that you just rotated”
cmp	: complement, e.g. “the other one”
sim	: similarity, e.g. “the same one”
num	: number, e.g. “the two triangle”
rpr	: repair, e.g. “the big, no, small triangle”
err	: obvious erroneous expression, e.g. “the square” referring to a triangle
nest	: nested expression; when a referring expression includes another referring expression, only the outermost expression is annotated with this attribute, e.g. “(the triangle to the left of (the small triangle))”
meta	: metaphorical expression, e.g. “the leg”, “the head”

2.2 Subjects and collected data

For our Japanese corpus, we recruited 12 Japanese graduate students of the Cognitive Science department, 4 females and 8 males, and split them into 6 pairs. All pairs knew each other previously and were of the same sex and approximately same age[3]. We collected 24 dialogues (4 trials by 6 pairs) of about 4 hours and 16 minutes. The average length of a dialogue was 10 minutes 40 seconds (SD = 3 minutes 18 seconds).

For the comparable English corpus, we recruited 12 native English speakers of various occupations, 6 males and 6 females. Their average age was 30. There were 6 pairs all of whom knew each other beforehand except for one pair. Whereas during the creation of the Japanese corpus we had to give extra attention to ensuring that social relationships did not have an impact on how the subjects communicated with one another, for the English corpus there was no such concern. We collected 24 dialogues (4 trials by 6 pairs) of 5 hours and 7 minutes total length. The average length of a dialogue was 12 minutes 47 seconds (SD = 3 minutes 34 seconds).

3 Annotation

The recorded speech data was transcribed and the referring expressions were annotated with the Web-based multi-purpose annotation tool SLAT (Noguchi et al., 2008)[4]. Our target expressions in this corpus are referring expressions referring to a puzzle piece or a set of puzzle pieces. We do not deal with expressions referring to a location, a part of a piece or a constructed shape. These expressions are put aside for future work. The annotation of referring expressions is threefold: (1) identification of the span of expressions, (2) identification of their referents, and (3) assignment of a set of attributes to each referring expression.

Using the multimodal annotation tool ELAN,[5] the annotations of referring expressions were then merged with extra-linguistic data recorded by the Tangram simulator. The available extra-linguistic information from the simulator consists of (1) the action on a piece, (2) the coordinates of the mouse cursor and (3) the position of each piece in the

[3] In Japan, the relationship of senior to junior or socially higher to lower placed might affect the language use. We carefully recruited pairs to avoid the effects of this social relationship such as the possible use of overly polite and indirect language, reluctance to correct mistakes etc.

[4] We did not use SLAT for English corpus annotation. Instead, ELAN was directly used for annotating referring expressions.

[5] http://www.lat-mpi.eu/tools/elan/

Table 3: Summary of trials

ID	time	success	OP-REX	SV-REX	ID	time	success	OP-REX	SV-REX
E01	15:00				J01	8:40	o	10	48
E02	15:00				J02	11:49	o	7	55
E03	15:00				J03	11:36	o	5	26
E04	15:00				J04	7:31	o	2	21
E05	15:00				J05	15:00		23	78
E06	15:00				J06	11:12	o	5	60
E07	15:00				J07	12:11	o	3	59
E08	15:00				J08	11:20	o	4	61
E09	10:39	o			J09	14:59	o	36	84
E10	15:00				J10	6:20	o	3	47
E11	15:00				J11	5:21	o	2	14
E12	8:30	o			J12	13:40	o	37	77
E13	14:33	o	8	95	J13	15:00		8	56
E14	7:27	o	1	62	J14	4:48	o	1	29
E15	14:02	o	16	127	J15	9:30	o	20	39
E16	3:57	o	1	31	J16	5:07	o	3	17
E17	13:00	o			J17	13:37	o	10	46
E18	6:40	o			J18	8:57	o	4	51
E19	15:00				J19	8:02	o	0	37
E20	12:32	o			J20	11:23	o	1	59
E21	15:00				J21	10:12	o	7	71
E22	15:00				J22	10:24	o	9	64
E23	15:00				J23	15:00		0	69
E24	5:36	o			J24	14:22	o	0	76
Ave.	12:47		6.5	78.8	Ave.	10:40		8.3	51.8
SD	3:34		7.14	41.4	SD	3:18		10.4	20.1
Total	5:06:56	10	26	315	Total	4:16:01	21	200	1,244

working area. Actions and mouse cursor positions are recorded at intervals of 10 msec, and are abstracted into (1) a time span labeled with an action symbol ("move", "rotate" or "flip") and its target piece number (1–7), and (2) a time span labeled with a piece number which is under the mouse cursor during that span. The position of pieces is updated and recorded with a timestamp when the position of any piece changes. Information about piece positions is not merged into the ELAN files and is kept in separate files. As a result, we have 11 time-aligned ELAN Tiers as shown in Table 1.

Two annotators (two of the authors) first annotated four Japanese dialogues separately and based on a discussion of discrepancies, decided on the following criteria to identify a referring expression.

- The minimum span of a noun phrase including necessary information to identify a referent is annotated. The span might include repairs with their reparandum and disfluency (Nakatani and Hirschberg, 1993) if needed.
- Demonstrative adjectives are included in expressions.
- Erroneous expressions are annotated with a special attribute.
- An expression without a definite referent (i.e. a group of possible referents or none) is assigned a referent number sequence consisting of a prefix, followed by the sequence of possible referents as its referent, if any are present.
- All expressions appearing in muttering to oneself are excluded.

Table 2 shows a list of attributes of referring expressions used in annotating the corpus.

The rest of the 20 Japanese dialogues were annotated by two of the authors and discrepancies were resolved by discussion. Four English dialogues have been annotated so far by one of the authors.

4 Preliminary corpus analysis

We have already completed the Japanese corpus, which is named REX-J (2008-08), but only 4 out of 24 dialogues have been annotated for the English counterpart (REX-E (2010-03)). Table 3 shows a summary of the trials. The horizontal lines divide the trials by pairs, "o" in the "success" column denotes that the trial was successfully completed in the time limit (15 minutes), and the "OP-REX" and "SV-REX" columns show the number of referring expressions used by the operator and the solver respectively. The following subsections describe a preliminary comparison of the English and Japanese corpora.

Table 4: Task completion time

Lang.\Shape	(a)	(b)	(c)	(d)
English	832.0	741.2	890.3	605.8
	(105.4)	(246.5)	(23.7)	(287.2)
Japanese	774.7	535.0	571.7	633.8
	(167.3)	(168.5)	(242.2)	(215.2)

* Average (SD)

4.1 Task performance

We conducted a two-way ANOVA with the task completion time as the dependent variable, and the goal shape and the language as the independent variables. Only the main effect of the language was significant ($F(1,40) = 5.82,\ p < 0.05$). Table 4 shows the average and the standard deviation of the completion time. Note that we set a time limit (15 minutes) for solving the puzzle. We considered the completion time as 15 minutes even when a puzzle was not actually solved in the time limit. We also conducted a two-way ANOVA using only the successful cases. Both main effects and their interaction were not significant.

We then conducted an ANOVA with the number of successfully solved puzzles by each pair as the dependent variable and the language as the independent variable. The main effect was significant ($F(1,10) = 6.79,\ p < 0.05$). Table 5 shows the average number of success goals per pair and the success rate with their standard deviations in parentheses.

Finally, we conducted an ANOVA with the number of pairs who succeeded in solving a goal shape as the dependent variable and the goal shape as the independent variable. The main effect was not significant.

Table 5: The number of solved trials and success rates

Lang.	solved trials	success rate [%]
Japanese	3.50 (0.55)	87.5 (13.7)
English	1.67 (1.63)	41.7 (40.8)

* Average (SD)

In summary, we found a difference in the task performance between the languages in terms of the task completion time and the success rate, but no difference among the goal shapes. This difference could be explained by the diversity of the subjects rather than the difference of languages. The Japanese subject group consisted of university graduate students from the same department (Cognitive Science) and roughly of the same age (Average = 23.3, SD = 1.5). In contrast, the English subjects have diverse backgrounds (e.g. high school students, university faculty, writer, programmer, etc.) and age (Average = 30.8, SD = 11.7). In addition, a familiarity with this kind of geometric puzzle might have some effect. However, we collected a familiarity with the puzzle only from the English subjects, we could not conduct further analysis on this viewpoint. Anyhow, in this respect, the independent variable should have been named "subject group" instead of "language".

4.2 Referring expressions

It is important to note that since we have only completed the annotation of four dialogs, all by one pair of subjects, our analyses of referring expressions are tentative and pending further analysis.

We have 200 and 1,243 referring expressions by the operator and the solver respectively, 1,444 in total in the 24 Japanese dialogues. On the other hand we have 26 (operator) and 315 (solver) referring expressions in 4 English dialogues. The average number of referring expressions per dialogue in Table 3 suggests that English subjects use more referring expressions than Japanese subjects. Since we have only the data from a single pair, we cannot say whether this tendency applies to the other pairs. We cannot draw a decisive conclusion until we complete the annotation of the English corpus.

Table 6 shows the total frequencies of the attributes and their frequencies per dialogue. The table gives us an impression of significantly frequent use of demonstrative pronouns (dpr) by the

Table 6: Comparison of attribute distribution

	English		Japanese	
	(4 dialogues)		(24 dialogues)	
attribute	frq	frq/dlg	frq	frq/dlg
dpr	226	56.5	678	28.3
dad	29	7.3	178	7.4
siz	68	17.0	288	12.0
typ	103	25.8	655	27.3
dir	0	0	7	0.3
prj	10	2.5	141	5.9
tpl	4	1	9	0.4
ovl	0	0	2	0.1
act	5	1.3	103	4.3
cmp	17	4.3	33	1.4
sim	0	0	7	0.3
num	22	5.5	35	1.5
rpr	0	0	1	0
err	0	0	1	0
nest	1	0.3	31	1.3
meta	1	0.3	6	0.3

English subjects. The Japanese subjects use more attributes of projective spatial relations (prj) and actions on the referent (act).[6] The English subjects use more complement attributes (cmp) as well as more number attributes (num).

5 Related work

Over the last decade, with a growing recognition that referring expressions frequently appear in collaborative task dialogues (Clark and Wilkes-Gibbs, 1986; Heeman and Hirst, 1995), a number of corpora have been constructed to study the nature of their use. This tendency also reflects the recognition that this area yields both challenging research topics as well as promising applications such as human-robot interaction (Foster et al., 2008; Kruijff et al., 2010).

The COCONUT corpus (Di Eugenio et al., 2000) was collected from keyboard-dialogs between two participants, who worked together on a simple 2-D design task, buying and arranging furniture for two rooms. The COCONUT corpus is limited in annotations which describe symbolic object information such as object intrinsic attributes and location in discrete co-ordinates. As an initial work of constructing a corpus for collaborative tasks, the COCONUT corpus can be characterised as having a rather simple domain as well as limited annotation.

The QUAKE corpus (Byron, 2005) and its successor, the SCARE corpus (Stoia et al., 2008) deal with a more complex domain, where two participants collaboratively play a treasure hunting game in a 3-D virtual world. Despite the complexity of the domain, the participants were only allowed limited actions, e.g. moving step forward, pushing a button etc.

As a part of the JAST project, the Joint Construction Task (JCT) corpus was created based on dialogues in which two participants constructed a puzzle (Foster et al., 2008). The setting of the experiment is quite similar to ours except that both participants have even roles. Since our main concern is referring expressions, we believe our asymmetric setting elicits more referring expressions than the symmetric setting of the JCT corpus.

In contrast to these previous corpora, our corpora record a wide range of information useful for analysis of human reference behaviour in situated dialogue. While the domain of our corpora is simple compared to the QUAKE and SCARE corpora, we allowed a comparatively large flexibility in the actions necessary for achieving the goal shape (i.e. flipping, turning and moving of puzzle pieces at different degrees), relative to the complexity of the domain. Providing this relatively larger freedom of actions to the participants together with the recording of detailed information allows for research into new aspects of referring expressions.

As for a multilingual aspect, all the above corpora are English. There have been several recent attempts at collecting multilingual corpora in situated domains. For instance, (Gargett et al., 2010) collected German and English corpora in the same setting. Their domain is similar to the QUAKE corpus. Van der Sluis et al. (2009) aim at a comparative study of referring expressions between English and Japanese. Their domain is still static at the moment. Our corpora aim at dealing with the dynamic nature of situated dialogues between very different languages, English and Japanese.

[6] We called such expressions as *action-mentioning expressions* (AME) in our previous work.

Table 7: The REX-J corpus family

name	puzzle	#pairs	#dialg.	#valid	status
T2008-08	Tangram	6	24	24	completed
T2009-03	Tangram	10	40	16	completed
T2009-11	Tangram	10	36	27	validating
N2009-11	Tangram	5	20	8	validating
P2009-11	Polyomino	7	28	24	annotating
D2009-11	2-Tangram	7	42	24	annotating

6 Conclusion and future work

This paper presented an overview of our English-Japanese bilingual multimodal corpora of referring expressions in a collaborative problem solving setting. The Japanese corpus was completed and has already been used for research (Spanger et al., 2009b; Spanger et al., 2010; Iida et al., 2010), but the English counterpart is still undergoing annotation. We have also presented a preliminary comparative analysis of these corpora in terms of the task performance and usage of referring expressions. We found a significant difference of the task performance, which could be attributed to the difference in diversity of subjects. We have tentative results on the usage of referring expressions, since only four English dialogues are available at the moment.

The data collection experiments were conducted in August 2008 for Japanese and in March 2010 for English. Between these periods, we conducted various data collections to build different types of Japanese corpora (March, 2009 and November 2009). These experiments involve capturing eye-gaze information of participants during problem solving, and introducing variants of puzzles (Polyomino, Double Tangram and Tangram without any hints[7]). They are also under preparation for publication. Table 7 gives an overview of the REX-J corpus family, where "#valid" denotes the number of dialogues with valid eye-gaze data. Eye-gaze data is difficult to capture cleanly throughout a dialogue. We discarded dialogues in which eye-gaze was captured successfully less than 70% of the total time of the dialogue. Namely, we annotated or will annotate dialogues with validated eye-gaze data only.

These corpora enable research on utilising eye-gaze information in reference resolution and generation, and evaluation in different tasks (puzzles) as well. We are planning to distribute the REX-J corpus family through GSK (Language Resources Association in Japan)[8], and the REX-E corpus from both University of Brighton and GSK.

References

Baran, Bahar, Berrin Dogusoy, and Kursat Cagiltay. 2007. How do adults solve digital tangram problems? Analyzing cognitive strategies through eye tracking approach. In *HCI International 2007 - 12th International Conference - Part III*, pages 555–563.

Bard, Ellen Gurman, Robin Hill, Manabu Arai, and Mary Ellen Foster. 2009. Accessibility and attention in situated dialogue: Roles and regulations. In *Proceedings of the Workshop on Production of Referring Expressions Pre-CogSci 2009*.

Belz, Anja, Eric Kow, Jette Viethen, and Albert Gatt. 2010. Referring expression generation in context: The GREC shared task evaluation challenges. In Krahmer, Emiel and Mariët Theune, editors, *Empirical Methods in Natural Language Generation*, volume 5980 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/Heidelberg.

Byron, Donna K. 2005. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. Technical report, Department of Computer Science and Enginerring, The Ohio State University.

Clark, H. Herbert. and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Di Eugenio, Barbara, Pamela W. Jordan, Richmond H. Thomason, and Johanna. D. Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human-Computer Studies*, 53(6):1017–1076.

Foster, Mary Ellen and Jon Oberlander. 2007. Corpus-based generation of head and eyebrow motion for an embodied conversational agent. *Language Resources and Evaluation*, 41(3–4):305–323, December.

Foster, Mary Ellen, Ellen Gurman Bard, Markus Guhe, Robin L. Hill, Jon Oberlander, and Alois Knoll. 2008. The roles of haptic-ostensive referring expressions in cooperative, task-based human-robot dialogue. In *Proceedings of 3rd Human-Robot Interaction*, pages 295–302.

[7] N2009-11 in Table 7

[8] http://www.gsk.or.jp/index_e.html

Gargett, Andrew, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The give-2 corpus of giving instructions in virtual environments. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*, pages 2401–2406.

Heeman, Peter A. and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21:351–382.

Hobbs, Jerry R. 1978. Resolving pronoun references. *Lingua*, 44:311–338.

Iida, Ryu, Shumpei Kobayashi, and Takenobu Tokunaga. 2010. Incorporating extra-linguistic information into reference resolution in collaborative task dialogue. In *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics*, pages 1259–1267.

Kiyokawa, Sachiko and Midori Nakazawa. 2006. Effects of reflective verbalization on insight problem solving. In *Proceedings of 5th International Conference of the Cognitive Science*, pages 137–139.

Kruijff, Geert-Jan M., Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, and Ivana Kruijff-Korbayova. 2010. Situated dialogue processing for human-robot interaction. In *Cognitive Systems: Final report of the CoSy project*, pages 311–364. Springer-Verlag.

Martin, Jean-Claude, Patrizia Paggio, Peter Kuehnlein, Rainer Stiefelhagen, and Fabio Pianesi. 2007. Special issue on Mulitmodal corpora for modeling human multimodal behaviour. *Language Resources and Evaluation*, 41(3-4).

Mitkov, Ruslan. 2002. *Anaphora Resolution*. Longman.

Nakatani, Christine and Julia Hirschberg. 1993. A speech-first model for repair identification and correction. In *Proceedings of 31th Annual Meeting of ACL*, pages 200–207.

Noguchi, Masaki, Kenta Miyoshi, Takenobu Tokunaga, Ryu Iida, Mamoru Komachi, and Kentaro Inui. 2008. Multiple purpose annotation using SLAT – Segment and link-based annotation tool. In *Proceedings of 2nd Linguistic Annotation Workshop*, pages 61–64.

Piwek, Paul L. A. 2007. Modality choise for generation of referring acts. In *Proceedings of the Workshop on Multimodal Output Generation (MOG 2007)*, pages 129–139.

Spanger, Philipp, Masaaki Yasuhara, Ryu Iida, and Takenobu Tokunaga. 2009a. A Japanese corpus of referring expressions used in a situated collaboration task. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 110 – 113.

Spanger, Philipp, Masaaki Yasuhara, Ryu Iida, and Takenobu Tokunaga. 2009b. Using extra linguistic information for generating demonstrative pronouns in a situated collaboration task. In *Proceedings of PreCogSci 2009: Production of Referring Expressions: Bridging the gap between computational and empirical approaches to reference*.

Spanger, Philipp, Ryu Iida, Takenobu Tokunaga, Asuka Teri, and Naoko Kuriyama. 2010. Towards an extrinsic evaluation of referring expressions in situated dialogs. In Kelleher, John, Brian Mac Namee, and Ielka van der Sluis, editors, *Proceedings of the Sixth International Natural Language Generation Conference (INGL 2010)*, pages 135–144.

Sternberg, Robert J. and Janet E. Davidson, editors. 1996. *The Nature of Insight*. The MIT Press.

Stoia, Laura, Darla Magdalene Shockley, Donna K. Byron, and Eric Fosler-Lussier. 2008. SCARE: A situated corpus with annotated referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 28–30.

Suzuki, Hiroaki, Keiga Abe, Kazuo Hiraki, and Michiko Miyazaki. 2001. Cue-readiness in insight problem-solving. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pages 1012 – 1017.

van Deemter, Kees, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 130–132.

van der Sluis, Ielka, Junko Nagai, and Saturnino Luz. 2009. Producing referring expressions in dialogue: Insights from a translation exercise. In *Proceedings of PreCogSci 2009: Production of Referring Expressions: Bridging the gap between computational and empirical approaches to reference*.

Labeling Emotion in Bengali Blog Corpus – A Fine Grained Tagging at Sentence Level

Dipankar Das
Department of Computer Science
& Engineering,
Jadavpur University
dipankar.dipnil2005@gmail.com

Sivaji Bandyopadhyay
Department of Computer Science
& Engineering,
Jadavpur University
sivaji_cse_ju@yahoo.com

Abstract

Emotion, the private state of a human entity, is becoming an important topic in Natural Language Processing (*NLP*) with increasing use of search engines. The present task aims to manually annotate the sentences in a web based Bengali blog corpus with the emotional components such as emotional expression (word/phrase), intensity, associated holder and topic(s). Ekman's six emotion classes (*anger, disgust, fear, happy*, *sad* and *surprise*) along with three types of intensities (*high*, *general* and *low*) are considered for the sentence level annotation. Presence of discourse markers, punctuation marks, negations, conjuncts, reduplication, rhetoric knowledge and especially emoticons play the contributory roles in the annotation process. Different types of fixed and relaxed strategies have been employed to measure the agreement of the sentential emotions, intensities, emotional holders and topics respectively. Experimental results for each emotion class at word level on a small set of the whole corpus have been found satisfactory.

1 Introduction

Human emotion described in texts is an important cue for our daily communication but the identification of emotional state from texts is not an easy task as emotion is not open to any objective observation or verification (Quirk *et al.*, 1985). Emails, weblogs, chat rooms, online forums and even twitter are considered as the affective communication substrates to analyze the reaction of emotional catalysts. Among these media, blog is one of the communicative and informative repository of text based emotional contents in the Web 2.0 (Lin *et al.*, 2007).

Rapidly growing web users from multilingual communities focus the attention to improve the multilingual search engines on the basis of sentiment or emotion. Major studies on Opinion Mining and Sentiment Analyses have been attempted with more focused perspectives rather than fine-grained emotions. The analyses of emotion or sentiment require some basic resource. An emotion-annotated corpus is one of the primary ones to start with.

The proposed annotation task has been carried out at sentence level. Three annotators have manually annotated the Bengali blog sentences retrieved from a web blog archive[1] with Ekman's six basic emotion tags (*anger* (A)*, disgust* (D)*, fear* (F)*, happy* (H), *sad* (Sa) and *surprise* (Su)). The emotional sentences are tagged with three types of intensities such as *high*, *general* and *low*. The sentences of non-emotional (*neutral*) and multiple (*mixed*) categories are also identified. The identification of emotional words or phrases and fixing the scope of emotional expressions in the sentences are carried out in the present task. Each of the emoticons is also considered as individual emotional expressions. The emotion holder and relevant topics associated with the emotional expressions are annotated considering the punctuation marks, conjuncts, rhetorical structures and other discourse information. The knowledge of rhetorical structure helps in removing the subjective discrepancies from the

[1] www.amarblog.com

Proceedings of the 8th Workshop on Asian Language Resources, pages 47–55,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

writer's point of view. The annotation scheme is used to annotate 123 blog posts containing 4,740 emotional sentences having single emotion tag and 322 emotional sentences for mixed emotion tagss along with 7087 *neutral* sentences in Bengali. Three types of standard agreement measures such as Cohen's *kappa* (κ) (Cohen, 1960; Carletta, 1996), Measure of Agreement on Set-valued Items (*MASI*) (Passonneau, 2004) and *agr* (Wiebe *et al*., 2005) metrics are employed for annotating the emotion related components. The relaxed agreement schemes like MASI and *agr* are specially considered for fixing the boundaries of emotional expressions and topic spans in the emotional sentences. The inter annotator agreement of some emotional components such as sentential emotions, holders, topics show satisfactory performance but the sentences of mixed emotion and intensities of *general* and *low* show the disagreement. A preliminary experiment for word level emotion classification on a small set of the whole corpus yielded satisfactory results.

The rest of the paper is organized as follows. Section 2 describes the related work. The annotation of emotional expressions, sentential emotion and intensities are described in Section 3. In Section 4, the annotation scheme for emotion holder is described. The issues of emotional topic annotation are discussed in Section 5. Section 6 describes the preliminary experiments carried out on the annotated corpus. Finally, Section 7 concludes the paper.

2 Related Work

One of the most well known tasks of annotating the private states in texts is carried out by (Wiebe *et al*., 2005). They manually annotated the private states including emotions, opinions, and sentiment in a 10,000-sentence corpus (the MPQA corpus) of news articles. The opinion holder information is also annotated in the MPQA corpus but the topic annotation task has been initiated later by (Stoyanov and Cardie, 2008a). In contrast, the present annotation strategy includes the fine-grained emotion classes and specially handles the emoticons present in the blog posts.

(Alm *et al*., 2005) have considered eight emotion categories (angry, disgusted, fearful, happy, sad, positively surprised, negatively surprised) to accomplish the emotion annotation task at sentence level. They have manually annotated 1580 sentences extracted from 22 Grimms' tales. The present approach discusses the issues of annotating unstructured blog text considering rhetoric knowledge along with the attributes, e.g. negation, conjunct, reduplication etc.

Mishne (2005) experimented with mood classification in a blog corpus of 815,494 posts from Livejournal (http://www.livejournal.com), a free weblog service with a large community. (Mihalcea and Liu, 2006) have used the same data source for classifying the blog posts into two particular emotions – *happiness* and *sadness*. The blog posts are self-annotated by the blog writers with *happy* and *sad* mood labels. In contrast, the present approach includes Ekman's six emotions, emotion holders and topics to accomplish the whole annotation task.

(Neviarouskaya *et al*., 2007) collected 160 sentences labeled with one of the nine emotions categories (anger, disgust, fear, guilt, interest, joy, sadness, shame, and surprise) and a corresponding intensity value from a corpus of online diary-like blog posts. On the other hand, (Aman and Szpakowicz, 2007) prepare an emotion-annotated corpus with a rich set of emotion information such as category, intensity and word or phrase based expressions. The present task considers all the above emotion information during annotation. But, the present annotation task additionally includes the components like emotion holder, single or multiple topic spans.

The emotion corpora for Japanese were built for recognizing emotions (Tokuhisa *et al*., 2008). An available emotion corpus in Chinese is Yahoo!'s Chinese news (http://tw.news.yahoo.com), which is used for Chinese emotion classification of news readers (Lin, *et al*., 2007). The manual annotation of eight emotional categories (expect, joy, love, surprise, anxiety, sorrow, angry and hate) along with intensity, holder, word/phrase, degree word, negative word, conjunction, rhetoric, punctuation and other linguistic expressions are carried out at sentence, paragraph as well as document level on 1,487 Chinese blog documents (Quan and Ren, 2009). In addition to the above emotion entities, the present approach also includes the annotation of single or multiple emotion topics in a target span.

Recent study shows that non-native English speakers support the growing use of the Internet[2]. This raises the demand of linguistic resources for languages other than English. Bengali is the fifth popular language in the World, second in India and the national language in Bangladesh but it is less computerized compared to English. To the best of our knowledge, at present, there is no such available corpus that is annotated with detailed linguistic expressions for emotion in Bengali or even for other Indian languages. Thus we believe that this corpus would help the development and evaluation of emotion analysis systems in Bengali.

3 Emotion Annotation

Random collection of 123 blog posts containing a total of 12,149 sentences are retrieved from Bengali web blog archive[3] (especially from comics, politics, sports and short stories) to prepare the corpus. No prior training was provided to the annotators but they were instructed to annotate each sentence of the blog corpus based on some illustrated samples of the annotated sentences. Specially for annotating the emotional expressions and topic(s) in emotional sentences, the annotators are free in selecting the texts spans. This annotation scheme is termed as relaxed scheme. For other emotional components, the annotators are given items with fixed text spans and instructed to annotation the items with definite tags.

3.1 Identifying Emotional Expressions for Sentential Emotion and Intensity

The identification of emotion or affect affixed in the text segments is a puzzle. But, the puzzle can be solved partially using some lexical clues (e.g. discourse markers, punctuation marks (*sym*), negations (*NEG*), conjuncts (*CONJ*), reduplication (*Redup*)), structural clues (e.g. rhetoric and syntactic knowledge) and especially some direct affective clues (e.g. emoticons (*emo_icon*)). The identification of structural clues indeed requires the identification of lexical clues.

Rhetorical Structure Theory (RST) describes the various parts of a text, how they can be arranged and connected to form a whole text (Azar, 1999). The theory maintains that consecutive discourse elements, termed *text spans*, which can be in the form of clauses, sentences, or units larger than sentences, are related by a relatively small set (20–25) of *rhetorical relations* (Mann and Thompson, 1988). RST distinguishes between the part of a text that realizes the primary goal of the writer, termed as *nucleus*, and the part that provides supplementary material, termed *satellite.* The separation of *nucleus* from *satellite* is done based on punctuation marks (, ! @?), emoticons, discourse markers (যেহেতু *jehetu* [as], যেমন *jemon* [e.g.], কারণ *karon* [because], মানে *mane* [means]), conjuncts (এবং *ebong* [and], কিন্তু *kintu* [but], অথবা *athoba* [or]), causal verbs (ঘটায় *ghotay* [caused]) if they are explicitly specified in the sentences.

Use of emotion-related words is not the sole means of expressing emotion. Often a sentence, which otherwise may not have an emotional word, may become emotion bearing depending on the context or underlying semantic meaning (Aman and Szpakowicz, 2007). An empirical analysis of the blog texts shows two types of emotional expressions. The first category contains explicitly stated emotion word (*EW*) or phrases (*EP*) mentioned in the *nucleus* or in the *satellite.* Another category contains the implicit emotional clues that are identified based on the context or from the metaphoric knowledge of the expressions. Sometimes, the emotional expressions contain direct emotion words (*EW*) (কৌতুক *koutuk* [joke], আনন্দ *ananda* [happy], আশ্চর্য *ashcharjyo* [surprise]), reduplication (*Redup*) (সন্দ সন্দ *sanda sanda* [doubt with fear], question words (*EW_Q*) (কি *ki* [what], কবে *kobe* [when]), colloquial words (ক্ষ্যামা *kshyama* [perdon]) and foreign words (থ্যাংকু *thanku* [thanks], গোস্যা *gossya* [anger]). On the other hand, the emotional expressions contain indirect emotion words e.g. proverbs, idioms (তাসের ঘর *taser ghar* [weakly built], গৃহদাহ *grrihadaho* [family disturbance]) and emoticons (☺,☹).

[2] http://www.internetworldstats.com/stats.htm
[3] www.amarblog.com

A large number of emoticons (*emo_icon*) present in the Bengali blog texts vary according to their emotional categories and slant. Each of the emoticons is treated as individual emotional expression and its corresponding intensity is set based on the image denoted by the emoticon. The labeling of the emoticons with Ekman's six emotion classes is verified through the inter-annotator agreement that is considered for emotion expressions.

The intensifiers (খুব *khub* [too much/very], অনেক *anek* [huge/large], ভীষণ *bhishon* [heavy/too much]) associated with the emotional phrases are also acknowledged in annotating sentential intensities. As the intensifiers depend solely on the context, their identification along with the effects of negation and conjuncts play a role in annotating the intensity. Negations (না *na* [no], নয় *noy* [not]) and conjuncts freely occur in the sentence and change the emotion of the sentence. For that very reason, a crucial analysis of negation and conjuncts is carried out both at intra and inter phrase level to obtain the sentential emotions and intensities. An example set of the annotated blog corpus is shown in Figure 1.

<ES_Sa><holder>গেদুচাচা:</holder> ভাইও <sym>!</sym> <EW_D>বেজার</EW_D> হইছেন?</ES_Sa>
<ES_A><ES_Su>তিনি <EW_Su><EW_Q> কি</EW_Q></EW_Su> বুঝলেন <EW_Su><EW_Q> কে</EW_Q></EW_Su> জানে<EW_Su>!</EW_Su> <Redup><EW_A>রেগেমেগে</EW_A></Redup> ভিতরে চলে গেলেন। আমাকে <EW_F>বোধহয় </EW_F> চিনতে পারেন <NEG>নি</NEG>।</ES_Su></ES_A>
<ES_H>আপনার <topic>কবিতাটা</topic> পড়তএ গিয়া এই <EW_H>কউতুকটা</EW_H> মনে পড়ছিলো </ES_H>

"Figure 1. Annotated sample of the corpus"

3.2 Agreement of Sentential Emotion and Intensity

Three annotators identified as A1, A2 and A3 have used an open source graphical tool to carry out the annotation [4]. As the Ekman's emotion classes and intensity types belong to some definite categories, the annotation agreement for emotion and intensities are measured using standard Cohen's *kappa* coefficient (κ) (Cohen, 1960). The annotation agreement for emoticons is also measured using the *kappa* metric. It is a statistical measure of inter-rater agreement for qualitative (categorical) items. It measures the agreement between two raters who separately classify items into some mutually exclusive categories.

The agreement of classifying sentential intensities into three classes (*high*, *general* and *low*) is also measured using kappa (κ). The intensities of mixed emotional sentences are also considered. Agreement results of emotional, non-emotional and mixed sentences, emoticons, along with results for each emotion class, intensity types are shown in Table 1. Sentential emotions with *happy*, *sad* or *surprise* classes produce comparatively higher *kappa* coefficient than the other emotion classes as the emotional expressions of these types were explicitly specified in the blog texts. It has been observed that the emotion pairs such as "sad-anger" and "anger-disgust" often cause the trouble in distinguishing the emotion at sentence level. Mixed emotion category, *general* and *low* intensity types give poor agreement results as expected. Instead of specifying agreement results of emoticons for each emotion class, the average results for the three annotation sets are shown in Table 1.

3.3 Agreement of Emotional Expressions

Emotional expressions are words or strings of words that are selected by the annotators. The agreement is carried out between the sets of text spans selected by the two annotators for each of the emotional expressions. As there is no fixed category in this case, we have employed two different strategies instead of kappa (κ) to calculate the agreement between annotators. Firstly, we chose the measure of agreement on set-valued items (MASI) (Passonneau, 2006) that was used for measuring agreement on co reference annotation (Passonneau, 2004) and in the semantic and pragmatic annotation (Passonneau, 2006). MASI is a distance between sets whose value is 1 for identical sets, and 0 for disjoint sets. For sets A and B it is defined as: MASI = J * M, where the Jaccard metric is:

[4] http://gate.ac.uk/gate/doc/releases.html

$J = |A \cap B| / |A \cup B|$

Monotonicity (M) is defined as,

$$1, if A = B$$

$$2/3, if A \subset B or B \subset A$$

$$1/3, if A \cap B \neq \phi, A - B \neq \phi, and B - A \neq \phi$$

$$0, if A \cap B = \phi$$

Secondly, the annotators will annotate different emotional expressions by identifying the responsible text anchors and the agreement is measured using *agr* metric (Wiebe *et al.*, 2005). If *A* and *B* are the sets of anchors annotated by annotators *a* and *b*, respectively, *agr* is a directional measure of agreement that measures what proportion of *a* was also marked by *b*. Specifically, we compute the agreement of *b* to *a* as:

$$agr(a \| b) = \frac{|A matching B|}{|A|}$$

The *agr (a|| b)* metric corresponds to the recall if *a* is the gold standard and *b* the system, and to precision, if *b* is the gold standard and *a* is the system. The results of two agreement strategies for each emotion class are shown in Table 1. The annotation agreement of emotional expressions produces slightly less values for both *kappa* and *agr*. It leads to the fact that the relaxed annotation scheme that is provided for fixing the boundaries of the expressions causes the disagreements.

4 Identifying Emotion Holder

The source or holder of an emotional expression is the speaker or writer or experiencer. The main criteria considered for annotating emotion holders are based on the nested source hypothesis as described in (Wiebe *et al.*, 2005). The structure of Bengali blog corpus (as shown in Figure 2) helps in the holder annotation process. Sometimes, the comments of one blogger are annotated by other bloggers in the blog posts. Thus the holder annotation task in user comments sections was less cumbersome than annotating the holders inscribed in the topic section.

Classes (# Sentences or Instances)	Agreement (pair of annotators)			
	A1-A2	A2-A3	A1-A3	Avg.
Emotion / Non-Emotion (5,234/7,087)	0.88	0.83	0.86	0.85
Happy (804)	0.79	0.72	0.83	0.78
Sad (826)	0.82	0.75	0.72	0.76
Anger (765)	0.75	0.71	0.69	0.71
Disgust (766)	0.76	0.69	0.77	0.74
Fear (757)	0.65	0.61	0.65	0.63
Surprise (822)	0.84	0.82	0.85	0.83
Mixed (322)	0.42	0.21	0.53	0.38
High (2,330)	0.66	0.72	0.68	0.68
General (1,765)	0.42	0.46	0.48	0.45
Low (1345)	0.21	0.34	0.26	0.27
Emoticons w.r.t six Emotion Classes (678)	0.85	0.73	0.84	0.80
Emoticons w.r.t three Intensities	0.72	0.66	0.63	0.67
Emotional Expressions (7,588) [*MASI*]	0.64	0.61	0.66	0.63
Emotional Expressions (7,588) [*agr*]	0.67	0.63	0.68	0.66

Table 1: Inter-Annotator Agreements for sentence level Emotions, Intensities, Emoticons and Emotional Expressions

```
 -<DOC docid = xyz>
      -<Topic>…. </Topic>
      -<User Comments>
                -<U uid=1>… </U>
                -<U uid=2>… </U>
                -<U uid=3>….
                -<U uid=1>… </U> …</U>…
        </User Comments>
  </DOC>
```

"Figure. 2. General structure of a blog document"

Prior work in identification of opinion holders has sometimes identified only a single opinion per sentence (Bethard *et al.*, 2004), and sometimes several (Choi *et al.*, 2005). As the blog corpus has sentence level emotion annotations, the former category is adopted. But, it is observed that the long sentences contain more than one emotional expression and hence associated with multiple emotion holders (*EH*). All probable emotion holders of a sentence are stored in an anchoring vector successively according to their order of occurrence.

The annotation of emotion holder at sentence level requires the knowledge of two basic constraints (*implicit* and *explicit*) separately. The *explicit* constraints qualify single prominent emotion holder that is directly involved with the emotional expression whereas the *implicit* constraints qualify all direct and indirect nested sources as emotion holders. For example, in the following Bengali sentences, the pattern shown in **bold** face denotes the emotion holder. In the second example, the appositive case (e.g. রামের সুখ (*Ram's pleasure*)) is also identified and placed in the vector by removing the inflectional suffix (-এর in this case). Example 2 and Example 3 contain the emotion holders ***রাম*** (*Ram*) and ***নাসরিন সুলতানা*** (*Nasrin Sultana*) based on *implicit* constraints.

Example 1. *EH_Vector*: < **সায়ণ** >
সায়ণ ভীষণ আনন্দ অনুভব
(**Sayan**) (bhishon) (anondo) (anubhob)
করেছিল
(korechilo)
Sayan felt very happy.

Example 2. *EH_Vector*: < রাশেদ, ***রাম*** >
রাশেদ অনুভব করেছিল যে **রামের**
(*Rashed*) (anubhob) (korechilo) (je) (**Ramer**)
সুখ অন্তহীন
(sukh) (antohin)
Rashed felt that Ram's pleasure is endless.

Example 3. *EH_Vector*: <গেদু চাচা, ***নাসরিন সুলতানা***
>
গেদু চাচা বলে : আমি **নাসরিন সুলতানার**
(GeduChaCha) (bole) (ami) (**Nasrin Sultanar**)
দুঃখের কথাতে কেঁদে ফেলি।
(dookher) (kathate) (kende) (feli)
Gedu Chacha says: No my sister, I fall into cry on the sad speech of Nasrin Sultana

4.1 Agreement of Emotion Holder Annotation

The emotion holders containing multi word Named Entities (*NEs*) are assumed as single emotion holders. As there is no agreement discrepancy in selecting the boundary of the single or multiple emotion holders, we have used the standard metric, Cohen's *kappa* (κ) for measuring the inter-annotator agreement. Each of the elementary emotion holders in an anchoring vector is treated as a separate emotion holder and the agreement between two annotators is carried out on each separate entity. It is to be mentioned that the anchoring vectors provided by the two annotators may be disjoint.

To emphasize the fact, a different technique is employed to measure the annotation agreement. If **X** is a set of emotion holders selected by the first annotator and **Y** is a set of emotion holders selected by the second annotator for an emotional sentence containing multiple emotion holders, inter-annotator agreement **IAA** for that sentence is equal to quotient of number of emotion holders in **X** and **Y** intersection divided by number of emotion holders in **X** and **Y** union:

IAA = X ∩ Y / X U Y

Two types of agreement results per emotion class for annotating emotion holders (*EH*) are shown in Table 2. Both types of agreements have been found satisfactory and the difference between the two agreement types is significantly less. The small difference indicates the minimal error involved in the annotation process. It is found that the agreement is highly moderate in case of single emotion holder, but is less in case of multiple holders. The disagreement occurs mostly in the case of satisfying the implicit constrains but some issues are resolved by mutual understanding.

5 Topic Annotation and Agreement

Topic is the real world object, event, or abstract entity that is the primary subject of the opinion as intended by the opinion holder (Stoyanov and Cardie, 2008). They mention that the topic identification is difficult within the single target span of the opinion as there are multiple potential topics, each identified with its own topic span and the topic of an opinion depends on the context in which its associated opinion expression occurs. Hence, the actual challenge lies on identification of the topics spans from the emotional sentences. The writer's emotional intentions in a sentence are reflected in the target span by mentioning one or more topics that are related to the emotional expressions. Topics are generally distributed in different text spans of writer's text and can be distinguished by capturing the rhetorical structure of the text.

Emotion Classes [# Sentences, # Holders]	Agreement between pair of annotators (κ) [IAA]			
	A1-A2	A2-A3	A1-A3	Avg.
Happy [804, 918]	(0.87) [0.88]	(0.79) [0.81]	(0.76) [0.77]	(0.80) [0.82]
Sad [826, 872]	(0.82) [0.81]	(0.85) [0.83]	(0.78) [0.80]	(0.81) [0.81]
Anger [765,780]	(0.80) [0.79]	(0.75) [0.73]	(0.74) [0.71]	(0.76) [0.74]
Disgust [766, 770]	(0.70) [0.68]	(0.72) [0.69]	(0.83) [0.84]	(0.75) [0.73]
Fear [757, 764]	(0.85) [0.82]	(0.78) [0.77]	(0.79) [0.81]	(0.80) [0.80]
Surprise [822, 851]	(0.78) [0.80]	(0.81) [0.79]	(0.85) [0.83]	(0.81) [0.80]

Table 2: Inter-Annotator Agreement for Emotion Holder Annotation

In blog texts, it is observed that an emotion topic can occur in *nucleus* as well as in *satellite*. Thus, the whole sentence is assumed as the scope for the potential emotion topics. The text spans containing emotional expression and emotion holder can also be responsible for being the candidate seeds of target span. In Example 3 of Section 4, the target span (নাসরিন সুলতানার দুঃখের কথাতে '*sad speech of Nasrin Sultana*') contains emotion holder (নাসরিন সুলতানার '*Nasrin Sultana*') as well as the emotional expression (দুঃখের কথাতে '*sad speech*') For that reason, the annotators are instructed to consider the whole sentence as their target span and to identify one or more topics related to the emotional expressions in that sentence.

As the topics are multi word components or string of words, the scope of the individual topics inside a target span is hard to identify. To accomplish the goal, we have not used the standard metrics Cohen's *kappa* (κ). We employed MASI and *agr* metric (as mentioned in Section 3) for measuring the agreement of topic spans annotation. The emotional sentences containing single emotion topic has shown less disagreement than the sentences that contain multiple topics. It is observed that the agreement for annotating target span is (≈ 0.9). It means that the annotation is almost satisfactory. But, the disagreement occurs in annotating the boundaries of topic spans. The inter-annotator agreement for each emotion class is shown in Table 3. The selection of emotion topic from other relevant topics causes the disagreement.

Emotion Classes [# Sentences, # topics]	Agreement between Pair of annotators (MASI) [*agr*]			
	A1-A2	A2-A3	A1-A3	Avg
Happy [804, 848]	(0.83) [0.85]	(0.81) [0.83]	(0.79) [0.82]	(0.81) [0.83]
Sad [826, 862]	(0.84) [0.86]	(0.77) [0.79]	(0.81) [0.83]	(0.80) [0.82]
Anger [765,723]	(0.80) [0.78]	(0.81) [0.78]	(0.86) [0.84]	(0.82) [0.80]
Disgust [766, 750]	(0.77) [0.76]	(0.78) [0.74]	(0.72) [0.70]	(0.75) [0.73]
Fear [757, 784]	(0.78) [0.79]	(0.77) [0.80]	(0.79) [0.81]	(0.78) [0.80
Surprise [822, 810]	(0.90) [0.86]	(0.85) [0.82]	(0.82) [0.80]	(0.85) [0.82]

Table 3: Inter-Annotator Agreement for Topic Annotation

6 Experiments on Emotion Classification

A preliminary experiment (Das and Bandyopadhyay, 2009b) was carried out on a small set of 1200 sentences of the annotated blog corpus using Conditional Random Field (CRF) (McCallum *et al.*, 2001). We have employed the same corpus and similar features (e.g. POS, punctuation symbols, sentiment words etc.) for classifying the emotion words using Support Vector Machine (SVM) (Joachims, 1999). The results on 200 test sentences are shown in Table 4. The results of the automatic emotion classification at word level show that SVM outperforms CRF significantly. It is observed that both classifiers fail to identify the emotion words that are enriched by morphological inflections. Although SVM outperforms CRF but both CRF and SVM suffer from sequence labeling and label bias problem with other non-emotional words of a sentence. (For error analysis and detail experiments, see Das and Bandyopadhyay, 2009b).

Emotion	Test Set	
Classes (# Words)	CRF	SVM
Happy (106)	67.67	80.55
Sad (143)	63.12	78.34
Anger (70)	51.00	66.15
Disgust (65)	49.75	53.35
Fear (37)	52.46	64.78
Surprise (204)	68.23	79.37

Table 4: Word level Emotion tagging Accuracies (in %) using CRF and SVM

Another experiment (Das and Bandyopadhyay, 2009a) was carried out on a small set of 1300 sentences of the annotated blog corpus. They assign any of the Ekman's (1993) six basic emotion tags to the Bengali blog sentences. Conditional Random Field (CRF) based word level emotion classifier classifies the emotion words not only in emotion or non-emotion classes but also the emotion words into Ekman's six emotion classes. Corpus based and sense based *tag weights* that are calculated for each of the six emotion tags are used to identify sentence level emotion tag. Sentence level accuracies for each emotion class were also satisfactory.

Knowledge resources can be leveraged in identifying emotion-related words in text and the lexical coverage of these resources may be limited, given the informal nature of online discourse (Aman and Szpakowicz, 2007). The identification of direct emotion words incorporates the lexicon lookup approach. A recently developed Bengali *WordNet Affect* lists (Das and Bandyopadhyay, 2010) have been used in determining the directly stated emotion words. But, the affect lists covers only 52.79% of the directly stated emotion words.

The fact leads not only to the problem of morphological enrichment but also to refer the problem of identifying emoticons, proverbs, idioms and colloquial or foreign words. But, in our experiments, the case of typographical errors and orthographic features (for e.g. ইসসসস '*disgusting*', বাব্বা '*surprising*') that express or emphasize emotion in text are not considered.

7 Conclusion

The present task addresses the issues of identifying emotional expressions in Bengali blog texts along with the annotation of sentences with emotional components such as intensity, holders and topics. Nested sources are considered for annotating the emotion holder information. The major contribution in the task is the identification and fixing the text spans denoted for emotional expressions and multiple topics in a sentence. Although the preliminary experiments carried out on the small sets of the corpus show satisfactory performance, but the future task is to adopt a corpus-driven approach for building a lexicon of emotion words and phrases and extend the emotion analysis tasks in Bengali.

References

Aman Saima and Stan Szpakowicz. 2007. Identifying Expressions of Emotion in Text. *V. Matoušek and P. Mautner (Eds.): TSD 2007*, *LNAI*, vol. 4629, pp.196-205.

Azar, M. 1999. Argumentative Text as Rhetorical Structure: An Application of Rhetorical Structure Theory. *Argumentation,* vol 13, pp. 97–114.

Bethard Steven, Yu H., Thornton A., Hatzivassiloglou V., and Jurafsky, D. 2004. Automatic Extraction of Opinion Propositions and their Holders. I*n AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.*

Carletta Jean. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, vol. 22(2), pp.249-254.

Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. 2005. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. *Human Language Technology / Empirical Method in Natural Language Processing.*

Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. *Human Language Technology - Empirical Method in Natural Language Processing*, pp. 579-586.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, vol. 20, pp. 37–46.

Das Dipankar and Sivaji Bandyopadhyay. 2009a. Word to Sentence Level Emotion Tagging for Bengali Blogs. *Association for Computational Linguistics –International Joint Conference of Natural Language Processing-2009*, pp. 149-152. Suntec, Singapore.

Das Dipankar and Sivaji Bandyopadhyay. 2009b. Emotion Tagging – A Comparative Study on Bengali and English Blogs. *7th International Conference On Natural Language Processing-09,* pp.177-184, India.

Das Dipankar and Sivaji Bandyopadhyay. 2010. Developing Bengali WordNet Affect for Analyzing Emotion. *International Conference on the Computer Processing of Oriental Languages-International Conference on Software Engineering and Knowledge Engineering-2010*, USA.

Ekman, P. 1992. An Argument for Basic Emotions. *Cognition and Emotion*. vol. 6, pp.169–200.

Joachims, Thorsten. 1998. Text Categorization with Support Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML),*137-142

Lin K. H.-Y., C. Yang and H.-H. Chen. 2007. What Emotions News Articles Trigger in Their Readers?. Proceedings of SIGIR, pp. 733-734.

Mann, W. C. and S. A. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *TEXT* 8, pp. 243–281.

McCallum Andrew, Fernando Pereira and John Lafferty. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and labeling Sequence Data. *ISBN*, 282 – 289.

Mihalcea Rada and Hugo Liu. 2006. A corpus-based approach to finding happiness. *Association for the Advancement of Artificial Intelligence*, pp. 139-144.

Mishne Gilad. 2005. Emotions from text: Machine learning for text-based emotion prediction. *SIGIR'05*, pp. 15-19.

Neviarouskaya Alena, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual Affect Sensing for Social and Expressive Online Communication. *2nd international conference on Affective Computing and Intelligent Interaction*, pp. 218-229.

Passonneau, R. 2004. Computing reliability for coreference annotation. *Language Resources and Evaluation*, Lisbon.

Passonneau, R.J. 2006. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. *Language Resources and Evaluation.*

Quan Changqin and Fuji Ren. 2009. Construction of a Blog Emotion Corpus for Chinese Emotional Expression Analysis. *Empirical Method in Natural Language Processing- Association for Computational Linguistics*, pp. 1446-1454, Singapore

Quirk, R., Greenbaum, S., Leech, G., Svartvik, J. 1985. A Comprehensive Grammar of the English Language. Longman, New York.

Stoyanov, V., and C. Cardie. 2008. Annotating topics of opinions. *Language Resources and Evaluation.*

Tokuhisa Ryoko, Kentaro Inui, and Yuji. Matsumoto. 2008. Emotion Classification Using Massive Examples Extracted from the Web. *COLING 2008*, pp. 881-888.

Wiebe Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, vol. 39, pp.164–210.

Yang C., K. H.-Y. Lin, and H.-H. Chen. 2007. Building Emotion Lexicon from Weblog Corpora. *Association for Computational Linguistics*, pp. 133-136.

SentiWordNet for Indian Languages

Amitava Das[1] and Sivaji Bandyopadhyay[2]
Department of Computer Science and Engineering
Jadavpur University
amitava.santu@gmail.com[1] sivaji_cse_ju@yahoo.com[2]

Abstract

The discipline where sentiment/ opinion/ emotion has been identified and classified in human written text is well known as sentiment analysis. A typical computational approach to sentiment analysis starts with prior polarity lexicons where entries are tagged with their prior out of context polarity as human beings perceive using their cognitive knowledge. Till date, all research efforts found in sentiment lexicon literature deal mostly with English texts. In this article, we propose multiple computational techniques like, WordNet based, dictionary based, corpus based or generative approaches for generating SentiWordNet(s) for Indian languages. Currently, SentiWordNet(s) are being developed for three Indian languages: Bengali, Hindi and Telugu. An online intuitive game has been developed to create and validate the developed SentiWordNet(s) by involving Internet population. A number of automatic, semi-automatic and manual validations and evaluation methodologies have been adopted to measure the coverage and credibility of the developed SentiWordNet(s).

1 Introduction

Sentiment analysis and classification from electronic text is a hard semantic disambiguation problem. The regulating aspects of semantic orientation of a text are natural language context information (Pang et al., 2002) language properties (Wiebe and Mihalcea, 2006), domain pragmatic knowledge (Aue and Gamon, 2005) and lastly most challenging is the time dimension (Read, 2005).

The following example shows that the polarity tag associated with a sentiment word depends on the time dimension. During 90's mobile phone users generally reported in various online reviews about their color phones but in recent times color phone is not just enough. People are fascinated and influenced by touch screen and various software(s) installation facilities on these new generation gadgets.

In typical computational approaches (Higashinaka et al., 2007; Hatzivassiloglou et al., 2000) to sentiment analysis researchers consider the problem of learning a dictionary that maps semantic representations to verbalizations, where the data comes from opinionated electronic text. Although lexicons in these dictionaries are not explicitly marked up with respect to their contextual semantics, they contain only explicit polarity rating and aspect indicators. Lexicon-based approaches can be broadly classified into two categories firstly where the discriminative polarity tag of lexicons is determined on labeled training data and secondly where the lexicons are manually compiled, the later constitutes the main effective approach.

It is undoubted that the manual compilation is always the best way to create monolingual semantic lexicons, but manual methods are expensive in terms of human resources, it involves a substantial number of human annotators and it takes lot of time as well. In this paper we propose several computational techniques to generate sentiment lexicons in Indian languages automatically and semi-automatically. In the present task, SentiWord-

Proceedings of the 8th Workshop on Asian Language Resources, pages 56–63,
Beijing, China, 21-22 August 2010.

Net(s) are being developed for the Bengali, Hindi and Telugu languages.

Several prior polarity sentiment lexicons are available for English such as SentiWordNet (Esuli et. al., 2006), Subjectivity Word List (Wilson et. al., 2005), WordNet Affect list (Strapparava et al., 2004), Taboada's adjective list (Taboada et al., 2006).

Among these publicly available sentiment lexicon resources we find that SentiWordNet is most widely used (number of citation is higher than other resources[1]) in several applications such as sentiment analysis, opinion mining and emotion analysis. Subjectivity Word List is most trustable as the opinion mining system OpinionFinder[2] that uses the subjectivity word list has reported highest score for opinion/sentiment subjectivity (Wiebe and Riloff, 2006). SentiWordNet is an automatically constructed lexical resource for English that assigns a positivity score and a negativity score to each WordNet synset.

The subjectivity word list is compiled from manually developed resources augmented with entries learned from corpora. The entries in the subjectivity word list have been labeled with part of speech (POS) tags as well as either strong or weak subjective tag depending on the reliability of the subjective nature of the entry.

These two resources have been merged automatically and the merged resource is used for SentiWordNet(s) generation in the present task.

The generated sentiment lexicons or SentiWordNet(s) for several Indian languages mostly contain synsets (approximately 60%) of respective languages. Synset based method is robust for any kind of monolingual lexicon creation and useful to avoid further word sense disambiguation problem in application domain.

Additionally we have developed an online intuitive game to create and validate the developed SentiWordNet(s) by involving Internet population.

The proposed approaches in this paper are easy to adopt for any new language. To measure the coverage and credibility of generated SentiWordNet(s) in Indian languages we have developed several automatic and semi-automatic evaluation methods.

2 Related Works

Various methods have been used in the literature such as WordNet based, dictionary based, corpus based or generative approaches for sentiment lexicon generation in a new target language.

Andreevskaia and Bergler, (2006) present a method for extracting sentiment-bearing adjectives from WordNet using the Sentiment Tag Extraction Program (STEP). They did 58 STEP runs on unique non-intersecting seed lists drawn from manually annotated list of positive and negative adjectives and evaluated the results against other manually annotated lists.

The proposed methods in (Wiebe and Riloff, 2006) automatically generate resources for subjectivity analysis for a new target language from the available resources for English. Two techniques have been proposed for the generation of target language lexicon from English subjectivity lexicon. The first technique uses a bilingual dictionary while the second method is a parallel corpus based approach using existing subjectivity analysis tools for English.

Automatically or manually created lexicons may have limited coverage and do not include most semantically contrasting word pairs like antonyms. Antonyms are broadly categorized (Saif Mohammed, 2008) as *gradable adjectives* (hot–cold, good–bad, friend–enemy) and *productive adjectives* (normal–abnormal, fortune–misfortune, implicit–explicit). The first type contains the semantically contrasting word pairs but the second type includes orthographic suffix/affix as a clue. The second type is highly productive using very less number of affixation rules.

Degree of antonymy (Mohammad et al., 2008) is defined to encompass the complete semantic range as a combined measure of the contrast in meaning conveyed by two antonymy words and is identified by distributional hypothesis. It helps to measure relative sentiment score of a word and its antonym.

Kumaran et al., (2008) introduced a beautiful method for automatic data creation by online intuitive games. A methodology has been

[1] http://citeseerx.ist.psu.edu/

[2] http://www.cs.pitt.edu/mpqa/

proposed for community creation of linguistic data by community collaborative framework known as wikiBABEL[3]. It may be described as a revolutionary approach to automatically create large credible linguistic data by involving Internet population for content creation.

For the present task we prefer to involve all the available methodologies to automatically and semi-automatically create and validate SentiWordNet(s) for three Indian languages. Automatic methods involve only computational methods. Semi-automatic methods involve human interruption to validate system's output.

3 Source Lexicon Acquisition

SentiWordNet and Subjectivity Word List have been identified as the most reliable source lexicons. The first one is widely used and the second one is robust in terms of performance. A merged sentiment lexicon has been developed from both the resources by removing the duplicates. It has been observed that 64% of the single word entries are common in the Subjectivity Word List and SentiWordNet. The new merged sentiment lexicon consists of 14,135 numbers of tokens. Several filtering techniques have been applied to generate the new list.

A subset of 8,427 sentiment words has been extracted from the English SentiWordNet, by selecting those whose orientation strength is above the heuristically identified threshold value of 0.4. The words whose orientation strength is below 0.4 are ambiguous and may lose their subjectivity in the target language after translation. A total of weakly subjective 2652 words are discarded from the Subjectivity word list as proposed in (Wiebe and Riloff, 2006).

In the next stage the words whose POS category in the Subjectivity word list is undefined and tagged as "*anypos*" are considered. These words may generate sense ambiguity issues in the next stages of subjectivity detection. The words are checked in the SentiWordNet list for validation. If a match is found with certain POS category, the word is added to the new merged sentiment lexicon. Otherwise the word is discarded to avoid ambiguities later.

Some words in the Subjectivity word list are inflected e.g., *memories*. These words would be stemmed during the translation process, but some words present no subjectivity property after stemming (*memory* has no subjectivity property). A word may occur in the subjectivity list in many inflected forms. Individual clusters for the words sharing the same root form are created and then checked in the SentiWordNet for validation. If the root word exists in the SentiWordNet then it is assumed that the word remains subjective after stemming and hence is added to the new list. Otherwise the cluster is completely discarded to avoid any further ambiguities.

Various statistics of the English SentiWordNet and Subjectivity Word List are reported in Table 1.

	SentiWordNet		**Subjectivity Word List**	
Entries	**Single**	**Multi**	**Single**	**Multi**
	115424	79091	5866	990
Uambiguous Words	20789	30000	4745	963
Discarded Ambiguous Words	**Threshold**	**Orientation Strength**	**Subjectivity Strength**	**POS**
	86944	30000	2652	928

Table 1: English SentiWordNet and Subjectivity Word List Statistics

4 Target Lexicon Generation

4.1 Bilingual Dictionary Based Approach

A word-level translation process followed by error reduction technique has been adopted for generating the Indian languages SentiWordNet(s) from the English sentiment lexicon merged from the English SentiWordNet and the Subjectivity Word List.

English to Indian languages synsets are being developed under Project English to Indian Languages Machine Translation Systems

[3] http://research.microsoft.com/en-us/projects/wikibabel/

(EILMT)[4], a consortia project funded by Department of Information Technology (DIT), Government of India. These synsets are robust and reliable as these are created by native speakers as well as linguistics experts of the specific languages. For each language we have approximately 9966 synsets along with the English WordNet offset. These bilingual synset dictionaries have been used along with language specific dictionaries.

A word level synset/lexical transfer technique is applied to each English synset/word in the merged sentiment lexicon. Each dictionary search produces a set of Indian languages synsets/words for a particular English synset/word.

4.1.1 Hindi

Two available manually compiled English-Hindi electronic dictionaries have been identified for the present task. First is the SHABDKOSH[5] and the second one is Shabdanjali[6]. These two dictionaries have been merged automatically by replacing the duplicates. The merged English-Hindi dictionary contains approximately 90,872 unique entries. The positive and negative sentiment scores for the Hindi words are copied from their English SentiWordNet.

The bilingual dictionary based translation process has resulted 22,708 Hindi entries.

4.1.2 Bengali

An English-Bengali dictionary (approximately 102119 entries) has been developed using the Samsad Bengali-English dictionary[7]. The positive and negative sentiment scores for the Bengali words are copied from their English SentiWordNet equivalents.

The bilingual dictionary based translation process has resulted in 35,805 Bengali entries. A manual checking is done to identify the reliability of the words generated from automatic process. After manual checking only 1688 words are discarded i.e., the final list consists of 34,117 words.

4.2 Telugu

Charles Philip Brown English-Telugu Dictionary[8], Aksharamala[9] English-Telugu Dictionary and English-Telugu Dictionary[10] developed by Language Technology Research Center (LTRC), International Institute of Hyderabad (IITH) have been chosen for the present task. There is no WordNet publicly available for Telugu and the corpus (Section 4.5) we used is small in size. Dictionary based approach is the main process for Telugu SentiWordNet generation.

These three dictionaries have been merged automatically by replacing the duplicates. The merged English-Telugu dictionary contains approximately 112310 unique entries. The positive and negative sentiment scores for the Telugu words are copied from their English SentiWordNet equivalents.

The dictionary based translation process has resulted in 30,889 Telugu entries, about 88% of final Telugu SentiWordNet synsets. An online intuitive game has been proposed in Section 4.6 to automatically validate the developed Telugu SentiWordNet by involving Internet population.

4.3 WordNet Based Approach

WordNet(s) are available for Hindi[11] (Jha et al., 2001) and Bengali[12] (Robkop et al., 2010) but publicly unavailable for Telugu.

A WordNet based lexicon expansion strategy has been adopted to increase the coverage of the generated SentiWordNet(s) through the dictionary based approach. The present algorithm starts with English SentiWordNet synsets that is expanded using synonymy and antonymy relations in the WordNet. For matching synsets we keep the exact score as in the source synset in the English SentiWordNet. The calculated positivity and negativity score

[4] http://www.cdacmumbai.in/e-ilmt

[5] http://www.shabdkosh.com/

[6] http://www.shabdkosh.com/content/category/downloads/

[7] http://dsal.uchicago.edu/dictionaries/biswas_bengali/

[8] http://dsal.uchicago.edu/dictionaries/brown/

[9] https://groups.google.com/group/aksharamala

[10] http://ltrc.iiit.ac.in/onlineServices/Dictionaries/Dict_Frame.html

[11] http://www.cfilt.iitb.ac.in/wordnet/webhwn/

[12] http://bn.asianwordnet.org/

for any target language antonym synset is calculated as:

$$T_p = 1 - S_p$$
$$T_n = 1 - S_n$$

where S_p, S_n are the positivity and negativity score for the source language (i.e, English) and T_p, T_n are the positivity and negativity score for target languages (i.e., Hindi and Bengali) respectively.

4.3.1 Hindi

Hindi WordNet is a well structured and manually compiled resource and is being updated since last nine years. There is an available API[13] for accessing the Hindi WordNet. Almost 60% of final SentiWordNet synsets in Hindi are generated by this method.

4.3.2 Bengali

The Bengali WordNet is being developed by the Asian WordNet (AWN) community. It only contains 1775 noun synsets as reported in (Robkop et al., 2010). A Web Service[14] has been provided for accessing the Bengali WordNet. There are only a few number of noun synsets in the Bengali WordNet and other important POS category words for sentiment lexicon such as adjective, adverb and verb are absent. Only 5% new lexicon entries have been generated in this process.

4.4 Antonym Generation

Automatically or manually created lexicons have limited coverage and do not include most semantically contrasting word pairs. To overcome the limitation and increase the coverage of the SentiWordNet(s) we present automatic antonymy generation technique followed by corpus validation to check orthographically generated antonym does really exist. Only 16 hand crafted rules have been used as reported in Table 2. About 8% of Bengali, 7% of Hindi and 11% of Telugu SentiWordNet entries are generated in this process.

Affix/Suffix	Word	Antonym
***ab*X**	Normal	*Ab*-normal
***mis*X**	Fortune	*Mis*-fortune
***im*X-exX**	*Im*-plicit	*Ex*-plicit
***anti*X**	Clockwise	*Anti*-clockwise
***non*X**	Aligned	*Non*-aligned
***in*X-*ex*X**	*In*-trovert	*Ex*-trovert
***dis*X**	Interest	*Dis*-interest
***un*X**	Biased	*Un*-biased
***up*X-*down*X**	*Up*-hill	*Down*-hill
***im*X**	Possible	*Im*-possible
***ill*X**	Legal	*Il*-legal
***over*X-*under*X**	Overdone	*Under*-done
***in*X**	Consistent	*In*-consistent
***r*X-*ir*X**	Regular	*Ir*-regular
X*less*-X*ful*	Harm-*less*	Harm-*ful*
***mal*X**	Function	*Mal*-function

Table 2: Rules for Generating Productive Antonyms

4.5 Corpus Based Approach

Language/culture specific words such as those listed below are to be captured in the developed SentiWordNet(s). But sentiment lexicon generation techniques via cross-lingual projection are unable to capture these words. As example:

सहेरा (Sahera: A marriage-wear)

দুর্গাপুজো (Durgapujo: A festival of Bengal)

To increase the coverage of the developed SentiWordNet(s) and to capture the language/culture specific words an automatic corpus based approach has been proposed. At this stage the developed SentiWordNet(s) for the three Indian languages have been used as a seed list. Language specific corpus is automatically tagged with these seed words and we have a simple tagset as SWP (Sentiment Word Positive) and SWN (Sentiment Word Negative). Although we have both positivity and negativity scores for the words in the seed list but we prefer a word level tag as either positive or negative following the highest sentiment score.

A Conditional Random Field (CRF[15]) based Machine Learning model is then trained with the seed list corpus along with multiple linguistics features such as morpheme, parts-of-

[13] http://www.cfilt.iitb.ac.in/wordnet/webhwn/API_downloaderInfo.php

[14] http://bn.asianwordnet.org/services

[15] http://crfpp.sourceforge.net

speech, and chunk label. These linguistics features have been extracted by the shallow parsers[16] for Indian languages. An n-gram (*n*=4) sequence labeling model has been used for the present task.

The monolingual corpuses used have been developed under Project English to Indian Languages Machine Translation Systems (EILMT). Each corpus has approximately 10K of sentences.

4.6 Gaming Methodology

There are several motivations behind developing an intuitive game to automatically create multilingual SentiWordNet(s). The assigned polarity scores to each synset may vary in time dimension. Language specific polarity scores may vary and it should be authenticated by numbers of language specific annotators.

In the history of Information Retrieval research there is a milestone when ESP[17] game (Ahn et al., 2004) innovate the concept of a game to automatically label images available in World Wide Web. Highly motivated by the historical research we proposed a intuitive game to create and validate SentiWordNet(s) for Indian languages by involving internet population.

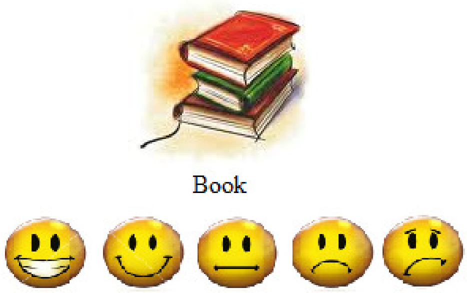


Figure 1: Intuitive Game for SentiWordNet(s) Creation

In the gaming interface a simple picture (retrieved by Google Image API[18]) along with a sentiment bearing word (retrieved randomly from SentiWordNet) is displayed to a player and he/she is then been asked to capture his immediate sentiment as extreme positive, positive, extreme negative, negative or neutral by pressing appropriate emoticon buttons. A snap of the game is shown in the Figure 1. The sentiment score is calculated by the different emoticons based on the inputs from the different players and then is assigned the scale as follows: extreme positive (pos: 0.5, neg: 0.0), positive (pos: 0.25, neg: 0.0), neutral (pos: 0.0, neg: 0.0), negative (pos: 0.0, 0.25), extreme negative (pos: 0.0, neg: 0.5).

The score of a particular player is calculated on the basis of pre-stored sentiment lexicon scores in the generated SentiWordNet(s).

5 Evaluation

Andera Esuli and Fabrizio Sebastiani (2006) have calculated the reliability of the sentiment scores attached to every synsets in the English SentiWordNet. They have tagged sentiment words in the English WordNet with positive and negative sentiment scores. In the present task, these sentiment scores from English WordNet have been directly copied to the Indian language SentiWordNet(s).

Two extrinsic evaluation strategies have been adopted for the developed Bengali SentiWordNet based on the two main usages of the sentiment lexicon as subjectivity classifier and polarity identifier. The Hindi and Telugu SentiWordNet(s) have not completely been evaluated.

5.1 Coverage

	NEWS	BLOG
Total number of documents	100	-
Total number of sentences	2234	300
Avgerage number of sentences in a document	22	-
Total number of wordforms	28807	4675
Avgerage number of wordforms in a document	288	-
Total number of distinct wordforms	17176	1235

Table 3: Bengali Corpus Statistics

We experimented with NEWS and BLOG corpora for subjectivity detection. Sentiment lexicons are generally domain independent but it provides a good baseline while working with sentiment analysis systems. The coverage of

[16] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

[17] http://www.espgame.org/

[18] http://code.google.com/apis/ajaxsearch/multimedia.html

the developed Bengali SentiWordNet is evaluated by using it in a subjectivity classifier (Das and Bandyopadhyay, 2009). The statistics of the NEWS and BLOG corpora is reported in Table 3.

For comparison with the coverage of English SentiWordNet the same subjectivity classifier (Das and Bandyopadhyay, 2009) has been applied on Multi Perspective Question Answering (MPQA) (NEWS) and IMDB Movie review corpus along with English SentiWordNet. The result of the subjectivity classifier on both the corpus proves that the coverage of the Bengali SentiWordNet is reasonably good. The subjectivity word list used in the subjectivity classifier is developed from the IMDB corpus and hence the experiments on the IMDB corpus have yielded high precision and recall scores. The developed Bengali SentiWordNet is domain independent and still its coverage is very good as shown in Table 4.

Languages	Domain	Precision	Recall
English	**MPQA**	76.08%	83.33%
	IMDB	79.90%	86.55%
Bengali	**NEWS**	72.16%	76.00%
	BLOG	74.6%	80.4%

Table 4: Subjectivity Classifier using SentiWordNet

5.2 Polarity Scores

This evaluation metric measures the reliability of the associated polarity scores in the sentiment lexicons. To measure the reliability of polarity scores in the developed Bengali SentiWordNet, a polarity classifier (Das and Bandyopadhyay, 2010) has been developed using the Bengali SentiWordNet along with some other linguistic features.

Features	Overall Performance Incremented By
SentiWordNet	**47.60%**

Table 5: Polarity Performance Using Bengali SentiWordNet

Feature ablation method proves that the associated polarity scores in the developed Bengali SentiWordNet are reliable. Table 5 shows the performance of a polarity classifier using the Bengali SentiWordNet. The polarity wise overall performance of the polarity classifier is reported in Table 6.

Polarity	Precision	Recall
Positive	56.59%	52.89%
Negative	75.57%	65.87%

Table 6: Polarity-wise Performance Using Bengali SentiWordNet

Comparative study with a polarity classifier that works with only prior polarity lexicon is necessary but no such works have been identified in literature.

An arbitrary 100 words have been chosen from the Hindi SentiWordNet for human evaluation. Two persons are asked to manually check it and the result is reported in Table 7. The coverage of the Hindi SentiWordNet has not been evaluated, as no manually annotated sentiment corpus is available.

Polarity	Positive	Negative
Percentage	88.0%	91.0%

Table 7: Evaluation of Polarity Score of Developed Hindi SentiWordNet

For Telugu we created a version of the game with Telugu words on screen. Only 3 users have played the Telugu language specific game till date. Total 92 arbitrary words have been tagged and the accuracy of the polarity scores is reported in Table 8. The coverage of Telugu SentiWordNet has not been evaluated, as no manually annotated sentiment corpus is available.

Polarity	Positive	Negative
Percentage	82.0%	78.0%

Table 8: Evaluation of Polarity Score of Developed Telugu SentiWordNet

6 Conclusion

SentiWordNet(s) for Indian languages are being developed using various approaches. The game based technique may be directed towards a new way for the creation of linguistic data not just only for SentiWordNet(s) but in either areas of NLP too.

Presently only the Bengali SentiWordNet[19] is downloadable from the author's web page.

[19] http://www.amitavadas.com/sentiwordnet.php

References

Andreevskaia Alina and Bergler Sabine. CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging. In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 117–120, Prague, June 2007.

Aue A. and Gamon M., Customizing sentiment classifiers to new domains: A case study. In Proceedings of Recent Advances in Natural Language Processing (RANLP), 2005.

Das A. and Bandyopadhyay S. (2010). Phrase-level Polarity Identification for Bengali, In International Journal of Computational Linguistics and Applications (IJCLA), Vol. 1, No. 1-2, Jan-Dec 2010, ISSN 0976-0962, Pages 169-182.

Das A. and Bandyopadhyay S. Subjectivity Detection in English and Bengali: A CRF-based Approach. In the Proceeding of ICON 2009.

Esuli Andrea and Sebastiani Fabrizio. SentiWordNet: A publicly available lexical resource for opinion mining. In Proceedings of Language Resources and Evaluation (LREC), 2006.

Hatzivassiloglou, Vasileios and Wiebe Janyce. Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings of COLING-00, 18th International Conference on Computational Linguistics. Saarbru¨cken, GE. Pages 299-305. 2000.

Higashinaka Ryuichiro, Walker Marilyn, and Prasad Rashmi. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. ACM Transactions on Speech and Language Processing (TSLP), 2007.

Jha S., Narayan D., Pande P. and Bhattacharyya P. A WordNet for Hindi, International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India, January 2001.

Kumaran A., Saravanan K. and Maurice Sandor. WikiBABEL: Community Creation of Multilingual Data, in the WikiSYM 2008 Conference, Porto, Portugal, Association for Computing Machinery, Inc., September 2008.

Mihalcea Rada, Banea Carmen and Wiebe Janyce. Learning multilingual subjective language via-cross-lingual projections. In Proceedings of the Association for Computational Linguistics (ACL), pages 976–983, Prague, Czech Republic, June 2007.

Mohammad Saif, Dorr Bonnie, and Hirst Graeme. Computing Word-Pair Antonymy. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-2008), October 2008, Waikiki, Hawaii.

Pang Bo, Lee Lillian, and Vaithyanathan Shivakumar. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86, 2002.

Read Jonathon. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL Student Research Workshop, 2005.

Robkop Kergrit, Thoongsup Sareewan, Charoenporn Thatsanee, Sornlertlamvanich Virach and Isahara Hitoshi.WNMS: Connecting the Distributed WordNet in the Case of Asian WordNet. . In the Proceeding of 5th International Conference of the Global WordNet Association (GWC-2010), Mumbai, India , 31st Jan. - 4th Feb., 2010.

Wiebe Janyce and Mihalcea Rada. Word sense and subjectivity. In Proceedings of COLING/ACL-06 the 21st Conference on Computational Linguistics/Association for Computational Linguistics. Sydney, Australia. Pages 1065--1072.

Wiebe Janyce and Riloff Ellen. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In Proceeding of International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Pages 475–486, 2006.

Wilson Theresa, Wiebe Janyce and Hoffmann Paul (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In Proceedings of HLT/EMNLP 2005, Vancouver, Canada.

Constructing Thai Opinion Mining Resource: A Case Study on Hotel Reviews

Choochart Haruechaiyasak, Alisa Kongthon,
Pornpimon Palingoon and **Chatchawal Sangkeettrakarn**
Human Language Technology Laboratory (HLT)
National Electronics and Computer Technology Center (NECTEC)
choochart.har@nectec.or.th, alisa.kon@nectec.or.th,
pornpimon.pal@nectec.or.th, chatchawal.san@nectec.or.th

Abstract

Opinion mining and sentiment analysis has recently gained increasing attention among the NLP community. Opinion mining is considered a domain-dependent task. Constructing lexicons for different domains is labor intensive. In this paper, we propose a framework for constructing Thai language resource for *feature-based* opinion mining. The feature-based opinion mining essentially relies on the use of two main lexicons, *features* and *polar words*. Our approach for extracting features and polar words from opinionated texts is based on syntactic pattern analysis. The evaluation is performed with a case study on hotel reviews. The proposed method has shown to be very effective in most cases. However, in some cases, the extraction is not quite straightforward. The reasons are due to, firstly, the use of conversational language in written opinionated texts and, secondly, the language semantic. We provide discussion with possible solutions on pattern extraction for some of the challenging cases.

1 Introduction

With the popularity of Web 2.0 or social networking websites, the amount of user-generated contents has increased exponentially. One interesting type of these user-generated contents is texts which are written with some opinions and/or sentiments. An in-depth analysis of these opinionated texts could reveal potentially useful information regarding the preferences of people towards many different topics including news events, social issues and commercial products. Opinion mining and sentiment analysis is such task for analyzing and summarizing what people think about a certain topic.

Due to its potential and useful applications, opinion mining has gained a lot of interest in text mining and NLP communities (Ding et al., 2008; Jin et al., 2009). Much work in this area focused on evaluating reviews as being positive or negative either at the document level (Turney, 2002; Pang et al., 2002; Dave et al., 2003; Beineke et al., 2004) or sentence level (Kim and Hovy, 2004; Wiebe and Riloff, 2005; Wilson et al., 2009; Yu and Hatzivassiloglou, 2003). For instance, given some reviews of a product, the system classifies them into positive or negative reviews. No specific details or features are identified about what customers like or dislike. To obtain such details, a *feature-based* opinion mining approach has been proposed (Hu and Liu, 2004; Popescu and Etzioni, 2005). This approach typically consists of two following steps.

1. Identifying and extracting features of an object, topic or event from each sentence upon which the reviewers expressed their opinion.

2. Determining whether the opinions regarding the features are positive or negative.

The feature-based opinion mining could provide users with some insightful information related to opinions on a particular topic. For example, for hotel reviews, the feature-based opinion

Proceedings of the 8th Workshop on Asian Language Resources, pages 64–71,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

mining allows users to view positive or negative opinions on hotel-related features such as price, service, breakfast, room, facilities and activities. Breaking down opinions into feature level is very essential for decision making. Different customers could have different preferences when selecting hotels to stay for vacation. For example, some might prefer hotels which provide full facilities, however, some might prefer to have good room service.

The main drawback of the feature-based opinion mining is the preparation of different lexicons including *features* and *polar words*. To make things worse, these lexicons, especially the features, are domain-dependent. For a particular domain, a set of features and polar words must be prepared. The process for language resource construction is generally labor intensive and time consuming. Some previous works have proposed different approaches for automatically constructing the lexicons for the feature-based opinion mining (Qiu et al., 2009; Riloff and Wiebe, 2003; Sarmento et al., 2009). Most approaches applied some machine learning algorithms for learning the rules from the corpus. The rules are used for extracting new features and polar words from untagged corpus. Reviews of different approaches are given in the related work section.

In this paper, we propose a framework for constructing Thai language resource for the feature-based opinion mining. Our approach is based on syntactic pattern analysis of two lexicon types: *domain-dependent* and *domain-independent*. The domain-dependent lexicons include features, sub-features and polar words. The domain-independent lexicons are particles, negative words, degree words, auxiliary verbs, prepositions and stop words. Using these lexicons, we could construct a set of syntactic rules based on the frequently occurred patterns. The rule set can be used for extracting more unseen sub-features and polar words from untagged corpus.

We evaluated the proposed framework on the domain of hotel reviews. The experimental results showed that our proposed method is very effective in most cases, especially for extracting polar words. However, in some cases, the extraction is not quite straightforward due to the use of conversational language, idioms and hidden semantic. We provide some discussion on the challenging cases and suggest some solutions as the future work.

The remainder of this paper is organized as follows. In next section, we review some related works on different approaches for constructing language resources for opinion mining and sentiment analysis. In Section 3, we present the proposed framework for constructing Thai opinion mining resource by using the dual pattern extraction method. In Section 4, we apply the proposed framework with a case study of hotel reviews. The performance evaluation is given with the experiment results. Some difficult cases are discussed along with some possible solutions. Section 5 concludes the paper with the future work.

2 Related work

The problem of developing subjectivity lexicons for training and testing sentiment classifiers has recently attracted some attention. The Multi-perspective Question Answering (MPQA) opinion corpus is a well-known resource for sentiment analysis in English (Wiebe et al., 2005). It is a collection of news articles from a variety of news sources manually annotated at word and phrase levels for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.). The annotation in this work also took into account the context, which is essential for resolving possible ambiguities and accurately determining polarity.

Although most of the reference corpora has been focused on English language, work on other languages is growing as well. Kanayama et al. (2006) proposed an unsupervised method to detect sentiment words in Japanese. In this work, they used clause level context coherency to identify candidate sentiment words from sentences that appear successively with sentences containing seed sentiment words. Their assumption is that unless the context is changed with adversative expressions, sentences appearing together in that context tend to have the same polarities. Hence, if one of them contains sentiments words, the other successive sentences are likely to contain sentiment words as well. Ku and Chen (2007) proposed the bag-of-characters approach to determine sentiment words in Chinese. This approach calculates the observation probabilities of characters from a set of seed sentiment words first, then dynamically expands the set and adjusts their probabilities. Later in 2009, Ku et al. (2009), extended their bag-of-characters approach by including morphological structures and syntactic structures between sentence segment. Their experiments showed better performance of word polarity detection and opinion sentence extraction.

Some other methods to automatically generate resources for subjectivity analysis for a foreign language have leveraged the resources and tools available for English. For example, Benea et al. (2008) applied machine translation and standard Naive Bayes and SVM for subjectivity classification for Romanian. Their experiments showed promising results for applying automatic translation to construct resources and tools for opinion mining in a foreign language. Wan (2009) also leveraged an available English corpus for Chinese sentiment classification by using the co-training approach to make full use of both English and Chinese features in a unified framework. Jijkoun and Hofmann (2009) also described a method for creating a Dutch subjectivity lexicon based on an English lexicon. They applied a PageRank-like algorithm that bootstraps a subjectivity lexicon from the translation of the English lexicon and rank the words in the thesaurus by polarity using the network of lexical relations (e.g., synonymy, hyponymy) in Wordnet.

3 The proposed framework

The performance of the feature-based opinion mining relies on the design and completeness of related lexicons. Our lexicon design distinguishes lexicons into two types, domain-dependent and domain-independent. The design of domain-dependent lexicons is based on the feature-based opinion mining framework proposed by Liu et al. (2005). The framework starts by setting the domain scope such as *digital camera*. The next step is to design a set of features associated with the given domain. For the domain of digital camera, features could be, for instance, "price", "screen size" and "picture quality". Features could contain sub-features. For example, the picture quality could have the sub-features as "macro mode", "portrait mode" and "night mode". Preparing multiple feature levels could be time-consuming, therefore, we limit the features into two levels: *main features* and *sub-features*.

Another domain-dependent lexicon is *polar words*. Polar words are sentiment words which represent either positive or negative views on features. Although some polar words are domain-independent and have explicit meanings such as "excellent", "beautiful", "expensive" and "terrible". Some polar words are domain-dependent and have implicit meanings depending on the contexts. For example, the word "large" is generally considered positive for the *screen size* feature of digital camera domain. However, for the *dimension* feature of mobile phone domain, the word "large" could be considered as negative.

On the other hand, the domain-independent lexicons are regular words which provide different parts of speech (POS) and functions in the sentence. For opinion mining task, we design six different domain-independent lexicons as follows (some examples are shown in Table 1).

- **Particles (PAR)**: In Thai language, these words refer to the sentence endings which are normally used to add politeness of the speakers (Cooke, 1992).

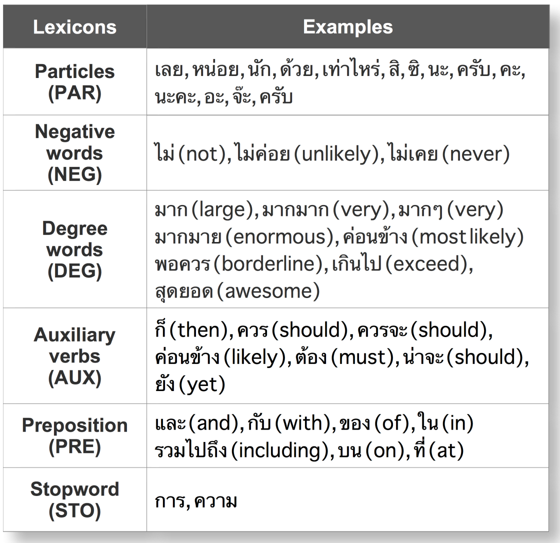
- **Negative words (NEG)**: Like English, these words are used to invert the opinion polarity. Examples are "not", "unlikely" and "never".

- **Degree words (DEG)**: These words are used as an intensifier to the polar words. Examples are "large", "very", "enormous".

- **Auxiliary verbs (AUX)**: These words are used to modify verbs. Examples are "should", "must" and "then".

- **Prepositions (PRE)**: Like English, Thai prepositions are used to mark the relations between two words.

- **Stop words (STO)**: These words are used for grammaticalization. Thai language is considered an isolating language, to form a noun the words "karn" and "kwam" are normally placed in front of a verb or a noun, respectively. Therefore, these words could be neglected when analyzing opinionated texts.



Lexicons	Examples
Particles (PAR)	เลย, หน่อย, นัก, ด้วย, เท่าไหร่, สิ, ซิ, นะ, ครับ, คะ, นะคะ, อะ, จ๊ะ, ครับ
Negative words (NEG)	ไม่ (not), ไม่ค่อย (unlikely), ไม่เคย (never)
Degree words (DEG)	มาก (large), มากมาก (very), มากๆ (very) มากมาย (enormous), ค่อนข้าง (most likely) พอควร (borderline), เกินไป (exceed), สุดยอด (awesome)
Auxiliary verbs (AUX)	ก็ (then), ควร (should), ควรจะ (should), ค่อนข้าง (likely), ต้อง (must), น่าจะ (should), ยัง (yet)
Preposition (PRE)	และ (and), กับ (with), ของ (of), ใน (in) รวมไปถึง (including), บน (on), ที่ (at)
Stopword (STO)	การ, ความ

Table 1: Domain-independent lexicons

Although some of the above lexicons are similar to English, however, some words are placed in different position in a sentence. For example, in Thai, a degree word is usually placed after a polar word. For example, "very good" would be written as "good very" in Thai.

Figure 1 shows all processes and work flow under the proposed framework. The process starts with a corpus which is tagged based on two lexicon types. From the tagged corpus, we construct patterns and lexicons. The pattern construction is performed by collecting text segments which contain both features and polar words. All patterns are sorted by the frequency of occurrence. The lexicon construction is performed by simply collecting words which are already tagged with the lexicon types. The lexicons are used for performing the feature-based opinion mining task such as classifying and summarizing the reviews as positive and negative based on different features. The completeness of lexicons is very important for the feature-based opinion mining. To collect more lexicons, patterns are used in the dual pattern extraction process to extract more features and polar words from the untagged corpus.

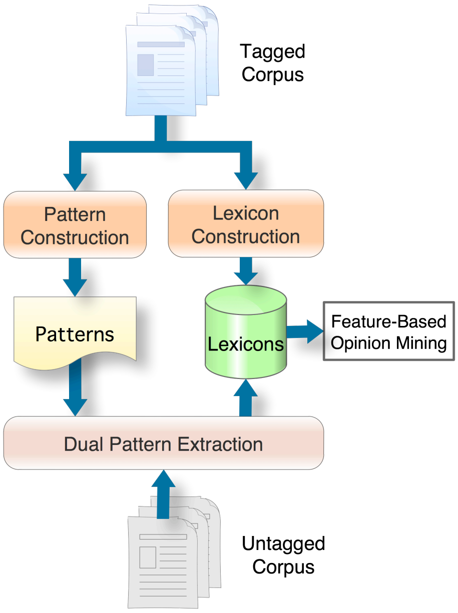


Figure 1: The proposed opinion resource construction framework based on the dual pattern extraction.

4 A case study of hotel reviews

To evaluate the proposed framework, we perform some experiments with a case study of hotel reviews. In Thailand, tourism is ranked as one of the top industries. From the statistics provided by the Office of Tourism Development[1], the number of international tourists visiting Thailand in 2009 is approximately 14 mil-

[1] The Office of Tourism Development, *http://www.tourism.go.th*

lions. The number of registered hotels in all regions of Thailand is approximately 5,000. Providing an opinion mining system on hotel reviews could be very useful for tourists to make decision on hotel choice when planning a trip.

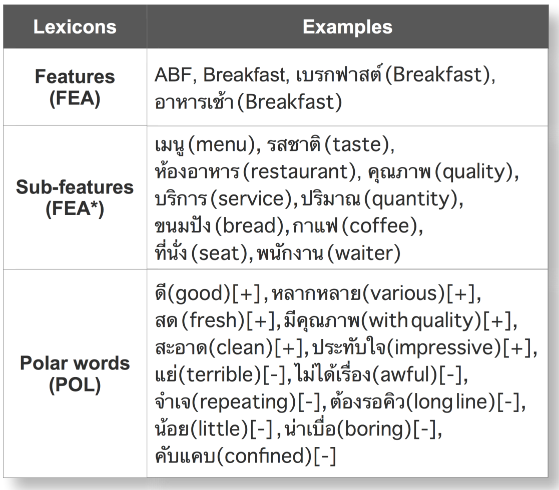
4.1 Corpus preparation

We collected customer reviews from the *Agoda* website[2]. The total number of reviews in the corpus is *8,436*. Each review contains the name of the hotel as the title and comments in free-form text format. We designed a set of 13 main features: service, cleanliness, hotel condition, location, food, breakfast, room, facilities, price, comfort, quality, activities and security. The set of main features is designed based on the features obtained from the Agoda website. Some additional features, such as activities and security, are added to provide users with more dimensions.

In this paper, we focus on two main features: *breakfast* and *service*. Table 2 shows the domain-dependent lexicons related to the breakfast feature. For breakfast main feature (FEA), we include all synonyms which could be used to describe breakfast in Thai. These include English terms with their synonyms, transliterated terms and abbreviations.

The breakfast sub-features (FEA*) are specific concepts of breakfast. Examples include “menu”, “taste”, “service” and “coffee”. It can be observed that some of the sub-features could also act as a main feature. For example, the sub-feature “service” of breakfast is also used as the main feature “service”. Providing sub-feature level could help revealing more insightful dimension for the users. However, designing multiple feature levels could be time-consuming, therefore, we limit the features into two levels, i.e., main feature and sub-feature. The polar words (POL) are also shown in the table. We denote the positive and negative polar words by placing [+] and [-] after each word. It can be observed that some polar words are dependent on sub-features. For example, the polar word “long line” can only be used for the sub-feature “restaurant”.

[2] Agoda website: *http://www.agoda.com*



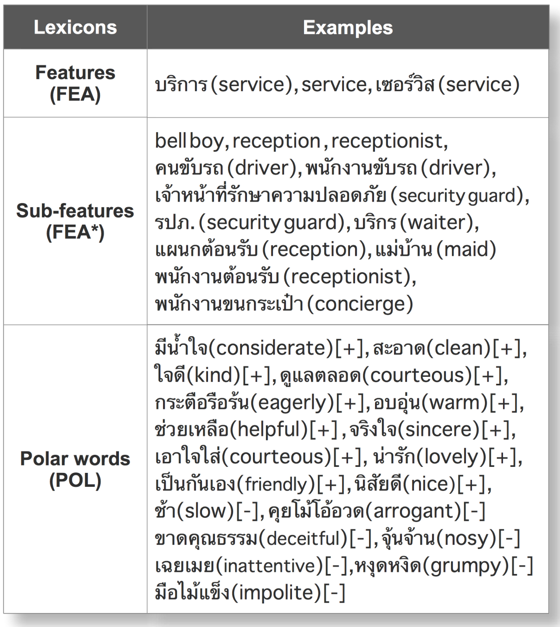
Lexicons	Examples
Features (FEA)	ABF, Breakfast, เบรกฟาสต์ (Breakfast), อาหารเช้า (Breakfast)
Sub-features (FEA*)	เมนู (menu), รสชาติ (taste), ห้องอาหาร (restaurant), คุณภาพ (quality), บริการ (service), ปริมาณ (quantity), ขนมปัง (bread), กาแฟ (coffee), ที่นั่ง (seat), พนักงาน (waiter)
Polar words (POL)	ดี(good)[+], หลากหลาย(various)[+], สด (fresh)[+], มีคุณภาพ(with quality)[+], สะอาด(clean)[+], ประทับใจ(impressive)[+], แย่(terrible)[-], ไม่ได้เรื่อง(awful)[-], จำเจ(repeating)[-], ต้องรอคิว(long line)[-], น้อย(little)[-], น่าเบื่อ(boring)[-], คับแคบ(confined)[-]

Table 2: Domain-dependent lexicons for the *breakfast* feature.

Table 3 shows the domain-dependent lexicons related to the service feature. The main features include synonyms, transliterated and English terms which describe the concept service. The service sub-features are, for example, “reception”, “security guard”, “maid”, “waiter” and “concierge”. Unlike the breakfast feature, the polar words for the service feature are quite general and could mostly be applied for all sub-features. Another observation is that some of the polar words are based on Thai idiom. For example, the phrase “having rigid hands” in Thai means “impolite”. In Thai culture, people show politeness by doing the “wai” gesture.

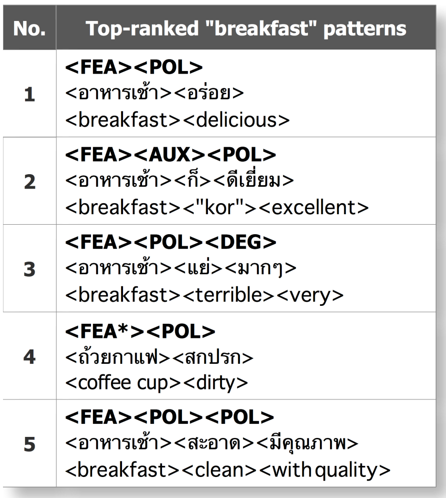
4.2 Experiments and results

Using the tagged corpus and the extracted lexicons, we construct the most frequently occurred patterns. For two main features, breakfast and service, the numbers of tagged reviews for each feature are 301 and 831, respectively. We randomly split the corpus into 80% as training set and 20% as test set. We only consider the patterns which contain both features (either main features or sub-features) and polar words. For the breakfast feature, the total number of extracted patterns is 86. For the service feature, the total number of extracted patterns is 192. Table 4 and 5 show some examples of most frequently



Lexicons	Examples
Features (FEA)	บริการ (service), service, เซอร์วิส (service)
Sub-features (FEA*)	bell boy, reception , receptionist, คนขับรถ (driver), พนักงานขับรถ (driver), เจ้าหน้าที่รักษาความปลอดภัย (security guard), รปภ. (security guard), บริกร (waiter), แผนกต้อนรับ (reception), แม่บ้าน (maid) พนักงานต้อนรับ (receptionist), พนักงานขนกระเป๋า (concierge)
Polar words (POL)	มีน้ำใจ(considerate)[+], สะอาด(clean)[+], ใจดี(kind)[+], ดูแลตลอด(courteous)[+], กระตือรือร้น(eagerly)[+], อบอุ่น(warm)[+], ช่วยเหลือ(helpful)[+] , จริงใจ(sincere)[+], เอาใจใส่(courteous)[+], น่ารัก(lovely)[+], เป็นกันเอง(friendly)[+], นิสัยดี(nice)[+], ช้า(slow)[-], คุยโม้โอ้อวด(arrogant)[-] ขาดคุณธรรม(deceitful)[-], จุ้นจ้าน(nosy)[-] เฉยเมย(inattentive)[-],หงุดหงิด(grumpy)[-] มือไม้แข็ง(impolite)[-]

Table 3: Domain-dependent lexicons for the *service* feature.



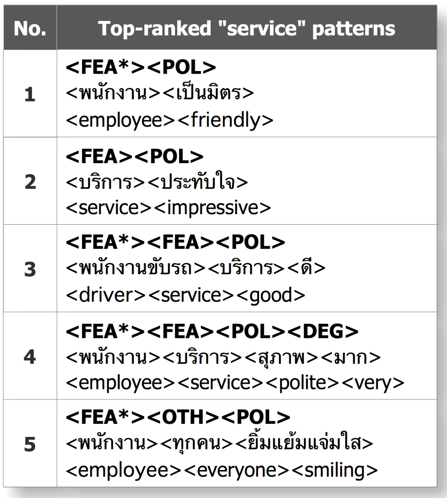
No.	Top-ranked "breakfast" patterns
1	**<FEA><POL>** <อาหารเช้า><อร่อย> <breakfast><delicious>
2	**<FEA><AUX><POL>** <อาหารเช้า><ก็><ดีเยี่ยม> <breakfast><"kor"><excellent>
3	**<FEA><POL><DEG>** <อาหารเช้า><แย่><มากๆ> <breakfast><terrible><very>
4	**<FEA*><POL>** <ถ้วยกาแฟ><สกปรก> <coffee cup><dirty>
5	**<FEA><POL><POL>** <อาหารเช้า><สะอาด><มีคุณภาพ> <breakfast><clean><with quality>

Table 4: Top-ranked *breakfast* patterns with examples

occurred patterns extracted from the corpus. The symbols of the tag set are as shown in Table 1 and 2 with the tag <OTH> denoting any other words.

From the tables, two patterns which occur frequently for both features are <FEA><POL> and <FEA*><POL>. These two patterns are very simple and show that the opinionated texts in Thai are mostly very simple. Users just simply write a word describing the feature followed by a polar word (either positive or negative) without using any verb in between. Some examples for the pattern <FEA*><POL> are <coffee cup><dirty> and <employee><friendly>. In English, a verb "to be" (is/am/are) is usually required between <FEA*> and <POL>.

Using the extracted patterns, we perform the dual pattern extraction process to collect the sub-features and polar words from the test data set. Table 6 shows the evaluation results of sub-features and polar words extraction for both breakfast and service features. It can be observed that the set of patterns could extract polar words (POL) with higher accuracy than sub-features (FEA*). This could be due to the patterns used to describe the polar words are straightforward and not complicated. This is especially true for the case of breakfast feature in which the accuracy is approximately 95%.



No.	Top-ranked "service" patterns
1	**<FEA*><POL>** <พนักงาน><เป็นมิตร> <employee><friendly>
2	**<FEA><POL>** <บริการ><ประทับใจ> <service><impressive>
3	**<FEA*><FEA><POL>** <พนักงานขับรถ><บริการ><ดี> <driver><service><good>
4	**<FEA*><FEA><POL><DEG>** <พนักงาน><บริการ><สุภาพ><มาก> <employee><service><polite><very>
5	**<FEA*><OTH><POL>** <พนักงาน><ทุกคน><ยิ้มแย้มแจ่มใส> <employee><everyone><smiling>

Table 5: Top-ranked *service* patterns with examples

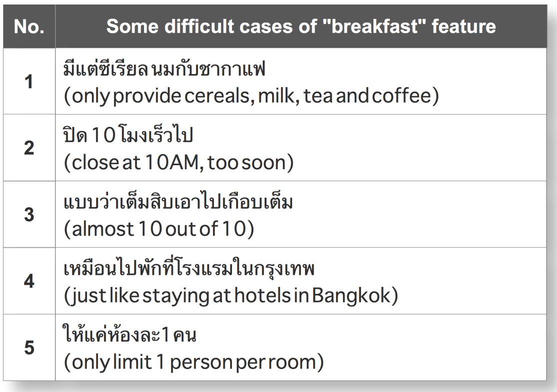
4.3 Discussion

Table 7 and 8 show some examples of challenging cases for breakfast and service features, respectively. The polar words shown in both tables are very difficult to extract since the patterns can

Feature	Accuracy (%)	
	FEA*	POL
Breakfast	80.00	95.74
Service	82.56	89.29

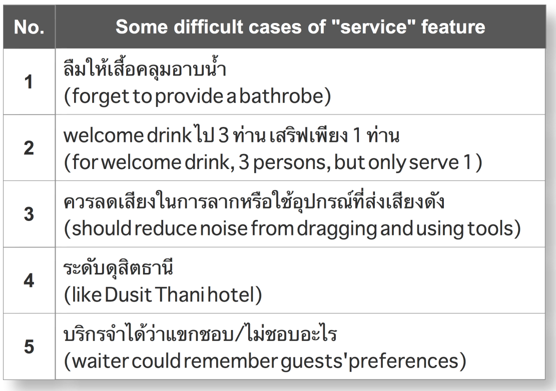
Table 6: Evaluation results of features and polar words extraction.

not be easily captured. The difficulties are due to many reasons including the language semantic and the need of world knowledge. For example, in case #5 of service feature, the whole phrase can be interpreted as "attentive". It is difficult for the system to generate the pattern based on this phrase. Another example is case #4 of both tables, the customers express their opinions by comparing to other hotels. To analyze the sentiment correctly would require the knowledge of a particular hotel or hotels in specific locations.



No.	Some difficult cases of "breakfast" feature
1	มีแต่ซีเรียล นมกับชากาแฟ (only provide cereals, milk, tea and coffee)
2	ปิด 10 โมงเร็วไป (close at 10AM, too soon)
3	แบบว่าเต็มสิบเอาไปเกือบเต็ม (almost 10 out of 10)
4	เหมือนไปพักที่โรงแรมในกรุงเทพ (just like staying at hotels in Bangkok)
5	ให้แค่ห้องละ1 คน (only limit 1 person per room)

Table 7: Examples of difficult cases of *breakfast* feature



No.	Some difficult cases of "service" feature
1	ลืมให้เสื้อคลุมอาบน้ำ (forget to provide a bathrobe)
2	welcome drink ไป 3 ท่าน เสริฟเพียง 1 ท่าน (for welcome drink, 3 persons, but only serve 1)
3	ควรลดเสียงในการลากหรือใช้อุปกรณ์ที่ส่งเสียงดัง (should reduce noise from dragging and using tools)
4	ระดับดุสิตธานี (like Dusit Thani hotel)
5	บริกรจำได้ว่าแขกชอบ/ไม่ชอบอะไร (waiter could remember guests'preferences)

Table 8: Examples of difficult cases of *service* feature

5 Conclusion and future work

We proposed a framework for constructing Thai opinion mining resource with a case study on hotel reviews. Two sets of lexicons, domain-dependent and domain-independent, are designed to support the pattern extraction process. The proposed method first constructs a set of patterns from a tagged corpus. The extracted patterns are then used to automatically extract and collect more sub-features and polars words from an untagged corpus. The performance evaluation was done with a collection of hotel reviews obtained from a hotel reservation website. From the experimental results, polar words could be extracted more easily than sub-features. This is due to the polar words often appear in specific positions with repeated contexts in the opinionated texts. In some cases, extraction of sub-features and polar words are not straightforward due to the difficulties in generalizing patterns. For example, some subjectivity requires complete phrases to describe the polarity. In some cases, the sub-features are not explicitly shown in the sentence. For future work, we plan to complete the construction of the corpus by considering the rest of main features. Another plan is to include the semantic analysis into the pattern extraction process. For example, the phrase "forget something" could imply negative polarity for the service feature.

References

Banea, Carmen, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. *Proc. of the 2008 empirical methods in natural language processing*, 127–135.

Beineke, Philip, Trevor Hastie and Shivakumar Vaithyanathan. 2004. The sentimental factor: improving review classification via human-provided information. *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, 263–270.

Cooke, J.R. 1992. Thai sentence particles: putting the puzzle together. *Proc. of the The Third International Symposium on Language and Linguistics*, 1105–1119.

Dave, Kushal, Steve Lawrence and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proc. of the 12th international conference on World Wide Web*, 519–528.

Ding, Xiaowen, Bing Liu and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. *Proc. of the int. conf. on web search and web data mining*, 231–240.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. *Proc. of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177.

Jin, Wei, Hung Hay Ho and Rohini K. Srihari. 2009. OpinionMiner: a novel machine learning system for web opinion mining and extraction. *Proc. of the 15th ACM SIGKDD*, 1195–1204.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. *Proc. of the 20th international conference on Computational Linguistics*, 1367–1373.

Qiu, Guang, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. *Proc. of the 21st International Joint Conferences on Artificial Intelligence*, 1199–1204.

Jijkoun, Valentin and Katja Hofmann. 2009. Generating a non-English subjectivity lexicon: relations that matter. *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 398–405.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing*, 355–363.

Ku, Lun-Wei and Hsin-Hsi Chen. 2007 Mining opinions from the Web: Beyond relevance retrieval. *Journal of American Society for Information Science and Technology*, 58(12):1838–1850.

Ku, Lun-Wei, Ting-Hao Huang and Hsin-Hsi Chen. 2009. Using morphological and syntactic structures for Chinese opinion analysis. *Proc. of the 2009 empirical methods in natural language processing*, 1260–1269.

Liu, Bing, Minqing Hu and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the Web. *Proc. of the 14th World Wide Web*, 342–351.

Pang, Bo, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. *Proc. of the ACL-02 conf. on empirical methods in natural language processing*, 79–86.

Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. *Proc. of the conf. on human language technology and empirical methods in natural language processing*, 339–346.

Riloff, Ellen and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. *Proc. of the 2003 conference on empirical methods in natural language processing*, 105–112.

Sarmento, Luís, Paula Carvalho, Mário J. Silva, and Eugénio de Oliveira. 2009. Automatic creation of a reference corpus for political opinion mining in user-generated content. *Proc. of the 1st CIKM workshop on topic-sentiment analysis for mass opinion*, 29–36.

Turney, Peter D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Proc. of the 40th ACL*, 417–424.

Wan, Xiaojun. 2009. Co-training for cross-lingual sentiment classification. *Proc. of the joint conf. of ACL and IJCNLP*, 235–243.

Wiebe, Janyce and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. *Proc. of Conference on Intelligent Text Processing and Computational Linguistics*, 486–497.

Wiebe, Janyce, Theresa Wilson and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wilson, Theresa, Janyce Wiebe and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Comput. Linguist.*, 35(3):399–433.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 129–136.

The Annotation of Event Schema in Chinese

Hongjian Zou[1], Erhong Yang[1], Yan Gao[2], Qingqing Zeng[1]
[1]Institute of Applied Linguistics, Beijing Language and Culture University
[2]Zhanjiang Normal University
hongjianzou@gmail.com, yerhong@blcu.edu.cn

Abstract

We present a strategy for revealing event schema in Chinese based on the manual annotation of texts. The overall event information is divided into three levels and events are chosen as the elementary units in annotation. Event-level annotation content and the obtaining of events patterns are explored in detail. The discourse-level annotation, annotation of relations between events and annotation of the functional attributes provide a simple way to represent event schema.

1 Introduction

When we want to understand a report on occurrences, we need to catch the following information: the categorization of events, the relationships between them, the participants and the attributes of the events such as polarity and modality, the attitudes towards the events and the following actions or consequences. Only the information above cannot be the precisely descried. Furthermore, we need to form a schema which incorporates all of the above, that is, to compile all this information together to get the integral structure about the report.

The available annotated corpora concerning the different types of information mentioned above include: the event-annotated corpora such as ACE corpora, the corpora annotating temporal information such as TimeBank, the corpora annotating event factuality such as FactBank, the corpora annotating various types of discourse relations such as RST corpus and Penn Discourse TreeBank. Meanwhile, we lack the annotation of event schema, which is important for providing the integral meaning of the reports.

Currently for Chinese language, the annotation of event information corpora is just beginning and still far from being sufficient, when compared with English, hence it needs further exploration.

2 Related Work

The work and theories concerning event schema annotation can be divided into three categories. The first kind is focused on annotation of the event argument structure, such as in ACE. The second kind is focused on annotation of the temporal information and event factuality. The last is focused on the annotation of the relations among different discourse units such as RST corpus and Penn Discourse TreeBank.

ACE(2005) is an in-depth study of research oriented annotated corpus for the purpose of textual information extraction. The annotation task includes event annotation besides the annotation of entities, values and relations between entities. The event annotation is limited to certain types and subtypes of events, that is, *Life, Movement, Transaction, Business, Conflict, Contact, Personnel*, and *Justice*. The argument structure of events including participants and other components such as time and place are predefined and tagged. Besides these, four kinds of attributes of events, *polarity*, *tense*, *genericity* and *modality*, are tagged. The expression characters of events, including the extent and the triggers, are also tagged.

TimeML(Pustejovsky et al., 2003; TimeML, 2005) is a system for representing not only all events but also temporal information. The events tagged are not limited to certain types as in ACE, but are classified in a different way. Event tokens and event instances are distin-

Proceedings of the 8th Workshop on Asian Language Resources, pages 72–79,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

guished and tagged respectively. For each event instance, four kinds of attributes, namely, *tense*, *aspect*, *polarity* and *modality* are tagged. TimeML defines three kinds of links between events and times. *TLINK* represents temporal relationships, including *simultaneous*, *before* and *after*. *SLINK* represents subordinative relationships. And *ALINK* represents relationships between an aspectual event and its argument event. Several TimeML corpora have been created now, including TimeBank and AQUAINT TimeML Corpus.

FactBank(Roser and Pustejovsky, 2008, 2009; Roser, 2008) is a corpus that adds factuality information to TimeBank. The factual value of events under certain sources is represented by two kinds of attributes, *modality* and *polarity*.

Besides the annotation of events and their temporal relationships or factuality information, there are various types of discourse annotation, which can be divided into two trends: one under the guidance of a certain discourse theory(such as RST) and the one independent of any specific theory(such as PDTB).

RST (Mann and Thompson, 1987; Taboada and Mann, 2006) was originally developed as part of studies on computer-based text generation by William Mann and Sandra Thompson in 1980s. In the RST framework, the discourse structure of a text can be represented as a tree. The leaves of the tree correspond to text fragments that represent the minimal units of the discourse; the internal nodes of the tree correspond to contiguous text spans; each node is characterized by its *nuclearity* and by a rhetorical relation that holds between two or more non-overlapping, adjacent text spans. RST chooses the clause as the elementary unit of discourse. All units are also spans, and spans may be composed of more than one unit. RST relations are defined in terms of four fields: (1) Constraints on the nucleus; (2) Constraints on the satellite; (3) Constraints on the combination of the nucleus and the satellite; and (4) Effects. The number and the types of relations are not fixed. It can be reduced or extended. Carlson et al. (2003) describes the experience of developing a discourse-annotated corpus grounded in the framework of Rhetorical Structure Theory. The resulting corpus contains 385 documents selected from the Penn Treebank.

Penn Discourse TreeBank(Miltsakaki et al., 2004; Webber et al., 2005) is to annotate the million-word WSJ corpus in the Penn TreeBank with a layer of discourse information. Although the idea of annotating connectives and their arguments comes from the theoretical work on discourse connectives in the framework of lexicalized grammar, the corpus itself is not tied to any particular theory. Discourse connectives were treated as discourse-level predicates of binary discourse relations that take two abstract objects such as events, states, and propositions. The two arguments to a discourse connective were simply labeled Arg1 and Arg2.

3 The Levels and Elementary Unit of Event Schema Annotation

3.1 The Elementary Unit of Event Schema Annotation

What counts as an elementary unit of Event Schema annotation in Chinese?

It is common to set sentences or clause as the basic units in discourse annotation such as RST corpus. However, there will be certain limitations if we choose sentences or clauses as the elementary units of Chinese event schema annotation:

First, a Chinese **sentence** is generally defined as a grammatical unit that has pauses before and after it, a certain intonation, and expresses a complete idea. But the definition is not exact or operational. The only notable borders of Chinese sentences in writings are the punctuations at the end of the sentences. The same is true of clauses in Chinese.

Second, there is generally more than one event in a sentence or a clause in Chinese. Hence, if we choose sentences or clauses as the basic units of event schema annotation, the relations between the events in one sentence/clause cannot be described in detail. For example:

1. 不到24小时，俄罗斯南部黑海边一个老年人之家又 ***燃起熊熊大火*** *，至少62人* ***葬身*** *火海。(In less than 24 hours, a* ***fire*** *swept through an old people's home in the Black Sea coast of southern Russia and* ***killed*** *at least 62 people.)*

2. 智利南部艾森大区自22日以来频繁发生 ***地震*** *，该地区政府已* ***宣布*** *该地区进入"早期警报"状态。(* ***Earthquakes*** *have hit the Aysen* *region in southern Chile frequently since the 22nd. The government has* ***declared*** *the region to be a state of "early warning".)*

In example 1, there are two events in bold type: the fire and the death in one sentence. In example 2, there are also two events in a single sentence: the earthquake and the declaration.

The “event” in this paper covers the same meaning defined by ACE(2005), which refers to "a specific occurrence involving participants".

Zou and Yang(2007) shows that an average of 2.3 times events per sentence are reported in Chinese texts and hence chose events as the basic discourse unit in their annotation. This consideration also fits the elementary unit of event schema annotation.

3.2 Three Levels of Event Schema Annotation

The overall event information in a report is complex and consists of different levels. In order to simplify the annotation task, we first divide the total event information into three levels, that is, the discourse level, the event level, and the entity level, choosing the event as the elementary unit of the event schema annotation.

The event level is defined as the level relating to atomic events. A report of occurrences always has many related events that are very easy to recognize. The events are atomic, which means the events are divided into small and minimal events. For example, when reading a report about an *earthquake* that happened in Haiti, the reader will not only know about the *earthquake* itself, but also other relating happenings such as the number of *casualty* or the following *search* and *rescue*. These things are divided into different atomic events, though they are still linked closely.

The entity level means the entities, times, and locations that are involved in events. For example, in “*China rescues 115 from a flooded mine*”, “*China*” is the agent of the rescue; “*115(miners)*” are the recipients; “*a flooded mine*” is the location. These three entities are the arguments of the *rescue* event and should be annotated before tagging them as the arguments of the *rescue* event.

The discourse level is the level above the event level which creates the integral meaning of the event schema. For example, the report concerning the rescue of miners from a flooded mine involves the *rescue*, the *coalmine accident* and possibly *injuries*. These events are linked together but have different significances within the report. So it is necessary to annotate the different significances of the events, as well as relations between events.

The following passages discuss in detail the event-level and the discourse-level annotation, while the entity-level annotation will not be discussed considering its relative simplicity.

4 Event-level Annotation

4.1 Definition of Events

ACE(2005) defines an **event** as follows: An event is a specific occurrence involving participants. An event is something that happens. An event can frequently be described as a change of state. According to ACE’s definition, we define **event** as the following: An Event is an occurrence that catches somebody's attention and a change of state.

4.2 Obtainment of Event Patterns

The event patterns are the argument structures of certain types of events, which are the directors of argument annotation. They are extracted from large-scale texts category by category. The above categories are based on the classification of sudden events. In other words, sudden events are divided into 4 categories: *natural disasters, accidental disasters, public health incidents*, and *social security incidents*, and each category includes different types of events, for example, the *natural disasters* includes *earthquakes, tsunamis, debris flows* and so on. In dealing with a specific kind of texts, only the closely related events that appear frequently are annotated. For example, when annotating the events of *earthquake*, only *earthquake* itself and closely related events such as *loss, rescue*, etc, are annotated.

The event patterns are manually extracted from real texts as follows, taking *earthquake* for instance:

- A search engine is used to obtain the reports whose titles and main bodies contain the key word ‘earthquake’, and then manually filter out those texts whose topics are not;
- The remaining texts are then split into sentences and only the sentences that narrate an earthquake or are closely relate to the earthquake are selected;

- Specific entities in these sentences are replaced with general tags such as '<TIME>', '<PER>' and '<LOC>' to get the patterns for earthquake type events;
- Frequently used patterns for earthquake events are extracted from the descriptions;
- The arguments of the event are numbered in sequence, and given corresponding explanations;
- The arguments are appended to event patterns when new roles are found.

The following principles should be abided by when extracting event patterns:

- Event triggers are the words or expressions that indicate existence of an event or events. If there is an event trigger in a sentence, we consider that there exists a corresponding event;
- Event triggers of different categories indicate different kinds of events;
- Some arguments of an event can be indistinct in a sentence. In other words, the different roles of the same event need to be merged into different patterns to get the complete argument structure of a certain event.

Some arguments are common roles in many events, such as *time*, *location*, and some arguments are specific to some events, such as *the magnitude*, and *the focus of an earthquake*. After the extraction of a certain amount of patterns, we can then merge the similar events. So far, we have obtained 31 categories of event patterns for 4 topics of news events.

Here is the event pattern corresponding to the *earthquake* event type extracted:

arg0	Time
arg1	Location
arg2	Magnitude
arg3	Epicenter
arg4	Focus
arg5	Focal depth
arg6	Quake-feeling locations
arg7	Frequency

Table 1. The earthquake event pattern.

4.3 Annotation of Types and Arguments

After obtaining the event patterns, we can annotate the types and the arguments of events according to the predefined types and patterns. If a certain event is not yet defined, the annotator should tag the event as "Other" and retag it later after obtaining the pattern of that category provided that the category is not too rare in similar reports.

The annotation of arguments consists of two steps. Firstly, we locate the entities and other expressions that belong to the arguments of a certain event. Then, we locate the roles of fixed arguments according to the corresponding event pattern. The arguments of an event are sought in the scope of the sentence in which the event trigger appears.

For example, according to the earthquake event pattern listed before, the annotation of the following sentence would be as follows:

美国地质勘探局称，这起地震发生在当地时间 12 日下午 4 时 53 分，震中位于海地首都太子港西南方向 16 公里处，震源深度为 10 公里，强度达到里氏 7.0 级。(The earthquake, with a magnitude estimated at 7.0, struck Haiti at 4:53 p.m. local time and was centered about 16 kilometers southwest of Port-au-Prince, at a depth of 10 km, the U.S. Geological Survey reported.)

arg0 Time	当地时间 12 日下午 4 时 53 分(about 4:53 p.m. local time)
arg1 Location	
arg2 Magnitude	里氏 7.0 级 (7.0)
arg3 Epicenter	海地首都太子港西南方向 16 公里处 (16 kilometers southwest of Port-au-Prince)
arg4 Focus	
arg5 Focal depth	10 公里 (10 km)
arg6 Quake feeling locations	
arg7 Frequency	1

Table 2. The annotation of the Haiti Earthquake.

4.4 Annotation of Event Attributes

Besides the types and arguments, the attributes of events are also tagged, which is necessary for a comprehensive description of events. Based on the analysis of various attributes in the reports, we decided to annotate the following: *Polarity, Modality, Tense, Aspect, Level, Frequency, Source*, and *Fulfillment*. Among these attributes, *Polarity, Modality* and *Tense* are adopted by both ACE and TimeML. *Aspect, Frequency* and *Source* are adopted by TimeML. The primary reason for annotating these attributes is that they have an important role in describing events in detail and different values of some attributes can even imply a totally different meaning.

Polarity is whether the event happened or would happen. The value of polarity can only be one between "*Positive*" and "*Negative*". For example, in

"*所幸这起火灾没有造成人员**伤亡***" (*Fortunately the fires did not result in any casualties*)

the polarity of event "*伤亡*"(*injuries-or-deaths*) is "*negative*".

Modality is the possibility of the event. Currently, we divide modality simply into "*Asserted*" and "*Other*". For example, in

"*震区许多居民担心再次发生**海啸***" (*Many residents in earthquake-hit areas worry about a recurrence of the tsunami*)

the modality of event "*海啸*"(*tsunami*) is "Other".

Tense is the time the event happened compared with the time of the report. It can be "*Past*", "*Present*", "*Future*", or "*Underspecified*". For example, in

"*警方目前正在进行**调查**"(A Police investigation is under way)*

the tense of event "*调查*" (*investigation*) is "Present".

Aspect is whether or not the event is continuing or completed. It can be "*Progressive*", "*Perfective*" or "*Underspecified*". In the sentence above, the aspect of event "*调查*" (*investigation*) is "*Progressive*".

Level is the extent of the events. It can be "*Serious*", "*Medium*" or "*Slight*". If the annotator cannot make sure, it can also be ignored. For example, in

"*强烈**地震**袭击印尼*" (*Strong earthquake hits Indonesian*)

the level of the event "*地震*" (*earthquake*) is "*Serious*".

Frequency is how many times the event happened. Usually it is only once, yet sometimes, as mentioned above, it may be twice or more.

Source consists of the source of the information about a certain event and the time the information issued. If not specialized, the source is equal to the source of the report itself and the time of source is equal to the time that the report was issued. For example, in

"*巴黎警方 10 日透露*" (*according to statements by the Paris police on 10th*)

the source is "*巴黎警方*"(*the Paris police*) and the time issued is "*10 日*"(*the 10th*).

Fulfillment is an interesting attribute of events that deserves further study and will be discussed in another paper. This is an attribute which is only applicable to man-made events with an emphasized intention, in other words, it is not applicable to those events occurring naturally. It can be "*Fulfilled*", "*Unfulfilled*", or "*Underspecified*". For example, a rescue event is deliberate and has or will have a result. For example, in

"*中国成功**救**出被困 8 昼夜的 115 名矿工*" (*China rescues 115 from flooded mine after 8 days*)

the fulfillment of the event "*救*"(*rescue*) is "*Fulfilled*".

The complete attributes of an event can be represented as a complex feature set as shown below:

$$\begin{bmatrix} \text{Polarity：Positive/Negative} \\ \text{Modality：Asserted/Other} \\ \text{Tense：Past/Present/Future/Underspecified} \\ \text{Aspect：Perfective/Progressive/Underspecified} \\ \text{Level：Slight/Medium/Serious} \\ \text{Frequency：} n(n \geq 1) \\ \text{Source：}\begin{cases} \langle \text{time 1，Source 1} \rangle \\ \quad \ldots\ldots \\ \langle \text{time m，Source n} \rangle \end{cases} \\ \text{Fulfillment：Fulfilled/Unfulfilled/Underspecified} \\ \ldots\ldots \end{bmatrix}$$

Figure 1. The complex feature set of attributes.

4.5 Annotation of Indicators

The recognition of types, arguments and attributes of the events not only depends on the sense of the annotator, but also depends on linguistic indicators within the text. To locate the existences of an event and its types, the annotator should find the lexical evidence that we called an **Event Word** (ACE call it a **trigger**) which clearly indicates something that has happened. In the following sentence,

*巴黎警方 10 日透露，位于巴黎一区一座公寓楼 10 日凌晨发生严重**火灾**，造成至少 2 名妇女**死亡**，两名消防队员**重伤**和巨大财物**损失**。* (*According to statements made by the Paris police on the 10th, serious **fire** swept through an apartment building in district one in*

*Paris on the morning of the 10th, **killing** at least 2 women, seriously **injuring two firemen** and causing huge property **damage**.*)

The Event Words "火灾"(*fire*), "死亡"(*killing*), "重伤"(*injuring*) and "损失"(*damage*) in the sentence above indicate four events respectively.

Besides annotating Event Words for events, the annotator also needs annotating indicators from texts to help to locate the attributes of the events. The attributes annotated should be clearly indicated by some linguistic hints, so the value of a certain attribute will not be specified if the hints are not so clear.

5 Discourse-level Annotation

The purpose of discourse level annotation is to integrate the information from the event-level into a structure. We annotate two kinds of discourse information, the relationships among events as annotated before and the functional attributes of events, to represent the event schema.

5.1 Annotation of Relations among Events

The events in the same report are not self-sufficient or independent, but are linked by various relationships, such as the causal relationships between an earthquake and an injury.

Taking into account of both the frequency of relationships between events and the ease and accuracy of distinguishing them, we have decided to focus on the following: *causality*, *co-reference*, *sequential*, *purpose*, *part-whole*, *juxtaposition* and *contrast*.

Causality is very common in reports. If event A is responsible for the happening of event B, then there exists a causal relationship between A and B. For example, in

"海地首都太子港附近发生里氏 7.0 级强烈**地震**，造成一家医院**倒塌**，另有多座政府建筑**损毁。**" *(A magnitude 7.0 earthquake hit Haiti, causing a hospital to collapse and damaging government buildings in the capital city of Port-au-Prince.)*

there are three events, called "地震"(*earthquake*), "倒塌"(*collapsing*) and "损毁"(*damaging*), and a causal relationship between "地震" and "倒塌" /"损毁".

Co-reference is not the relationship between two different events but the relationship between two expressions of events that refer to the same object.

Sequential is the relation between A and B such that B follows A chronologically but there is not necessarily a causal relationship between them. For example, in

"尼日利亚南部经济中心拉各斯一名 22 岁妇女 1 月 17 日因病**死亡**，后经尼卫生部门**检测**，该妇女死于高致病性禽流感。" (*A 22-year-old woman died of illness on Jan. 17 in Lagos, Nigeria's southern economic hub. After being tested by the Nigerian health sector, it was found that the woman had died of bird flu.*)

the events "死亡"(*death*) and "检测"(*testing*) have sequential relationship.

Purpose is the relation between A and B that A happened for B. For example, in

"尼日利亚政府目前已经在全国范围内加大了卫生**监管**力度，以**控制**高致病性禽流感的扩散。" (*The Nigerian government has already strengthened hygienic supervision and regulation nationwide to control the spread of the highly pathogenic avian influenza.*)

the purpose of the event "监管"(*supervision*) is to "控制"(*control*).

Part-whole relationship between A and B is when B is part of A. For example, in

"台风"桑美"给福鼎市带来重大人员**伤亡**，截至目前已有 138 人死亡，其中海上**遇难**人员 116 人，陆上遇难人员 22 人，还有 86 人失踪。" (*Saomai caused significant casualties in Fuding: at least 138 people have been killed so far, including 116 at sea, and 22 were on land, with 86 missing.*)

the event "遇难" (*killed*) appeared first and is part of the event "伤亡" (*casualties*).

Juxtaposition relationship means that A and B are caused by the same thing, or that A and B are simultaneous. For example, in

"大同市、左云县有关部门已对被困矿工家属进行了妥善**安置**。同时，环境部门正在对水质进行**监测**。" (*Datong, Zuoyun authorities have made proper arrangements for the families of trapped miners. Meanwhile, the department for environmental protection has been monitoring water quality.*)

the “*安置*”(*arrangement*) and “*监测*” (*monitoring*) are simultaneous.

Contrast relationship is when A would usually cause B, but here A happened and didn’t in fact cause B. For example, in

“*萨尔瓦多中部地区 2 日发生里氏 5．3 级* ***地震***，*但没有造成人员* ***伤亡*** *和财产损失。*” (*A 5.3 magnitude earthquake hit the central region of Salvador on the 2nd, but caused no casualties or property losses.*)

the “*地震*” (*earthquake*) usually causes “*伤亡*” (*casualties*), but here there is no “*伤亡*” .

The contrast relationship between A and B is not equal to the negation of a causal relationship, because in a contrast relationship A is positive and B is negative, while in the negation of causal relationship, the A is negative.

Besides those relationships between events described above, the annotator could tag the relation as “Underspecified” if he/she feels that relationship belongs to a new kind and deserves to be annotated.

These relations are also annotated with the attributes similar to those of events, but only including **Polarity**, **Modality**, **Tense**, **Aspect** and **Source**.

5.2 Annotation of Functional Attributes

The annotation of relations among events only represents the local discourse structure of the report. To represent the overall information it is necessary to integrate the event-level information globally. We find that the events annotated in one text are not owning equal significance, and they can be divided into at least two basic kinds according to their role in expressing the highlight of the text. The two basic kinds of role we decide to tag are “**core**” and “**related**”. We call this the **functional attribute** of the events.

The **core events** are the events that are the topics of the reports. Other events are the **related events**. If core events were removed, the elementary topics would change and the remaining events could not easily be organized together well. For example, in a report concerning the *earthquake* that happened in Haiti several months ago, the report’s core events are the events representing the *earthquake*. The other events such as the *rescue* or the *injuries* are not integral and cannot be meaningful alone. But if the other events were removed, the topic and logic of the report would still be clear, though the details might be somewhat incomplete.

After annotating the relationships among events and functional attributes of these events, we can represent a report about an earthquake which happened in Kyrgyzstan as follow:

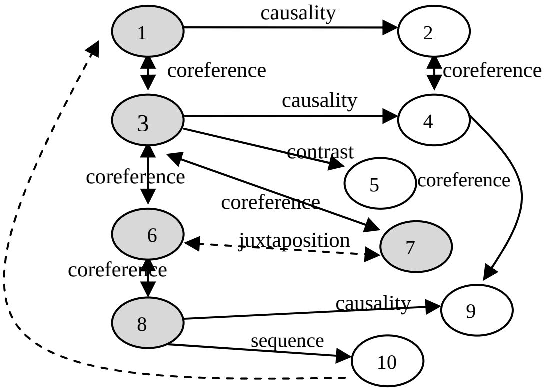


Figure 2. Event schema of Kyrgyzstan earthquake. Nodes of 1, 3, 6, 7 and 8 represent earthquakes; Nodes of 2, 4, and 9 represent damage; Node 5 represents casualty; Nodes 10 represents investigation.

In the graph above, the nodes represent the events, and the edges represent the relationships between events. The gray nodes represent the core events, while the white nodes represent the related events. As can be seen from the graph, the core events are at the center of the text and the related events are attached to the core events.

6 Preliminary Results and Discussion

In order to check the taggability of the annotation strategy mentioned above, three graduate students manually annotated about 60 news reports in 3 categories, including *earthquake*, *fire* and *attack*, using *sina* search engine, according to the method and principles above. Each text was annotated by two annotators and discussed jointly if the annotation results were inconsistent or not proper.

As can be seen from Table 3 below, 1) the event patterns extracted can cover the texts well because up to 78% sentences have been annotated. 2) There are 1.6 times more annotated events than annotated sentences. This shows that there is generally more than one event in a sentence. So, it is reasonable to assume that the annotation method can accomplish the task of a detailed description of relationships between events. 3) The relevant events are more numerous than the core events. This shows that it is necessary to distinguish the core events from the relevant events.

C	T	S	NS	EV	CE	RE	AR
C1	20	277	45	361	191	170	588
C2	20	309	66	394	183	211	515
C3	20	356	93	401	121	280	605
C4	60	942	204	1156	495	661	1708

Table 3. The annotation of EVENTs
C: Sub-category; C1: earthquake; C2: fire;
C3: terrorist attacks; C4: total
T: the number of texts; S: the number of sentences
NS: the number of sentences not annotated
EV: the number of EVENTs
CE: the number of core EVENTs
RE: the number of relevant EVENTs
AR: the number of arguments

We have also analyzed the event attributes in detail(Zou and Yang, 2010). An interesting event attribute is Fulfillment, which is only applicable to those events with intentions whose result is often emphasized. Sometimes, readers care about the intended results or outcomes as much as or more than the events themselves. Therefore it would be useful to explore the notion of Fulfillment, and investigate which linguistic categories could play a role in deciding the value of Fulfillment. We plan to create a Fulfillment corpus in the next stage.

The annotation of event schema is time-consuming, partly because it needs to annotate all three levels of event information of every text, and partly because of the difficulties to identify the event information from trivial descriptions, in other words, one question we often discuss is whether it deserves to annotate certain parts of a text. Also, we often need to make a balance between obtaining enough event patterns to cover various types of related events well and omitting low frequent event types to simply the obtainment of event patterns. In discourse-level annotation, the main difficulty is the identification of relations between events without lexical hints. This discourse-level annotation is only just underway. We also plan to give detailed analysis in the next stage.

Acknowledgements. This paper is sponsored by National Philosophy and Social Sciences Fund projects of China (No. 06YY047).

References

ACE. 2005. *ACE (Automatic Content Extraction) Chinese Annotation Guidelines for Events*. http://www.ldc.upenn.edu/Projects/ACE/docs/Chinese-Events-Guidelines_v5.5.1.pdf

Carlson L., D. Marcu, M. E. Okurowski. 2003. *Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory*. Current Directions in Discourse and Dialogue, Jan van Kuppevelt and Ronnie Smith eds., Kluwer Academic Publishers.

Mann W. and S. Thompson, 1987. *Rhetorical Structure Theory: A Theory of Text Organization* (No. ISI/RS-87-190). Marina del Rey, CA, Information Sciences Institute.

Miltsakaki E., R. Prasad, A. Joshi, and B. Webber. 2004. *Annotating Discourse Connectives and their Arguments*. Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation. Boston, MA.

Pustejovsky J., J. Castaño, R. Ingria, S. Roser R. Gaizauskas, A. Setzer and G. Katz. 2003. *TimeML: Robust Specification of Event and Temporal Expressions in Text*. Fifth International Workshop on Computational Semantics.

Taboada M. and W. Mann. 2006. *Rhetorical Structure Theory: Looking Back and Moving Ahead*. Discourse Studies 8(3): 423-459.

TimeML. 2005. *Annotation Guidelines Version 1.2*. http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf.

Webber B., A. Joshi, E. Miltsakaki, et al. 2005. *A Short Introduction to the Penn Discourse TreeBank*. Copenhagen Working Papers in Language and Speech Processing.

Roser S. 2008. *A Factuality Profiler for Eventualities in Text*. Ph.D. Thesis. Brandeis University.

Roser S. and J. Pustejovsky. 2008. *From Structure to Interpretation: A Double-layered Annotation for Event Factuality*. Prooceedings of the 2nd Linguistic Annotation Workshop.

Roser S. and J. Pustejovsky. 2009. *FactBank: A Corpus Annotated with Event Factuality*. Language Resources and Evaluation.

Zou H.J. and E.H. Yang. 2007. *Event Counts as Elementary Unit in Discourse Annotation*. International Conference on Chinese Computing 2007.

Zou H.J. and E.H. Yang. 2010. *Annotation of Event Attributes*. The 11th Chinese Lexical Semantics Workshop.

Query Expansion for Khmer Information Retrieval

Channa Van and Wataru Kameyama
GITS, Waseda University
Honjo, Saitama, Japan
channa@fuji.waseda.jp, wataru@waseda.jp

Abstract

This paper presents the proposed Query Expansion (QE) techniques based on Khmer specific characteristics to improve the retrieval performance of Khmer Information Retrieval (IR) system. Four types of Khmer specific characteristics: spelling variants, synonyms, derivative words and reduplicative words have been investigated in this research. In order to evaluate the effectiveness and the efficiency of the proposed QE techniques, a prototype of Khmer IR system has been implemented. The system is built on top of the popular open source information retrieval software library Lucene[1]. The Khmer word segmentation tool (Chea et al., 2007) is also implemented into the system to improve the accuracy of indexing as well as searching. Furthermore, the Google web search engine is also used in the evaluation process. The results show the proposed QE techniques improve the retrieval performance both of the proposed system and the Google web search engine. With the reduplicative word QE technique, an improvement of 17.93% of recall can be achieved to the proposed system.

1 Introduction

Similar to the other major languages in the world, the number of Khmer digital content has been rapidly growing over the world-wide web, and it is becoming very difficult to obtain the relevant information as needed from the Internet. Although some major web search engine providers such as Google has a localized version of Khmer in its web search engine[2], it is not specifically designed for Khmer due to the lack of integration of Khmer specific characteristics. Consequently, it misses a lot of information of Khmer websites found in the Internet. For this reason, we propose the QE techniques using the specific characteristic of Khmer to improve the effectiveness of Khmer IR system. Four types of QE technique are proposed based on the four types of Khmer specific characteristic that are spelling variants, synonyms, derivative words (Khin, 2007) and reduplicative words (Long, 2007). Moreover, a prototype of Khmer IR system is implemented in order to evaluate the effectiveness of these proposed QE techniques. The proposed system is built on top of the popular open source information retrieval software library Lucene in order to take the advantage from its powerful indexing and searching algorithms. Furthermore, to improve the accuracy of indexing and searching, we have also implemented the specific Khmer word segmentation into the system. Due to the lack of Khmer text collection which is required in the evaluation process, we have also created our own Khmer text corpus. The corpus has been built to be useful and beneficial to all types of the research in Khmer Language Processing.

2 Khmer Specific Characteristics

Khmer is the official language of Cambodia. It is the second most widely spoken Austroasiatic language family[3]. Due to the long historical contact

[1] Apache Lucene: http://lucene.apache.org

[2] Google Khmer: http://www.google.com.kh

[3] Khmer language: http://en.wikipedia.org/wiki/Khmer_language

Proceedings of the 8th Workshop on Asian Language Resources, pages 80–87,
Beijing, China, 21-22 August 2010.

with India, Khmer has been heavily influenced by the Sanskrit and Pali. However, Khmer still possesses its own specific characteristics so far such as the word derivation rules and word reduplication techniques. Furthermore, the specific written rule is also found in Khmer, for instance the case of multiple spelling words. These characteristics are very useful especially in the Khmer IR due to the lexical-semantic relation between the words.

2.1 Spelling Variants

In Khmer, multiple spelling words exist. They have same meaning and pronunciation but only different in spelling. Most of the spelling variants are loan words, and others are the result of the substitutability between characters. For example:

- The word សមុទ្រ [sămŭt] "sea" ,which is originated from Sanskrit, can also be spelled សមុទ្ទ [sămŭt] "sea" (Khmer Dictionary, 1967) which is originated from Pali.
- And the word ឫទ្ធិ [rœ̆tthĭ] "power" has another spelling រឹទ្ធិ [rœ̆tthĭ] "power" because "ឫ" can be substituted by "រឹ".

2.2 Synonyms

Synonyms exist in all the natural languages. Thus, there is no exception for Khmer as well. Khmer has rich and variety of synonym vocabularies. Most of these synonyms are found in the loan words (influenced by Sanskrit and Pali) and the social status's words. For instance the word ញ៉ាំ "to eat" has many synonyms for each social status: ស៊ី (impolite word), ពិសា (polite word), ឆាន់ (religious word), សោយ (royal word) and etc.

2.3 Derivative Words

Derivative words in Khmer are the words which are derived from the root words by affixation, including prefixation, infixation and suffixation (Jenner, 1969). Interestingly, derivative word's meaning is semantically related to its root word. For example:

ចាស់ "old" + កញ (prefix) →កញ្ចាស់ "very old"

ដើរ "to walk" + មណ (infix) →ដំណើរ "the walk"

ឯករាជ្យ "independence" + ភាព (suffix) →
ឯករាជ្យភាព "the state of independence"

2.4 Reduplicative Words

Reduplicative words are very common in Khmer. They are used for several purposes including emphasis, pluralization and complex thought expressions. Three kinds of duplicating techniques are found in Khmer: word duplication, synonym duplication and symmetrical duplication (Noeurng and Haiman, 2000).

2.4.1 Word duplication

Word duplication is the process of duplicating a word to express pluralisation meaning. The duplication symbol "ៗ" is put after the word to indicate the duplication. For example:

ឆ្កែធំ [chhkê thum] "a big dog" →
ឆ្កែធំៗ [chhkê thum thum] "many big dogs"

2.4.2 Synonym duplication

Also called synonym compounding words. This kind of words are created by combining two different words which are either synonyms or related meaning. For example:

បំណង "goal" + ប្រាថ្នា "intention" →
បំណងប្រាថ្នា "goal and intention"

2.4.3 Symmetrical duplication

Symmetrical duplication is the process of creating a word by combining a word and its similar phonetic syllables. It is similar to create words which sound like "*zigzag*" or "*hip hop*" in English. This duplication technique is usually used to emphasize the meaning of the original word. There are quite remarkable amount of this kind of words found in Khmer. For example:

ធំ [thum] "big" + ធេង [théng] (similar phonetic syllable) →ធំធេង [thum théng] "very big"

ក្រី [krei] (similar phonetic syllable) + ក្រ [krar] "poor" →ក្រីក្រ [krei krar] "very poor"

3 Khmer Text Corpus

A large collection of Khmer text is required in the evaluation process. Due to the lack of Khmer language resource especially the Khmer text corpus, we have built our own Khmer text corpus. The corpus was designed to be useful and beneficial to all

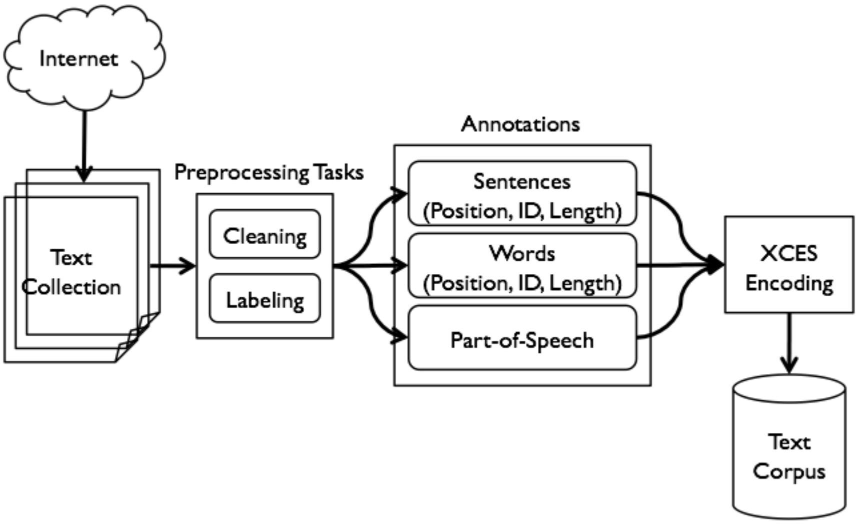


Figure 1: System Design of Building a Khmer Text Corpus

kinds of the research in Khmer language processing. Building such text corpus for Khmer is a challenging task since there is no implementation yet on Khmer optical character recognition, and a few research in Khmer have been done . All texts have to be collected manually from various Khmer website in the Internet. The corpus includes some basic corpus annotations such as word annotation, sentence annotation and part-of-speech annotation. It was encoded in eXtensible Corpus Encoding Standard (XCES) (Nancy et al., 2000) to assure the future extensibility. Figure 1 illustrates the whole process of building a Khmer text corpus in which four main steps were carried out: text collection, preprocessing tasks, corpus annotations and corpus encoding. The detail of each step is described in the following subsections:

3.1 Text Collection

Collecting Khmer digital text is the most difficult and time consuming task in the process of building this corpus. As there is no implementation on Khmer optical character recognition, it is not possible to scan or extract Khmer texts neither from books nor from other paper sources. However, thanks to the websites as well as Khmer blog community that provide the valuable Khmer digital texts for this research. All text were manually collected from all the available sources.

3.2 Preprocessing Tasks

The texts collected from the Internet are usually not clean and unstructured. It may contain unwanted elements such as images, links and some HTML elements. Therefore, cleaning process is carried out to remove the unwanted elements and to restructure the texts before proceeding to the next step.

After cleaning, each text is categorized by its domain according to its content by the labeling process. There are twelve domains in this corpus: newspaper, magazine, medical, technology, culture, history, law, agriculture, essay, novel, story and other. The text descriptions such as author's name, publisher's name, publishing date and publisher's digital address, are kept along with each text.

3.3 Corpus Annotations

Corpus annotation information is very important for the corpus-based research. Therefore, this corpus also includes the sentence annotation, word annotation and POS annotation.

3.3.1 Sentence Annotation

Each sentence is annotated with three kinds of information: position, identification and length.

1. Position: it is defined by the position of the first character and the last character of a sentence in the text.

2. Identification: the sequence number of a sentence within a text file.

3. Length: the number of characters of a sentence.

Like English, each sentence in Khmer can be separated by special symbols. In modern Khmer, there exists a series of characters that are used to mark the boundaries of different kind of sentences (Khin, 2007). Based on these characters, each sentence in a text can be separated easily.

- ។ and ។ល។ : end of declarative sentences.
- ? : end of interrogative sentences.
- ! : end of exclamative sentences.
- ៕ end of the last sentence in a text.

3.3.2 Word Annotation

The position, identification and length of each word are also annotated as in sentence annotation.

1. Position: it is defined by the position of the first character and the last character of a word in the text.

2. Identification: the sequence number of a word within a text file.

3. Length: the number of characters of a word.

Khmer is non-segmented script. There is no separator between words which is very difficult to segment. In order to do that, we have used the Khmer word segmentation tool (Chea et al., 2007) developed by PAN Localization Cambodia[4].

3.3.3 Part-of-Speech Annotation

To enhance the usefulness of the corpus, we also include the Part-of-Speech annotation. We have used a Khmer POS tagger which is based on the work of Nou et al. where a transformation-based approach with hybrid unknown word handling for Khmer POS tagger is proposed (Nou et al., 2007). There are 27 types of Khmer tagset which can be obtained by this Khmer POS tagger. Each obtained POS tag is assigned to each word in the corpus, and it is kept along with the word annotation.

3.4 Corpus Encoding

To assure the extensibility of corpus and the facility of development for the future works, this corpus has been encoded in eXtensible Corpus Encoding Standard (XCES) (Nancy et al., 2000). XCES is an XML-based standard to codify text corpus. It is highly based on the previous Corpus Encoding Standard (Nancy, 1998) but using XML as the markup language. Since the corpus encoding is based on XML, the corpus is suitable for many programming languages which support XML. Furthermore, it can fully take the advantage of the powerful XML framework including XQuery, XPath and so on. In addition, XCES supports many types of corpora especially the annotation corpora which our corpus is based on. The encoding of annotation files and text description files are conformed to XCES schema version 1.0.4.

3.5 Corpus Statistic

Table 1 shows the corpus statistic. We have achieved more than one million words within twelve different domains of text. The corpus size is relatively small at the moment, the expansion of the corpus size is continuously undergoing.

Table 1: Corpus Statistic

Domain	**# of Article**	**# of Sentence**	**# of Word**
Newspaper	571	13222	409103
Magazine	52	1335	42566
Medical	3	76	2047
Technical	15	607	16356
Culture	33	1178	43640
Law	43	5146	101739
History	9	276	7778
Agriculture	29	1484	30813
Essay	8	304	8318
Story	108	5642	196256
Novel	78	12012	236250
Other	5	134	5522
Total	**954**	**41416**	**1100388**

4 Retrieval Environment

This section provides the necessary background to understand the context in which the experiment

[4] Cambodia PAN Localization: `http://www.pancambodia.info/`

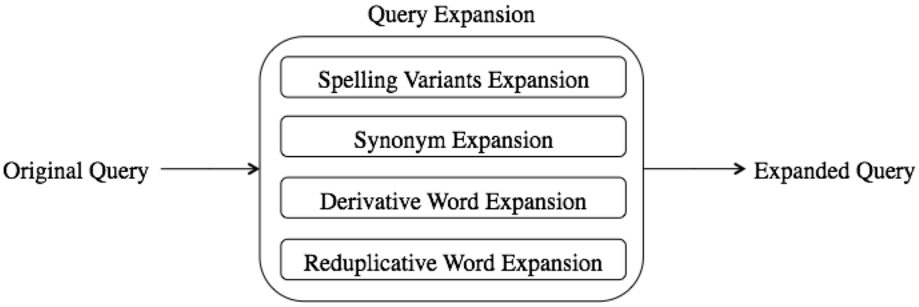

Query Expansion
Spelling Variants Expansion
Synonym Expansion
Derivative Word Expansion
Reduplicative Word Expansion
Original Query
Expanded Query

Figure 2: Query Expansion Procedures

was carried out.

4.1 Query Expansion Procedures

Query Expansion is a technique commonly used in IR (Manning et al., 2009) to improve retrieval performance by reformulating the original query. We have proposed four types of QE technique based on the four types of Khmer specific characteristics that we have presented in the section 2. During the expansion process, the original search query is analyzed and expanded corresponding to the type of words (Figure 2). Four types of QE is carried out: spelling variants expansion, synonym expansion, derivative word expansion and reduplicative word expansion. The expanded query is obtained after adding the the expansion term to the original query.

4.2 Khmer IR System Design and Implementation

A prototype of Khmer IR system, which is shown in the Figure 3, has been implemented to evaluate the efficiency of the proposed QE techniques. There are two main processes in the system implementation: indexing and searching. We have started to implement the system by constructing searching index from the text corpus. All texts in the corpus are tokenized into words. Then these words are indexed by the Lucene's indexer, and stored in an index database. On the other hand in the searching part, the search query is tokenized into words before being analyzed in the QE process. Finally, the search results are obtained by the Lucene's searcher which searches through the index database and returns results that correspond to the expanded query.

4.3 Indexing

Indexing is very important because it determines what search results are returned when a user submits query. Our proposed system's indexer is based on the Lucene's indexer with the modifications adapted to Khmer language. Lucene indexes a document by indexing the tokens, which are words, obtained by tokenizing text of the document (Hatcher et al., 2009). Since the default Lucene's tokenizer can only work with segmented western languages such as English or French where spaces are used between each word, it is impossible to tokenize document in Khmer which belongs to the class of non-segmenting language group, where words are written continuously without using any explicit delimiting character. Khmer word tokenizer, which is developed by the a research group of Cambodia PAN Localization, has been used to handle this task.

5 Experiments

The experiment to evaluate the proposed QE techniques was initially conducted only on the prototype of the Khmer IR system that we have implemented. As Google also has a localized Khmer version of its popular web search engine, and it can explicitly specify websites to be searched[5], we have extended our experiment to Google web search engine in order to obtain more precise result.

5.1 Experiment Setup

The Khmer text corpus, which consists 954 documents collected from various websites in the In-

[5] `http://www.google.com/advanced_search`

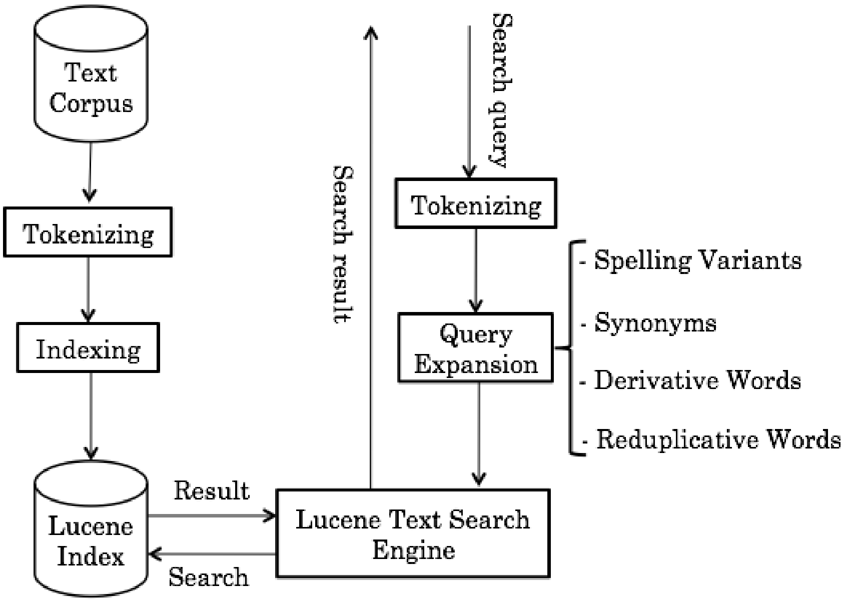


Figure 3: Proposed Khmer Information Retrieval System Design

ternet, was used for the experiments. A website, which contains all documents from the corpus, was hosted in our laboratory's web server in order that these documents can be indexed by the Google's indexer. Then we followed up the indexing progress by consulting the Google Webmaster Tools[6] service. In Khmer where words are written one after the other, word processing programs, Internet browsers and other programs that format text need to know where they can cut a sentence in order to start a new line. This kind of problem does not appear in the western languages where space are used between words. Thus, the zero-width space character was proposed to solve this problem by inputting the zero-width space at the end of each word while typing[7]. In Unicode, the zero-width space character is a type of space character but it is invisible. Using the zero-width space is very confusing because of the invisibility of the character, plus it is unnatural to Khmer writing system. As a result, most people only partly used it for the text display purpose. Therefore, we can find the zero-width space in almost all Khmer texts found in the Internet. Since all texts in the corpus are collected from the Internet, the zero-width space also can be found in almost all the texts in the corpus. Based on the zero-width space character, the Google can separate and index the Khmer texts in our text corpus hosted in our laboratory web server. After Google completely indexed all the documents, the experiment was proceeded.

5.2 Experiment Procedures

We conducted the experiment for each type of proposed QE technique in our implemented system and the Google web search engine. Four similar experiments of the four proposed QE techniques were carried out to the both systems. Due to the small size of the text corpus, only ten original queries have been chosen for each type of experiment. Each query possesses a specific topic in order that we can judge the relevant results after. The experiment processes are as following:

1. Input ten original queries into the both systems, and calculate the precisions and recalls. All queries are selected from the different topics, and each query contains at least an expandable word corresponding to its expansion type.

2. Expand the original queries according to their expansion type. Then input the ten expanded

[6] `http://www.google.com/webmasters/tools`

[7] Khmer Unicode typing: `http://www.khmeros.info/download/KhmerUnicodeTyping.pdf`

Table 2: Results of Spelling Variants and Synonyms Expansions

	Spelling Variants			Synonyms		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Google	39.79%	46.72%	37.35%	38.91%	39.99%	34.66%
Proposed Sys.	62.09%	47.63%	50.78%	46.74%	47.03%	44.71%
Google & QE	46.83%	53.04%	46.13%	44.32%	58.99%	45.28%
Proposed Sys. & QE	60.73%	64.01%	58.38%	48.34%	64.59%	51.66%

Table 3: Results of Derivative Word and Reduplicative Word Expansions

	Derivative Words			Reduplicarive Words		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Google	28.35%	56.04%	36.56%	21.14%	42.61%	26.16%
Proposed Sys.	41.07%	51.07%	44.10%	31.69%	39.05%	29.00%
Google & QE	29.18%	60.41%	38.32%	24.69%	48.58%	26.35%
Proposed Sys. & QE	33.93%	62.38%	41.71%	34.28%	56.98%	36.25%

queries into the both systems and recalculate the precisions and recalls.

The relevance judgments were manually done based on the content of each document obtained by the both IR systems. Since there is no Khmer digital thesaurus yet, the expansions were manually done by respecting the query syntax of Lucene and Google[8]. Moreover, as Google web search engine cannot tokenize Khmer words, the tokenization was also done manually. For example: the query តុលាការខ្មែរក្រហម "Khmer Rough tribunal" consists two words តុលាការ "tribunal" and ខ្មែរក្រហម "Khmer Rough". We know that the synonym of តុលាការ "tribunal" is សាលាក្តី "tribunal". So the expanded query for our proposed system is "តុលាការខ្មែរក្រហម OR សាលាក្តីខ្មែរក្រហម". On the other hand, the expanded query for Google is "[តុលាការ OR សាលាក្តី] AND ខ្មែរក្រហម".

5.3 Results and Discussion

Table 2 and 3 show the results of precision, recall and F-Measure before and after implementing each QE techniques to the proposed system and Google web search engine. The improvement in recall of our proposed system is 16.38%, 17.56%, 11.31%, 17.93% after applying the respective QE techniques, while the increase in recall at 6.32%, 19.00%, 4.37%, 5.97% after applying the QE techniques respectively to Google web search engine.

In addition, comparing our proposed system with QE to Google web search engine without QE, the recall improvement is 17.29%, 24.60%, 6.34%, 14.37% respectively, while to Google web search engine with QE, the recall improvement is 10.97%, 5.60%, 1.97%, 8.40% respectively.

As a summary, the search results using our proposed system with QE techniques is significantly better than the conventional Google search results. This can be seen clearly from the improvement of F-Measure 21.03%, 17.00%, 5.15%, 10.09% respectively.

6 Conclusion and Future Works

In this research, we have investigated four types of QE technique based on Khmer linguistic characteristics. These QE techniques are specifically designed to improve the retrieval performance of Khmer IR System. The experiments have demonstrated the improvement of retrieval performance of the proposed system and the Google web search engine after applying the proposed QE techniques. However, the improvement in precision after utilizing our proposed QE techniques is not so significant. In the case of derivative words, it even shows some slight decrement. This is one of the

[8] `http://www.google.com/support/websearch/bin/answer.py?answer=136861`

main problems that we will tackle in our future research to reduce non-relevant contents by semantically analyzing Khmer content. At the moment the size of the corpus is very small, and we are actively dealing with this issue in hope to provide a good Khmer language resource for the future research in Khmer language processing.

In addition, due to the lack of research in Khmer IR as well as in Khmer language processing, a lot of aspects can still be worked on in order to improve the system performance. For example, the improvement of Khmer word segmentation and the building of Khmer thesaurus for IR system, which are expected to improve the IR system performance, are also in the priority tasks of our future works.

References

Beaulieu, Micheline and Susan Jones. 1998. Interactive searching and interface issues in the Okapi best match probabilistic retrieval system. *Interacting With Computers*, 10(3):237--248.

Buddhist Institute. 1967. វចនានុក្រមខ្មែរ "*Khmer Dictionary*". Buddhist Institute, Phnom Penh, Cambodia.

Chea, Sok-Huor, Rithy Top, Pich-Hemy Ros, Navy Vann, Chanthirith Chin and Tola Chhoeun. 2007. Word Bigram Vs Orthographic Syllable Bigram in Khmer Word Segmentation. *http://www.panl10n.net/english/OutputsCambodia1 .htm*, (Last retrieved 30 April 2010).

Hatcher, Erik, Otis Gospodnetić and Michael McCandless. 2009. *Lucene in Action, Second Edition*. Manning Publications, Connecticut, USA.

Ide, Nancy 1998. Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In *Proceedings of the First International Language Resources and Evaluation Conference*, pp. 463-70.

Ide, Nancy, Patrice Bonhomme and Laurent Romary 2000. XCES: An XML-based standard for linguistic corpora. In *Proceeding of Second Language Resources and Evaluation Conference (LREC)*, pp. 825--830, Athens. Greece.

Jenner, Philip Norman. 1969. Affixation in modern Khmer. *A dissertation submitted to the graduate division*, University of Hawaii.

Khin, Sok. 2007. វេយ្យាករណ៍ភាសាខ្មែរ "*Khmer Grammar*". Royal Academy of Cambodia, Phnom Penh, Cambodia.

Long, Siem. 1999. បញ្ហាវេចនស័ព្ទវិទ្យាខ្មែរ "*Khmer Lexicology Problem*". National Language Institute, Royal University of Phnom Penh, Phnom Penh, Cambodia.

Manning, Christopher D., Prabhakar Raghavan and Hinrich Schütze 2009. *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.

Nou, Chenda and Wataru Kameyama 2007. Khmer POS Tagger: A Transformation-based Approach with Hybrid Unknown Word Handling. In *Proceeding of International Conference on Semantic Computing (ICSC)*, pp. 482--492. Irvine, USA.

Ourn, Noeurng and John Haiman. 2000. Symmetrical Compounds in Khmer. *Studies in Language*. 24(3), pp. 483--514.

Singhal, Amit. 2001. Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35--43.

Word Segmentation for Urdu OCR System

Misbah Akram
National University of Computer and Emerging Sciences
misbahakram@gmail.com

Sarmad Hussain
Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
sarmad.hussain@kics.edu.pk

Abstract

This paper presents a technique for word segmentation for the Urdu OCR system. Word segmentation or word tokenization is a preliminary task for Urdu language processing. Several techniques are available for word segmentation in other languages. A methodology is proposed for word segmentation in this paper which determines the boundaries of words given a sequence of ligatures, based on collocation of ligatures and words in the corpus. Using this technique, word identification rate of 96.10% is achieved, using trigram probabilities normalized over the number of ligatures and words in the sequence.

1 Introduction

Urdu uses Nastalique style of Arabic script for writing, which is cursive in nature. Characters join together to form ligatures, which end either with a space or with a non-joining character. A word may be composed of one of more ligatures. In Urdu, space is not used to separate two consecutive words in a sentence; instead readers themselves identify the boundaries of words, as the sequence of ligatures, as they read along the text. Space is used to get appropriate character shapes and thus it may even be used within a word to break the word into constituent ligatures (Naseem 2007, Durrani 2008). Therefore, like other languages (Theeramunkong & Usanavasin, 2001; Wan and Liu, 2007; Khankasikam & Muansuwan, 2005; Haruechaiyasak et al., 2008; Haizhou & Baosheng, 1998), word segmentation or word tokenization is a preliminary task for Urdu language processing. It has applications in many areas like spell checking, POS tagging, speech synthesis, information retrieval etc. This paper focuses on the word segmentation problem from the point of view of Optical Character Recognition (OCR) System. As space is not visible in typed and scanned text, spacing cues are not available to the OCR for word separation and therefore segmentation has to be done more explicitly. This word segmentation model for Urdu OCR system takes input in the form of a sequence of ligatures recognized by an OCR to construct a sequence of words from them.

2 Literature Review

Many languages, e.g., English, French, Hindi, Nepali, Sinhala, Bengali, Greek, Russian, etc. segment text into a sequence of words using delimiters such as space, comma and semi colon etc., but on the other hand many Asian languages like Urdu, Persian, Arabic, Chinese, Dzongkha, Lao and Thai have no explicit word boundaries. In such languages, words are segmented using more advanced techniques, which can be categorized into three methods:

(i) Dictionary/lexicon based approaches
(ii) Linguistic knowledge based approaches
(iii) Machine learning based approaches/statistical approaches (Haruechaiyasak et al., 2008)

Longest matching (Poowarawan, 1986; Richard Sproat, 1996) and maximum matching (Sproat et al., 1996; Haizhou & Baosheng, 1998) are examples of lexicon based approaches. These techniques segment text using the lexicon. Their

Proceedings of the 8th Workshop on Asian Language Resources, pages 88–94,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

accuracy depends on the quality and size of the dictionary.

N-Grams (Chang et al., 1992; Li Haizhou et al., 1997; Richard Sproat, 1996; Dai & Lee, 1994; Aroonmanakun, 2002) and Maximum collocation (Aroonmanakun, 2002) are Linguistic knowledge based approaches, which also rely very much on the lexicon. These approaches select most likely segmentation from the set of possible segmentations using a probabilistic or cost-based scoring mechanism.

Word segmentation using decision trees (Sornlertlamvanich et al., 2000; Theeramunkong & Usanavasin, 2001) and similar other techniques fall in the third category of word segmentation techniques. These approaches use a corpus in which word boundaries are explicitly marked. These approaches do not require dictionaries. In these approaches ambiguity problems are handled by providing a sufficiently large set of training examples to enable accurate classification.

A knowledge based approach has been adopted for earlier work on Urdu word segmentation (Durrani 2007; also see Durrani and Hussain 2010). In this technique word segmentation of Urdu text is achieved by employing knowledge based on the Urdu linguistics and script. The initial segmentations are ranked using min-word, unigram and bigram techniques. It reports 95.8 % overall accuracy for word segmentation of Urdu text. Mukund et al. (2009) propose using character model along with linguistic rules and report 83% precision. Lehal (2009) proposes a two stage process, which first uses Urdu linguistic knowledge, and then uses statistical information of Urdu and Hindi (also using transliteration into Hindi) in the second stage for words not addressed in the first stage, reporting an accuracy of 98.57%.

These techniques use characters or words in the input, whereas an OCR outputs a series of ligatures. The current paper presents work done using statistical methods as an alternative, which works with ligatures as input.

3 Methodology

Current work uses the co-occurrence information of ligatures and words to construct a statistical model, based on manually cleaned and segmented training corpora. Ligature and word statistics are derived from these corpora. In the decoding phase, first all sequences of words are generated from input set of ligatures and ranking of these sequences is done based on lexical lookup. Top *k* sequences are selected for further processing, based on the number of valid words. Finally, the probability of each of the *k* sequences is calculated for the final decision. Details are described in the subsequent sections.

3.1 Data collection and preparation

An existing lexicon of 49630 unique words is used (derived from Ijaz et al. 2007). The corpus used for building ligature grams consists of half a million words. Of these, 300,000 words are taken from the Sports, Consumer Information and Culture/Entertainment domains of the 18 million word corpus (Ijaz et al. 2007), 100,000 words are obtained from Urdu-Nepali-English Parallel Corpus (available at www.PANL10n.net), and another 100,000 words are taken from a previously POS tagged corpus (Sajjad, 2007; tags of this corpus are removed before further processing). This corpus is manually cleaned for word segmentation errors, by adding missing spaces between words and replacing spaces with Zero Width Non-Joiner (ZWNJ) within words. For the computation of word grams, the 18 million word corpus of Urdu is used (Ijaz et al. 2007).

3.2 Count and probability calculations

Table 1 and Table 2 below give the counts for unigram, bigrams and trigram of the ligatures and the words derived from the corpora respectively.

Ligature Tokens	Ligature Unigram	Ligature Bigrams	Ligature Trigrams
1508078	10215	35202	65962

Table 1. Unigram, bigram and trigram counts of the ligature corpus

Word Tokens	Word Unigrams	Word Bigrams	Word Trigrams
17352476	157379	1120524	8143982

Table 2. Unigram, bigram and trigram counts of the word corpus

After deriving word unigrams, bigrams, and trigrams, the following cleaning of corpus is performed. In the 18 million word corpus, certain words are combined due to missing space, but are separate words. Some of these words occur with very high frequency in the corpus. For example "ہوگا" (*ho ga*, "will be") exists as single word rather than two words due to missing space. To solve this space insertion problem, a list of about 700 words with frequency greater than 50 is obtained from the word unigrams. Each word of the list is manually reviewed and space is inserted, where required. Then these error words are removed from the word unigram and added to the word unigram frequency list as two or three individual words incrementing respective counts.

For the space insertion problem in word bigrams, each error word in joined-word list (700-word list) is checked. Where these error words occurs in a bigram word frequency list, for example "کیا ہوگا" (*kiya ho ga* "will have done") exists in the bigram list and contains "ہوگا" error word, then this bigram entry "کیا ہوگا" is removed from the bigram list and counts of "ہو گا" and "کیا ہو" are increased by the count of "کیا ہوگا". If these words do not exist in the word bigram list then they are added as a new bigrams with the count of "کیا ہوگا". Same procedure is performed for the word trigrams.

The second main issue is with word-affixes, which are sometimes separated by spaces from the words. Therefore, in calculations, these are treated as separate words and exist as bigram entries in the list rather than a unigram entry. For example "صحت مند" (*sehat*+*mand*, "healthy") exists as a bigram entry but in Urdu it is a single word. To cope with this problem, a list of word-affixes is used. If any entry of word bigram matches with an affix, then this word is combined by removing spurious space from it (and inserting ZWNJ, if required to maintain its glyph shape). Then this word is inserted in the unigram list with its original bigram count and unigram list updated accordingly. Same procedure is performed if a trigram word matches with an affix.

After cleaning, unigram, bigram and trigram counts for both words and ligatures are calculated. To avoid data sparseness One Count Smoothing (Chen & Goodman, 1996) is applied.

3.3 Word sequences generation from input

The input, in the form of sequence of ligatures is used to generate all possible words. These sequences are then ranked based on real words. For this purpose, a tree of these sequences is incrementally built. The first ligature is added as a root of tree, and at each level two to three additional nodes are added. For example the second level of the tree contains the following tree nodes.

- Current ligature forms a separate word, separated with space, from the sequence at its parent, l_1 l_2
- Current ligature concatenates, without a space, with the sequence at its parent, l_1l_2
- Current ligature concatenates, without a space, with the sequence at its parent but with an additional, l_1ZWNJl_2

For each node, at each level of the tree, a numeric value is assigned, which is the sum of squares of the number of ligatures in each word which is in the dictionary. If a word does not exist in dictionary then it does not contribute to the total sum. If a node-string has only one word and this word does not occur in the dictionary as a valid word then it is checked that this word may occur at the start of any dictionary entry. In this case numeric value is also assigned.

After assignment, nodes are ranked according to these values and best *k* (beam value) nodes are selected. These selected nodes are further ranked using statistical methods discussed below.

3.4 Best word segmentation selection

For selection of the most probable word segmentation sequence word and ligature models are used. For word probabilities the following is used.

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} P(w_1^n)$$

To reduce the complexity of computing, Markov assumption are taken to give bigram and trigram approximations (e.g., see Jurafsky & Martin 2006) as given below.

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} \prod_1^n P(\, w_i | w_{i-1})$$

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S} (\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})$$

Similarly the ligature models are built by taking the assumption that sentences are made up of sequences of ligatures rather than words and space is also a valid ligature. By taking the Markov bigram and trigram assumption for ligature grams we get the following.

$$P(L) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}))$$

$$P(L) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}l_{i-2}))$$

Given the ligatures, e.g. as input from and OCR, we can formulate the decoding problem as the following equation.

$$P(W|L) = \quad \mathrm{argmax}_{w_1^n \in S}P(w_1^n|\ l_1^m)$$

where $w_1^n = w_{1,}w_{2,}w_{3,}w_{4,\ldots}w_n$ and $l_1^m = l_{1,}l_{2,}l_{3,}l_{4,\ldots}l_m$; n represents number of words and m represents the number of ligatures. This equation also represents that m number of ligatures can be assigned to n number of words. By applying the Bayesian theorem we get the following derivation.

$$P(W|L) = \quad \mathrm{argmax}_{w_1^n \in S}\frac{P(l_1^m|w_1^n).P(w_1^n)}{P(l_1^m)}$$

As P (l_1^m) is same for all w_1^n , so the denominator does not change the equation, simplifying to the following expression.

$$P(W|L) = \mathrm{argmax}_{w_1^n \in S}P(l_1^m|w_1^n).\ P(w_1^n)$$

where

$$P(l_1^m|w_1^n) = P\ (l_1l_2l_{3,} \ldots l_m|w_1^n)$$
$$= P(l_1|w_1^n) * P\ (l_2|w_1^n l_1) * \ P(l_3|w_1^n l_1 l_2) * P(l_4|w_1^n l_1 l_2 l_3) * \ldots\ P(l_m|w_1^n l_1 l_2 l_3 \ldots l_{m-1})$$

Assuming that a ligature l_i depends only on the word sequence w_1^n and its previous ligature l_{i-1}, and not the ligature history, the above equation can be simplifed as follows.

$$P(l_1^m|w_1^n) = P\ (l_1|w_1^n) * P\ (l_2|w_1^n l_1) * \ P(l_3|w_1^n l_2) * P(l_4|w_1^n l_3) * \ldots\ P(l_m|w_1^n l_{m-1})$$
$$= \prod_1^m P\ (l_i|w_1^n l_{i-1})$$

Further, if it is assumed that l_i depends on the word in which it appears, not whole word sequence, the equation can be further simplified to the following (as probability of l_i within a word is 1).

$$P(l_1^m|w_1^n) = \prod_1^m P\ (l_i|l_{i-1})$$

Thus, considering bigrams, $P(W|L) =$

$$\mathrm{argmax}_{w_1^n \in S}\left(\prod_1^m(P\ (l_i|l_{i-1})\right)(\prod_{k=1}^n P(w_k|w_{k-1}))$$

This gives the maximum probable word sequence among all the alternative word sequences. The precision of the equation can be taken at bigram or trigram level for both ligature and word, giving the following possibilities. Additionally, normalization is also done to better compare different sequences, as each sequences has different number of words and ligatures per word.

- Ligature trigram and word bigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}l_{i-2})) * (\prod_{k=1}^n P(w_k|w_{k-1}))$$

- Ligature bigram and word trigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1})) * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))$$

- Ligature trigram and word trigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}l_{i-2})) * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))$$

- Normalized ligature bigram and word bigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1})\ ^{1/NW}$$

- Normalized ligature trigram and word bigram based technique

$$P(W) = \ \mathrm{argmax}_{w_1^n \in S}\left((\prod_1^m(P\ (l_i|l_{i-1}l_{i-2}))^{1/NL}\right) * (\prod_{k=1}^n P(w_k|w_{k-1}))\ ^{1/NW}$$

- Normalized ligature bigram and word trigram based technique

$$P(W) = \quad \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2})\ ^{1/NW}$$

- Normalized ligature trigram and word trigram based technique

$$P(W) = \ \mathrm{argmax}_{w_1^n \in S}(\prod_1^m(P\ (l_i|l_{i-1}l_{i-2}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2})\ ^{1/NW}$$

In the current work, all the above techniques are used and the best sequence from each one is shortlisted. Then the word sequence which occurs the most times in this shortlist is finally selected.

NL represents the number of ligature bigrams or trigrams and NW represents the number of word bigram or trigrams that exist in the given sentence.

4 Results and Discussion

The model is tested on a corpus of 150 sentences composed of 2156 words and 6075 ligatures. In these sentences, 62 words are unknown, i.e. the words that do not exist in our dictionary. The average length of the sentence is 14 words and 40.5 ligatures. The average length of word is 2.81 ligatures. All the techniques are tested with a beam value, *k*, of 10, 20, 30, 40, and 50.

The results can be viewed from two perspectives: sentence identification rate, and word identification rate. A sentence is considered incorrect even if one word of the sentence is identified wrongly. The technique gives the sentence identification rate of 76% at the beam value of 30. At word level, Normalized Ligature Trigram Word Trigram Technique outperforms other techniques and gives a 96.10% word identification rate at the beam value of 50.

The normalized data gives much better prediction compared to the un-normalized data.

Sentence identification errors depend heavily on the unknown words. For example, at the beam value of 30 we predict 38 incorrect sentences, of which 25 sentence level errors are due to unknown-words and 13 errors are due to known word identification errors. Thus improving system vocabulary will have significant impact on accuracy.

Many of the word errors are caused due to insufficient cleaning of word the larger corpus. Though the words with frequency greater than 50 from the 18 million word corpus have been cleaned, the lower frequency words cause these errors. For example word list still contains "بنیادپر"(*bunyad per*, "depends on"), "سےتقسیم " (*se taqseem*, "divided by") with frequency of 40 and 5 respectively, and each should be two words with a space between them. If low frequency words are also cleaned results will further improve, though it would take a lot of manual effort.

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	110/150	73.33%	2060/2156	95.55%	2024/2092	96.75%	36/64	56.25%
20	112/150	74.67%	2066/2156	95.83%	2027/2092	96.89%	39/64	60.94%
30	114/150	76%	2062/2156	95.64%	2019/2083	96.93%	43/73	58.90%
40	105/150	70%	2037/2156	94.48%	2000/2092	95.60%	37/64	57.81%
50	106/150	70.67%	2040/2156	94.62%	2000/2092	95.60%	40/64	62.50%

Table 3. Results changing beam width *k* of the tree

Technique	Total sentences identified	%age	Total words identified	%age	Total known words Identified	%age	Total unknown words identified	%age
Ligature Bigram	50/150	33.33%	1835/2156	85.11%	1806/2092	86.33%	29/64	45.31%
Ligature Bigram Word Bigram	68/150	45.33%	1900/2156	88.13%	1865/2092	89.15%	35/64	54.69%
Ligature Bigram Word Trigram	83/150	55.33%	1960/2156	90.91%	1924/2092	91.97%	36/64	56.25%
Ligature Trigram	16/150	10.67%	1637/2156	75.93%	1610/2092	76.96%	27/64	42.19%
Ligature Trigram Word Bigram	42/150	28%	1776/2156	82.38%	1746/2092	83.46%	30/64	46.88%
Ligature Trigram Word Trigram	62/150	41.33%	1868/2156	86.64%	1835/2092	87.72%	33/64	51.56%
Normalized Ligature Bigram Word Bigram	90/150	60%	2067/2156	95.87%	2024/2092	96.75%	43/64	67.19%
Normalized Ligature Bigram Word Trigram	100/150	66.67%	2070/2156	96.01%	2028/2092	96.94%	42/64	65.63%
Normalized Ligature Trigram Word Bigram	93/150	62%	2071/2156	96.06%	2030/2092	97.04%	41/64	64.06%
Normalized Ligature Trigram Word Trigram	101/150	67.33%	2072/2156	96.10%	2030/2092	97.04%	42/64	65.63%
Word Bigram	47/150	31.33%	1827/2156	84.74%	1796/2092	85.85%	31/64	48.44%
Word Trigram	74/150	49.33%	1937/2156	89.84%	1903/2092	90.97%	34/64	53.13%

Table 4. Results for all techniques for the beam value of 50

Errors are also caused if an alternate ligature sequence exists. For example the proper noun "کارتک" (*kartak*) is not identifiec as it does not exist in dictionary, but the alternate two word sequence "کار تک" (*kar tak*, “till the car”) is valid.

This work uses the knowledge of ligature grams and word grams. It can be further enhanced by using the character grams. We have tried to clean the corpus. Further cleaning and additional corpus will improve the results as well. Improvement can also be achieved by handling abbreviations and English words transliterated in the text. The unknown word detection rate can be increased by applying POS tagging to further help rank the multiple possible sentences.

5 Conclusions

This work presents an initial effort on statistical solution of word segmentation, especially for Urdu OCR systems. This work develops a cleaned corpus of half a million Urdu words for statistical training of ligature based data, which is now available for the research community. In addition, the work develops a statistical model for word segmentation using ligature and word statistics. Using ligature statistics improves upon using just the word statistics. Further normalization has significant impact on accuracy.

References

Aroonmanakun, W. (2002). *Collocation and Thai Word Segmentation.* In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, (pp. 68-75). Pathumthani.

Chang, Jyun-Shen, Chen, S.-D., Zhen, Y., Liu, X.-Z., & Ke, S.-J. (1992). *Large-corpus-based methods for Chinese personal name recognition.* Journal of Chinese Information Processing , 6 (3), 7-15.

Chen, F., & Goodman, T. (1996). *An Empirical Study of Smoothing Techniques for Language Modeling.* In the Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, (pp. 310-318).

Church, K. W., & Gale, W. A. (1991). *A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams*. Computer Speech and Language , 5, 19-54.

Dai, J.-C., & Lee, H.-J. (1994). *Paring with Tag Information in a probabilistic generalized LR parser*. International Conference on Chinese Computing. Singapore.

Durrani, N. (2007). *Typology of word and automatic word Segmentation in Urdu text corpus*. Thesis, National University of Computer & Emerging Sciences, Lahore , Pakistan.

Durrani, N., Hussain, S. (2010). *Urdu Word Segmentation*, In the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), Los Angeles, US.

Haizhou, L., & Baosheng, Y. (1998). *Chinese Word Segmentation*. In Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation, (pp. 212-217).

Haruechaiyasak, C., Kongyoung, S., & Dailey, M. N. (2008). *A Comparative Study on Thai Word Segmentation Approaches*.

Hussain, S. (2008). *Resources for Urdu Language Processing*. In Proceedings of the Sixth Workshop on Asian Language Resources.

Ijaz, M., Hussain, S. (2007). *Corpus Based Urdu Lexicon Development,* in the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan.

Jurafsky, D., & Martin., l. J(2000). *Speech and Language Processing: An Introduction to Natural Language Processing (1st ed.)*. Computational Linguistics and Speech Recognition Prentice Hall.

Khankasikam, K., & Muansuwan, N. (n.d.). Thai Word Segmentation a Lexical Semantic Approach.

Khankasikam, K., & Muansuwan, N. (2005). *Thai Word Segmentation a Lexical Semantic Approach*. In the Proceedings of the Tenth Machine Translation Summit, (pp. 331-338). Thailand.

Lehal, G. (2009). *A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script.* World Academy of Science, Engineering and Technology 60.

MacKay, D. J., & Peto, L. C. (1995). *A Hierarchical Dirichlet Language Mode*. Natural Language Engineering , 1 (3), 1-19.

Mukund, S. & Srihari, R. (2009). *NE Tagging for Urdu based on Bootstrap POS Learning*. In the Proceedings of CLIAWS3, Third International Cross Lingual Information Access Workshop, pages 61–69, Boulder, Colorado, USA.

Naseem, T., & Hussain, S. (2007). *Spelling Error Trends in Urdu,* In the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan.

Pascale, F., & Dekai, W. (1994). *Statistical augmentation of a Chinese machine readable dictionary.* Proceedings of the Second Annual Workshop on Very Large Corpora, (pp. 69-85).

Poowarawan, Y. (1986). *Dictionary-based Thai Syllable Separation*. Proceedings of the Ninth Electronics Engineering Conference.

Richard Sproat, C. S. (1996). *A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*. Computational Linguistics , 22 (3).

Sajjad, H. (2007). *Statistical Part-of-Speech for Urdu*. MS thesis, National University of Computer and Emerging Sciencies, Centre for Research in Urdu Language Processing, Lahore , Pakistan.

Sornlertlamvanich, V., Potipiti, T., & charoenporn, T. (2000). *Automatic Corpus-Based Thai Word Algorithm Extraction with the C4.5 Learning*. Proceedings of the 18th conference on Computational linguistics.

Sproat, R., Shih, C., Gale, W., & Chang, N. (1996). *A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*. Computational Linguistics , 22 (3).

Theeramunkong, T., & Usanavasin, S. (2001). *Non-Dictionary-Based Thai Word Segmentation Using Decision Trees*. Proceedings of the first international conference on Human language technology research.

Urdu-Nepali-English Parallel Corpus. (n.d.). Retrieved from Center for Research in Urdu Language Processing: *http://www.crulp.org/software/ling_resources/urdunepalienglishparallelcorpus.htm*

Wang, X.-J., Liu, W., & Qin, Y. (2007). A Search-based Chinese Word Segmentation Method. 16th International World Wide Web Conference.

Wong, P.-k., & Chan, C. (1996). *Chinese Word Segmentation based on Maximum Matching and Word Binding Force*. In Proceedings of the 16th conference on Computational linguistics.

Dzongkha Word Segmentation

Sithar Norbu, Pema Choejey, Tenzin Dendup
Research Division
Department of Information Technology & Telecom
{snorbu, pchoejay, tdendup}@dit.gov.bt

Sarmad Hussain, Ahmed Mauz
Center for Research in Urdu Language Processing
National University of Computer & Emerging Sciences
{sarmad.hussain, ahmed.mauz}@nu.edu.pk

Abstract

Dzongkha, the national language of Bhutan, is continuous in written form and it fails to mark the word boundary. Dzongkha word segmentation is one of the fundamental problems and a prerequisite that needs to be solved before more advanced Dzongkha text processing and other natural language processing tools can be developed. This paper presents our initial attempt at segmenting Dzongkha sentences into words. The paper describes the implementation of Maximal Matching (Dictionary based Approach) followed by bigram techniques (Non-dictionary based Approach) in segmenting the Dzongkha scripts. Although the used techniques are basic and naive, it provides a baseline of the Dzongkha word segmentation task. Preliminary experimental results show percentage of segmentation accuracy. However, the segmentation accuracy is dependent on the type of document domain and size and quality of the lexicon and the corpus. Some of the related issues for future directions are also discussed.

Keywords: Dzongkha script, word segmentation, maximal matching, bigram technique, smoothing technique.

1 Introduction

Segmentation of a sentence into word is one of the necessary preprocessing tasks and is essential in the analysis of natural language processing. This is because word is both syntactically and semantically, the fundamental unit for analyzing language structure. Like in any other language processing task, Dzongkha word segmentation is also viewed as one of the fundamental and foremost steps in Dzongkha related language processing tasks.

The most challenging features of Dzongkha script is the lack of word boundary separation between the words[1]. So, in order to do the further linguistic and natural language processing tasks, the scripts should be transformed into a chain of words. Therefore, segmenting a word is an essential role in Natural Language Processing. Like Chinese, Japanese and Korean (CJK) languages, Dzongkha script being written continuously without any word delimiter causes a major problem in natural language processing tasks. But, in case of CJK, Thai, and Vietnamese languages, many solutions have been published before. For Dzongkha, this is the first ever word segmentation solution to be documented.

In this paper, we describe the Dzongkha word segmentation, which is performed firstly using the Dictionary based approach where the principle of maximal matching algorithm is applied to the input text. Here, given the collection of lexicon, the maximal matching algorithm selects the segmentation that yields the minimum number of words token from all possible segmentations of the input sentence. Then, it uses non-dictionary based approach where bigram technique is applied. The probabilistic model of a word sequence is

[1] http://www.learntibetan.net/grammar/sentence.htm

Proceedings of the 8th Workshop on Asian Language Resources, pages 95–102,
Beijing, China, 21-22 August 2010.

studied using the Maximum Likelihood Estimation (MLE). The approach using the MLE has an obvious disadvantage because of the unavoidably limited size of the training corpora (Nuges, 2006). To this problem of data sparseness, the idea of Katz back-off model with Good-Turing smoothing technique is applied.

2 Dzongkha Script

Dzongkha language is the official and national language of Bhutan. It is spoken as the first language by approximately 130,000 people and as the second language by about 470,000 people (Van Driem and Tshering, 1998).

Dzongkha is very much related to Sino-Tibetan language which is a member of Tibeto-Burmese language family. It is an alphabetic language, with phonetic characteristics that mirror those of Sanskrit. Like many of the alphabets of India and South East Asia, the Bhutanese script called Dzongkha script is also a syllabic[2]. A syllable can contain as little as one character or as many as six characters. And a word can be of one syllable, two syllable or multiple syllables. In the written form, Dzongkha script contains a dot, called Tsheg (་) that serve as syllable and phrase delimiter, but words are not separated at all.

For example,

Dzongkha	Transliteration	English	Syllables
དམརཔོ	dmarpo	red	Single-syllabled
སློབ་དཔོན	slop-pon	Teacher	Two-syllabled
འཇམ་ཏོག་ཏོ	hjam-tog-to	easy	Three-syllabled
འར་རི་ཧུར་རི	har-ri-hur-ri	crowdedness /confusion	Four-syllabled

Table 1: Different syllabled Dzongkha scripts.

The sentence is terminated with a vertical stroke called Shad (།). This Shad acts as a full_stop. The frequent appearance of whitespace in the Dzongkha sentence serves as a phrase boundary or comma, and is a faithful representation of speech: after all in speech, we pause not between words, but either after certain phrases or at the end of sentence.

The sample dzongkha sentence reads as follows:

རྫོང་ཁ་གོང་འཕེལ་ལྷན་ཚོགས་འདི་ འབྲུག་རྒྱལ་ཁབ་ནང་ གཞུང་གི་ཁ་ཐུག་ལས་ འབྲུག་གི་རྒྱལ་ཡོངས་སྐད་ཡིག་ རྫོང་ཁའི་སྲིད་བྱུས་རྩམ་མི་དང་ རྫོང་ཁའི་མཐར་ཐུག་གི་དབང་འཛིན་པ་ རང་དབང་རང་སྐྱོང་གི་འདུས་ཚོགས་ མཐོ་ཤོས་ཅིག་ཨིན། འདུས་ཚོགས་འདི་ འབྲུག་རྒྱལ་བཞི་པ་མི་དབང་མངའ་བདག་རིན་པོ་ཆེ་ དཔལ་འཇིགས་མེད་སེང་གེ་དབང་ཕྱུག་མཆོག་གི་ཐུགས་དགོངས་དང་འཁྲིལ་ཏེ་ སྤྱི་ལོ་ ༡༩༨༦ ལུ་གཞི་བཙུགས་གནང་གནངམ་ཨིན།

(English Translation of example text)

[The Dzongkha Development Commission is the leading institute in the country for the advancement of Dzongkha, the national language of Bhutan. It is an independent organization established by the Fourth King of Bhutan, His Majesty the King Jigme Singye Wangchuck, in 1986.]

3 Materials and Methods

Since, our language has no word boundary delimiter, the major resource for Dzongkha word segmentation is a collection of lexicon (dictionary). For such languages, dictionaries are needed to segment the running texts. Therefore, the coverage of a dictionary plays a significant role in the accuracy of word segmentation (Pong and Robert, 1994).

The dictionary that we used contains 23,333 word lists/lexicons. The lexicons were collected from "Dzongkha Dictionary", 2nd Edition, Published by Dzongkha Development Authority, Ministry of Education, 2005, (ddc@druknet.bt). The manually segmented text corpus containing 41,739 tokens are also used for the method. The text corpora were collected from different sources like newspaper articles, dictionaries, printed books, etc. and belong to domains such as World Affairs, Social Sciences, Arts, Literatures, Adventures, Culture and History. Some texts like poetry and songs were added manually.

[2] http://omniglot.com/writing/tibetan.htm

Table below gives the glimpse of textual domains contained in the text corpora used for the method (Chungku et al., 2010).

Domain	Sub domain	(%)
World Affairs	Bilateral relations	12%
Social Science	Political Science	2%
Arts	Poetry/Songs/Ballad	9%
Literatures	Essays/Letters/Dictionary	72%
Adventures	Travel Adventures	1%
Culture	Culture Heritage/Tradition	2%
History	Myths/Architecture	2%

Table 2: Textual domain contained in Corpus

Figure 1 below shows the Dzongkha Word Segmentation Process.

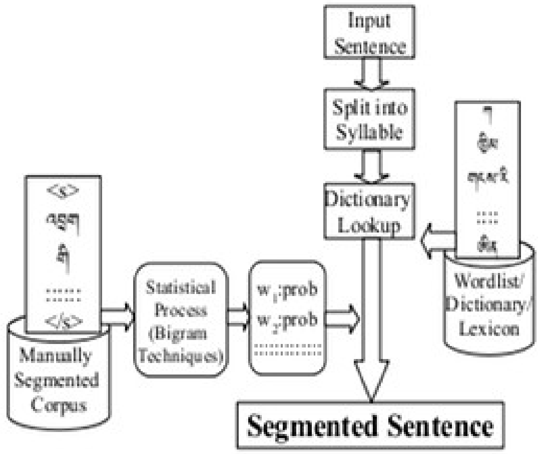

Input Sentence
Split into Syllable
Dictionary Lookup
<s>
</s>
Manually Segmented Corpus
Statistical Process (Bigram Techniques)
w1:prob
w2:prob
Wordlist/ Dictionary/ Lexicon
Segmented Sentence

Figure 1: Dzongkha Word Segmentation Process.

Dzongkha word segmentation implements a principle of maximal matching algorithm followed by statistical (bigram) method. It uses a word list/lexicon at first to segment the raw input sentences. It then uses MLE principles to estimate the bigram probabilities for each segmented words. All possible segmentation of an input sentence by Maximal Matching are then re-ranked and picked the mostly likely segmentation from the set of possible segmentations using a statistical approach (bigram technique). This is to decide the best possible segmentation among all the words (Huor et al., 2007) generated by the maximal matching algorithm. These mechanisms are described in the following

3.1 Maximal Matching Algorithm

The basic idea of Maximal matching algorithm is, it first generates all possible segmentations for an input sentence and then selects the segmentation that contains the minimum number of word tokens. It uses dictionary lookup.

We used the following steps to segment the given input sentence.

1. Read the input of string text. If an input line contains more than one sentence, a sentence separator is applied to break the line into individual sentences.
2. Split input string of text by Tsheg(་)into syllables.
3. Taking the next syllables, generate all possible strings
4. If the number of string is greater than *n* for some value *n*[3]

- Look up the series of string in the dictionary to find matches, and assign some weight-age[4] accordingly.
- Sort the string on the given weight-age
- Delete (number of strings – *n*) low count strings.

5. Repeat from Step 2 until all syllables are processed.

The above mentioned steps produced all possible segmented words from the given input sentence based on the provided lexicon. Thus, the overall accuracy and performance depends on the coverage of lexicon (Pong and Robert, 1994).

[3] The greater the value of n, the better the chances of selecting the sentence with the fewest words from the possible segmentation.

[4] If the possible string is found in the dictionary entries, the number of syllable in the string is counted. Then, the weight-age for the string is calculated as (number of syllable)^2 else it carries the weight-age 0

3.2 Bigram Method

(a) **Maximum Likelihood Estimation**[5]

In the bigram method, we make the approximation that the probability of a word depends on identifying the immediately preceding word. That is, we calculate the probability of next word given the previous word, as follows:

$$P\left(w_1^n\right)=\Pi_{i=1}^{n} P\left(w_i/w_{i-1}\right)$$

where

- $P\left(w_i/w_{i-1}\right)=\dfrac{count\left(w_{i-1}w_i\right)}{count\left(w_{i-1}\right)}$

where

- $count\left(w_{i-1}w_i\right)$ is a total occurrence of a word sequence $w_{i-1}w_i$ in the corpus, and
- $count\left(w_{i-1}\right)$ is a total occurrence of a word w_{i-1} in the corpus.

To make $P\left(w_i/w_{i-1}\right)$ meaningful for $i=1$, we use the distinguished token <s> at the beginning of the sentence; that is, we pretend w_0 = <s>. In addition, to make the sum of the probabilities of all strings equal 1, it is necessary to place a distinguished token </s> at the end of the sentence.

One of the key problems with the MLE is insufficient data. That is, because of the unavoidably limited size of the training corpus, vast majority of the word are uncommon and some of the bigrams may not occur at all in the corpus, leading to zero probabilities.

Therefore, following smoothing techniques were used to count the probabilities of unseen bigram.

(b) **Smoothing Bigram Probabilistic**

The above problem of data sparseness underestimates the probability of some of the sentences that are in the test set. The smoothing technique helps to prevent errors by making the probabilities more uniform. Smoothing is the process of flattening a probability distribution implied by a language model so that all reasonable word sequences can occur with some probability. This often involves adjusting zero probabilities upward and high probabilities downwards. This way, smoothing technique not only helps prevent zero probabilities but the overall accuracy of the model are also significantly improved (Chen and Goodman, 1998).

In Dzongkha word segmentation, Katz back-off model based on Good-Turing smoothing principle is applied to handle the issue of data sparseness. The basic idea of Katz back-off model is to use the frequency of n-grams and if no n-grams are available, to back off to *(n-1)* grams, and then to *(n-2)* grams and so on (Chen and Goodman, 1998).

The summarized procedure of Katz smoothing technique is given by the following algorithm:[6]

$$P_{katz}\left(w_i|w_{i-1}\right)=\begin{Bmatrix} C\left(w_{i-1}/w_i\right) & if r>k \\ d_r C\left(w_{i-1}/w_i\right) & if k\geq r>0 \\ \alpha\left(w_{i-1}\right)P\left(w_i\right) & if r=0 \end{Bmatrix}$$

where

- r is the frequency of bigram counts
- k is taken for some value in the range of 5 to 10, other counts are not re-estimated.
- $d_r = \dfrac{\dfrac{r^*}{r}-\left(k+1\right)\dfrac{n_{K+1}}{n_1}}{1-\dfrac{\left(k+1\right)n_{k+1}}{n_1}}$
- $\alpha\left(w_{i-1}\right) = \dfrac{1-\sum_{w_i:r>0} P_{Katz}\left(w_i|w_{i-1}\right)}{1-\sum_{w_i:r>0} P_{Katz}\left(w_i\right)}$

With the above equations, bigrams with non-zero count *r* are discounted according to the

[5] P.M, Nugues. An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with Special Consideration of English, French, and German (Cognitive Technologies) (95 – 104).

[6] X. Huang, A. Acero, H.-W.Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, (Prentice-Hall Inc., New Jersey 07458, 2001), 559 - 561.

discount ratio $d_r = \frac{r^*}{r}$ i.e., the count subtracted from the non-zero count are redistributed among the zero count bigrams according to the next lower-order distribution, the unigram model.

4 Evaluations and Results

Subjective evaluation has been performed by comparing the experimental results with the manually segmented tokens. The method was evaluated using different sets of test documents from various domains consisting of 714 manually segmented words. Table 3 summarizes the evaluation results.

Document text	Correct Detect (Correctly segmented tokens / total no. of words)	Accuracy
Astrology.txt	102/116	87.9%
dzo_linux.txt	85/93	91.4%
movie_awards.txt	76/84	90.5%
News.txt	78/85	91.8%
Notice.txt	83/92	90.2%
Religious.txt	63/73	89.0%
Song.txt	57/60	95.0%
Tradition.txt	109/111	98.2%
Total	**653/714**	**91.5%**

Table 3: Evaluation Results

Accuracy in %age are measured as:

$$\text{Accuracy(\%)} = \frac{N}{T} * 100$$

where

- N is the number of correctly segmented tokens
- T is the total number of manually segmented tokens/ Total number of words.

We have taken the extract of different test data hoping it may contain fair amount of general terms, technical terms and common nouns. The manually segmented corpus containing 41,739 tokens are used for the method.

In the sample comparison below, the symbol (་) does not make the segmentation unit's mark, but (।) takes the segmentation unit's mark, despite its actual mark for comma or full_stop. The whitespace in the sentence are phrase boundary or comma, and is a faithful representation of speech where we pause not between words, but either after certain phrases or at the end of sentence.

Consider the sample input sentence:

རྫོང་ཁ་ལི་ནགསི་འདི་ རྫོང་ཁ་གློག་རིག་ནང་བཙུགས་ནིའི་དོན་ལུ་ རྒྱབ་སྐྱོར་ཧྲིལ་བུ་གཅིག་ཁར་བསྡོམས་མི་ རང་དབང་ལི་ནགསི་ བཀོལ་སྤྱོད་རིམ་ལུགས་འདི་གི་ ཉེ་གནས་སྦྲེ་མཐུན་འགྱུར་བཟོ་ཡོད་པའི་ཐོན་རིམ་ཅིག་ཨིན། དེ་གིས་ ཆ་ཚང་སྦྲེ་སྐད་བསྒྱུར་འབད་ཡོད་པའི་ལག་ལེན་པའི་ངོས་འདྲ་བ་ཚུ་སྟོནམ་ཨིན།

Manually segmented sentence of the sample input sentence:

རྫོང་ཁ།ལི་ནགསི།འདི། རྫོང་ཁ།གློག་རིག།ནང།བཙུགས།ནིའི།དོན་ལུ། རྒྱབ་སྐྱོར།ཧྲིལ་བུ།གཅིག།ཁར།བསྡོམས།མི། རང་དབང།ལི་ནགསི། བཀོལ་སྤྱོད།རིམ་ལུགས།འདི།གི། ཉེ་གནས།སྦྲེ།མཐུན་འགྱུར།བཟོ།ཡོད།པའི། ཐོན།རིམ།ཅིག།ཨིན། དེ་གིས། ཆ་ཚང།སྦྲེ།སྐད་བསྒྱུར།འབད།ཡོད།པའི། ལག་ལེན།པའི།ངོས།འདྲ་བ།ཚུ།སྟོནམ།ཨིན།

Using maximal matching algorithm:

རྫོང་ཁ། ལི། ནགསི། འདི། རྫོང་ཁ། གློག་རིག། ནང། བཙུགས། ནིའི། དོན། ལུ། རྒྱབ་སྐྱོར། ཧྲིལ། བུ། གཅིག། ཁར། བསྡོམས། མི། རང་དབང། ལི། ནགསི། བཀོལ་སྤྱོད། རིམ་ལུགས། འདི་གི། ཉེ་གནས། སྦྲེ། མཐུན་འགྱུར། བཟོ། ཡོད། པའི། ཐོན། རིམ། ཅིག་ཨིན། དེ། གིས། ཆ་ཚང། སྦྲེ། སྐད་བསྒྱུར། འབད། ཡོད། པའི། ལག་ལེན། པའི། ངོས། འདྲ། བ། ཚུ། སྟོནམ། ཨིན།

System segmented version of the sample input sentence: Underlined text shows the incorrect segmentation.

རྫོང་ཁ། ལི་ནགསི་འདི། རྫོང་ཁ།གློག་རིག།ནང།བཙུགས། ནིའི་དོན་ལུ། རྒྱབ་སྐྱོར།ཧྲིལ་བུ།གཅིག།ཁར།བསྡོམས།མི། རང་དབང།ལི་ནགསི།བཀོལ་སྤྱོད།རིམ་ལུགས།འདི།གི། ཉེ་གནས།སྦྲེ།མཐུན་འགྱུར།བཟོ།ཡོད།པའི།

ཐོན|རིམ|ཅིག|ཨིན། དེ|གིས|ཆ་ཚང|སྦེ|སྐད་བསྒྱུར|འབད|ཡོད|པའི|
ལག་ལེན|པའི|ངོས|འདྲ་བ|རྩོ|སྟོནམ| ཨིན།

5 Discussions

During the process of word segmentation, it is understood that the maximal matching algorithm is simply effective and can produce accurate segmentation only if all the words are present in the lexicon. But since not all the word entry can be found in lexicon database in real application, the performance of word segmentation degrades when it encounters words that are not in the lexicon (Chiang et al., 1992).

Following are the significant problems with the dictionary-based maximal matching method because of the coverage of lexicon (Emerson, 2000):

- incomplete and inconsistency of the lexicon database
- absence of technical domains in the lexicon
- transliterated foreign names
- some of the common nouns not included in the lexicon
- lexicon/word lists do not contains genitive endings པའི (expresses the genitive relationship as a quality or characteristic of the second element, for example, དབུལ་པའི་བུ 'son of a pauper') and འི (first singular possessive, for example, ངེའི་བུམོ which actually is ངེ་གི་བུམོ 'my daughter') that indicates possession or a part-to-whole relationship, like English 'of'.

A Dzongkha sentence like:
འདི་རྫོང་ཁ་གི་ ཞིབ་འཚོལ་ཡིག་ཆ་ ཨིན།
may include the following ambiguous possible segmentation based on simple dictionary lookup:

1.འདི|རྫོང་ཁ|གི|ཞིབ་འཚོལ|ཡིག་ཆ|ཨིན
this | Dzongkha | of | research | written document | is

2.འདི|རྫོང་ཁ|གི|ཞིབ|འཚོལ|ཡིག་ཆ|ཨིན
this | Dzongkha | of | arrange together | search/ expose | written document | is

3.འདི|རྫོང|ཁ|གི|ཞིབ་འཚོལ|ཡིག་ཆ|ཨིན
this | fortress | mouth/surface | of | research | written document | is

These problems of ambiguous word divisions, unknown proper names, are lessened and solved partially when it is re-ranked using the bigram techniques. Still the solution to the following issues needs to be discussed in the future. Although the texts were collected from widest range of domains possible, the lack of available electronic resources of informative text adds to the following issues:

- small number of corpus were not very impressive for the method
- ambiguity and inconsistent of manual segmentation of a token in the corpus resulting in incompatibility and sometimes in conflict.

Ambiguity and inconsistency occurs because of difficulties in identifying a word. Since the manual segmentation of corpus entry was carried out by humans rather than computer, such humans have to be well skilled in identifying or understanding what a word is.

The problem with the Dzongkha scripts that also hampers the accuracy of dzongkha word segmentation includes the issues such as ambiguous use of *Tsheg* (་) in different documents. There are two different types of *Tsheg:* Unicode 0F0B (་) called *Tibetan mark inter syllabic tsheg* is a normal *tsheg* that provides a break opportunity. Unicode 0F0C (༌) called *Tibetan Mark Delimiter Tsheg Bstar* is a non-breaking *tsheg* and it inhibits line breaking.
For example,
input sentence with Tsheg 0F0B:
སངས་རྒྱས་དང་ཚེ་རིང་གཉིས་ བརྟ་དོན་དང་འཕྲུལ་རིག་ནང་ ལྷ་འབད་
དོ་ཡོདཔ་ཨིན་པས།
achieves 100% segmentation as follow:

སངས་རྒྱས། དང། ཆོ་རིང། གཉིས། བརྡ་དོན། དང། འཕྲུལ། རིག། ནང། ལཱ། འབད། དོ། ཡོདཔ། ཨིན། པས།

whereas the same input sentence with Tsheg 0F0C is incorrectly segmented as follows:

སངས་རྒྱས་དང་ཆོ་རིང་གཉིས་། བརྡ་དོན་དང་འཕྲུལ་རིག་ནང་། ལཱ་འབད་དོ་ཡོདཔ་ཨིན་པས།

There are also cases like shortening of words, removing of inflectional words and abbreviating of words for the convenience of the writer. But this is not so reflected in the dictionaries, thus affecting the accuracy of the segmentation.

Following words has a special abbreviated way of writing a letter or sequence of letters at the end of a syllable as

རྡོ་རྗེ as རྡོེ

ཡེ་ཤེས as ཡཻས

etc..

6 Conclusion and Future works

This paper describes the initial effort in segmenting the Dzongkha scripts. In this preliminary analysis of Dzongkha word segmentation, the preprocessing and normalizations are not dealt with. Numberings, special symbols and characters are also not included. These issues will have to be studied in the future. A lot of discussions and works also have to be done to improve the performance of word segmentation. Although the study was a success, there are still some obvious limitations, such as its dependency on dictionaries/lexicon, and the current Dzongkha lexicon is not comprehensive. Also, there is absence of large corpus collection from various domains. Future work may include overall improvement of the method for better efficiency, effectiveness and functionality, by exploring different algorithms. Furthermore, the inclusion of POS Tag sets applied on n-gram techniques which is proven to be helpful in handling the unknown word problems might enhance the performance and accuracy. Increasing corpus size might also help to improve the results.

Acknowledgment

This research work was carried out as a part of PAN Localization Project (http://www.PANL10n.net) with the aid of a grant from the International Development Research Centre (IDRC), Ottawa, Canada, administered through the Center of Research in Urdu Language Processing (CRULP), National University of Computer and Emerging Sciences (NUCES), Pakistan. The research team would also like to express the gratitude to all the PAN Localization Project members of Bhutanese team based at Department of Information Technology and Telecom, for their efforts in collecting, preparing and providing with the lexicon, corpus, useful training and testing materials and finally for the their valuable support and contribution that made this research successful.

References

Chen, Stanley F., Joshua Goodman, 1998. *An Empirical Study of Smoothing Techniques for Language Modeling*, Computer Science Group, Harvard University, Cambridge, Massachusetts

Chiang, T-Hui., J-Shin Chang,, M-Yu Lin, K-Yih Su, 2007. *Statistical models for word segmentation and unknown word resolution.* Department of Electrical Engineering , National Tsing Hua University, Hsinchu, Taiwan.

Chungku., Jurmey Rabgay, Gertrud Faaβ, 2010. *NLP Resources for Dzongkha.* Department of Information Technology & Telecom, Ministry of Information & Communications, Thimphu, Bhutan.

Durrani, Nadir and Sarmad Hussain, 2010. *Urdu Word Segmentation.* Human Language Technologies: 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, June 2010.

Emerson, Thomas. 2000. *Segmenting Chinese in Unicode.* 16th International Unicode conference, Amsterdam, The Netherlands, March 2000

Haizhou, Li and Yuan Baosheng, 1998. *Chinese Word Segmentation.* Language, Information and Computation (PACLIC12), 1998.

Haruechaiyasak, C., S Kongyoung, M.N. Dailey, 2008. *A Comparative Study on Thai Word*

Segmentation Approaches. In Proceedings of ECTI-CON, 2008.

Huang, X., A. Acero, H.-W. Hon, 2001. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development (pp. 539 – 578).* Prentice-Hall Inc., New Jersey 07458.

Huor, C.S., T. Rithy, R.P. Hemy, V. Navy, C. Chanthirith, C. Tola, 2007. *Word Bigram Vs Orthographic Syllable Bigram in Khmer Word Segmentation.* PAN Localization Working Papers 2004 - 2007. PAN Localization Project, National University of Computer and Emerging Sciences, Lahore, Pakistan.

Jurafsky, D., A. Acero, H.-W. Hon, 1999. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (pp. 189 – 230).* Prentice-Hall Inc., New Jersey 07458.

Nugues, P.M. 2006. *An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with Special Consideration of English, French, and German (Cognitive Technologies) (pp. 87 – 104).* Springer-Verlag Berlin Heidelberg

Pong, L.W. and Robert. 1994. *Chinese word segmentation based on maximal matching and bigram techniques.* Retrieved from The Association for Computational Linguistics and Chinese Language Processing. On-line: http://www.aclclp.org.tw/rocling/1994/P04.pdf

Sunthi, Thepchai. 2007. *Word Segmentation and POS tagging.* ADD-2 Workshop, SIIT, NECTEC, Thailand.

Van Driem, George. and Karma Tshering, (Collab), *"Languages of Greater Himalayan Region"*, 1998.

Building NLP resources for Dzongkha: A Tagset and A Tagged Corpus

Chungku Chungku, Jurmey Rabgay
Research Division
Department of Information Technology
& Telecom
{chungku,jrabgay}@dit.gov.bt

Gertrud Faaß
Institute für Maschinelle
Sprachverarbeitung(NLP processing),
University of Stuttgart
faasz@ims.uni-stuttgart.de

Abstract

This paper describes the application of probabilistic part of speech taggers to the Dzongkha language. A tag set containing 66 tags is designed, which is based on the Penn Treebank[1]. A training corpus of 40,247 tokens is utilized to train the model. Using the lexicon extracted from the training corpus and lexicon from the available word list, we used two statistical taggers for comparison reasons. The best result achieved was 93.1% accuracy in a 10-fold cross validation on the training set. The winning tagger was thereafter applied to annotate a 570,247 token corpus.

1 Introduction

Dzongkha is the national language of Bhutan. Bhutan has only begun recently applying Natural Language Processing (henceforth NLP) methodologies and tools. However, Dzongkha computing is currently progressing rapidly.

Part of speech (henceforth POS) tagging means annotating each word with their respective POS label according to its definition and context. Such annotation generates a description of the text on a meta-level, i.e. a representation of linguistic units on the basis of their properties. This POS-level provides significant information usable by further linguistic research, may it be of the morphological, syntactic or semantic kind. Producing such enriched data is proven to be useful especially when designing NLP representations of higher levels of representation, e.g. syntactic parses.

Our project is designed to annotate Dzongkha cyclopedia text with parts of speech using a probabilistic tagger. This means that a set of tags is to be developed and applied manually to parts of these texts creating training data. Probabilistic taggers can then be applied to annotate other texts with their parts of speech automatically. In this paper, we make use of two such taggers and report on their results.

At present, our POS tagged data is already in use in projects concerning Dzongkha Text to Speech (TTS) processing, further tests on word segmentation (see current state below) and in corpus-linguistic research. Future work entails its utilization for higher-level NLP tasks such as parsing, building parallel corpora, research on semantics, machine translation, and many more.

Sections 2 and 3 of this paper describe the Dzongkha script and the challenges in Dzongkha, section 4 presents our resources, tagset and corpus. Section 5 describes tagging and validation processes and reports on their results. Section 6 concludes and discusses future work.

2 The Dzongkha Language

Dzongkha is recognized as the national and official language of Bhutan. It is categorized as a Sino-Tibetan Language and said to have derived from the classical Tibetan or choka: Dzongkha consonants, vowels, phonemes, phonetics and writing system are all identical.

[1] [http://www.cis.upenn.edu/~treebank/]

Proceedings of the 8th Workshop on Asian Language Resources, pages 103–110,
Beijing, China, 21-22 August 2010.

From a linguistic perspective, Dzongkha script is syllabic, a syllable can contain one character or as many as six characters. A syllable marker known as “tsheg”, which is simply a superscripted dot, separates the syllables of a word. Linguistic words may contain one or more syllables and are also separated by the same symbol, “tsheg”, thus the language is lacking word boundaries.

Sentences of Dzongkha contain one or more phrases which themselves contain one or more words. A character known as “shed” marks a sentence border, it looks like a vertical pipe.

Phonetic information is available, too: In most sentences, a pause of silence is taken after each phrase while speaking the Dzongkha language. The written form of Dzongkha represents this pause with a space after each phrase in the case that it occurs not at the end of the sentence. The Dzongkha writing system leads to a serious problem: the detection of word borders, because only phrases are separated by a space. POS tagging usually requires a one-token-per line format, which is produced by a process called word segmentation. The tagger then adds the POS category to each token.

The training data of (40247 tokens) was segmented manually to achieve higher accuracy of word boundary and also due to lack of word segmentation during that time. After a Dzongkha word segmentation[2] tool was developed, the remaining text was segmented with this tool, which works basically with a lexicon and the longest string matching method.

3 Challenges and suggestions for tagging Dzongkha texts

3.1 Words unknown to the language model

A statistical tagger learns from POS distributions in manually tagged data while being trained and, when being applied to unknown text, “guesses” the POS of each word. The TreeTagger (Schmid, 1994) additionally makes use of a lexicon externally provided when producing its language model (the “parameter file”). We had opted for using the TreeTagger and hence we have listed about 28,300 Dzongkha words with their POS in a lexicon selected from the 570,247 token corpus to be tagged. We fed these data to the tagger during its training phase. Note, however, that such a lexicon may never be complete, as there are morphologically productive processes creating new forms (these belong to POS classes that are often named “open”). Such forms may be taken into account when developing a tagset, however, in this work, we opted for postponing the issue until a morphological analyser can be developed.

3.2 Ambiguous function words

A number of Dzongkha function words are ambiguous; for each occurrence of such a word, the tagger has to decide on the basis of the word’s contexts, which of the possible tags is to be assigned. Here, the tagset itself comes into view: whenever it is planned to utilize probabilistic POS taggers, the tagset should be designed on the basis of the words’ distributions, otherwise the potential accuracy of the taggers may never be achieved.

In Dzongkha it is mainly the function words that are ambiguous in terms of their POS. A typical example is ལས་/le/(from) belonging to the category PP (post position) and ལས་/le/(so) which is of the category CC (conjunction).

3.3 Fused forms

Some morpho-phonemic processes in Dzongkha lead to the fusing of words, presenting another challenge for tagging. Such words[3] are not very frequent, thus proposing a challenge to statistical taggers. The word རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, is fused from the phrase རྒྱལ་པོ་གི་/gelpo gi/ (king is); another example is the fused form བསེདན་/sen/ ([to] kill), made from བསེད་་བ་ཅིན་/se wa cin/ (if [to] kill).

When a tagset does not cater for fused forms of words, one could split these forms while tokenizing adding an intermediate level of representation between original text level and

[2] This tool was developed at NETEC(National Electronics and Computer Technology Center), Thailand.

[3] In our training set, there were 1.73% of all words detected as fused forms.

the POS level: a level of text material to be utilized for tagging or other further processing, as e.g. done by (Taljard et al., 2008) for the Bantu Language Northern Sotho. However, the forms could not easily be split, as the resulting first parts of the words would not contain the separator “tsheg”. Splitting the word རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, would result in རྒྱལ་པོ་/gelpo/(king) and འི་/yi/$_{[+genitive]}$. The language model does not cater for words ending in characters other than "tsheg" (word border) or being directly followed by "shed" (a word like རྒྱལ་པོ་/gelpo/(king) may only appear if preceding a sentence border). Tagging accuracy for such theoretical forms are not expected to be acceptable. Fusing processes are productive, therefore, further research in the frame of a project developing a Dzongkha tokenizer is deemed necessary.

We examined all fused words contained in our textual data to find an workable solution at the current stage of the project. As long as the problem of tokenizing automatically is not solved, we opted for keeping the fused forms as they are. To enhance our tagger results, we suggest to add a number of tags to our tagset that consist of the two POS tags involved. རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, is tagged as “NN+CG” and བསེདན་/sen/ ([to] kill) as “VB+SC”. The “+” indicates combined individual tags. All known forms are added to a new version of the lexicon. Note, however, that all tagging results reported upon in this paper, are still based on the tag set described below.

4 Resources used

4.1 Tagset

During the first phase of PAN Localization project, the first Dzongkha POS tagset[4] was created. It consisted of 47 tags, its design is based on the Penn Guidelines[5] and its categories of POS correspond to the respective English Penn categories. PAN generally makes use of the Penn Treebank tags as a basis for tagging. Examining the similar features exhibited by both the languages (Dzongkha and English), tags that were applicable to Dzongkha were taken directly from the Penn Treebank. In cases where these languages showed dissimilarities in their nature, new tags for Dzongkha were assigned (based e.g. on the work on Urdu of Sajjad and Schmid, 2009). As an example for such dissimilarity, Dzongkha postpositions are mentioned here, cf. (1); the respective tag (PP) only exist for Dzongkha whereas in English the whole set of ad position tags (preposition and postpositions) exist.

(1) བྱི་ལི་ ཤིང་གི་འོག་ལུ་འདུག

i'ili	shing-gi	wôlu	-dû
Cat	tree$_{[posp]}$	under$_{[PP]}$	be

"A cat is under the tree"

Whenever a tagset designed on theoretical implications is applied to text, it will be found in the course of creating the training data that not all morpho-syntactic phenomena of the language had been considered. This happened for Dzongkha, too: words appeared in the texts that didn't fit in any of the pre-defined classes.

Dzongkha uses honorific forms: ན་བཟའ་/nam za/ (cloths) is the honorific form of the noun གོ་ལ་/gola/(cloths), གསུངས་/sung/(tell) the honorific form of the verb སླབ་/lab/(tell). We opted to mark them by adding the tag NNH (honorific common noun) and VBH (honorific verb) to enable future research on this specific usage of Dzongkha language. A number of tags were added to the set, of which we describe four in more detail: two of the additional tags are sub-classes of verbs: VBH (honorific verb form), and VBN which describes past participle forms, like, e.g. བྱུངམ་/jun/(created), the past particle form of བྱུང་/jung/(create).

Concerning case, we added two subclasses of case: CDt and CA. These differentiate between dative (CDt) and ablative (CA): The CDt (Dative case) labels e.g.དོན་ལས་/doen le/(for it) and དོན་ལུ་/doen lu/(for this). The Ablative case

[4] The original Dzongkha tag set is described at http://www.panl10n.net

[5] The Penn Guidelines can be downloaded from: http://www.cis.upenn.edu/~treebank/

(CA) is used when the argument of the preposition describes a source. For example, in the phrase ཤིང་ལས་ཀང་ཁྲི་/shing le kang thri/(from wood chair), ལས་/le/from/ will be labeled CA since the chair described is made from (the source) wood (Muaz, et al. 2009). The tagset utilized in our experiment consists of a total of 66 parts of speech as shown in Appendix (A).

4.2 Collecting a Corpus and generating a training data set

The Corpus collection process. The process of collecting a corpus should be based on its purpose. As our goal was the design a Dzongkha text corpus as balanced as possible in terms of its linguistic diversity, the text data was gathered from different sources like newspaper articles, samples from traditional books, and dictionaries, some text was added manually (poetry and songs). The text selection was also processed with a view on the widest range of genres possible: texts from social science, arts and culture, and texts describing world affairs, travel adventure, fiction and history books were added as our goal is to make it representative of every linguistic phenomena of language (Sarkar, et al. 2007). The corpus is however not balanced for a lack of available electronic resources of informative text (so far only 14% belong to this category). Future work will therefore entail collecting more data from respective websites and newspapers.
The entire corpus contains 570,247 tokens; it made from the domains described in table (1).

Domain	Share %	Text type
1) World Affairs	12%	Informative
2) Social Science	2%	Informative
3) Arts	9%	Descriptive
4) Literature	72%	Expository
5) Adventure	1%	Narrative
6) Culture	2%	Narrative
7) History	2%	Descriptive

Table (1): Textual domains contained in the corpus

The Training data set

Cleaning of texts. Raw text is usually to be modified before it can be offered to a tagger. It is to be cleaned manually, e.g. by removing extra blank spaces, inserting missing blanks, correcting spelling mistakes, and by removing duplicate occurrences of sequences. Secondly, the process of tokenization ("Word Segmentation") is to be applied.

Design and generation of training data. The training data set was produced in several steps: Firstly, 20,000 tokens were manually labeled with their respective parts of speech (for a comparison of tagging techniques, cf. Hasan et al., 2007). Thereafter, the problems that had occurred during the manual process were summarized and the tagset revised as described in section 4.1. Thereafter, we added another 20,247 tokens. The final training data set hence consists of 40,247 tokens (2,742 sentences, 36,362 words, 3,265 punctuation, 650 numbers).

4.3 Tagging technique: TreeTagger and TnT

TreeTagger (Schmid, 1994): TreeTagger (Schmid, 1994) is a probabilistic part of speech tagger operating on the basis of decision trees. Helmut Schmid developed it in the frame of the "TC"[6] project at the Institute for Computational Linguistics at the University of Stuttgart, Germany.

The software consists of two modules

a) train-tree-tagger: utilized to generate a parameter file from a lexicon and a hand-tagged corpus.

b) tree-tagger: makes use of the parameter file generated with a); annotates text (which is to be tokenized first) with part-of-speech automatically.

a) Generating a language model: Training

When generating a language model stored in a so-called "parameter file", three files are required: a lexicon describing tokens and their respective tags, a list of open tags, and training

[6] The tagger is freely available at http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

data. The “train-tree-tagger” module generates a binary parameter file.

The lexicon is a file that contains tokens (word forms and punctuation), in the format of one per line. The TreeTagger was developed for languages with inflection, i.e. languages where one word may occur in a number of allomorphs. To ease the task of tagging such languages, the tool can operate on the level of a base form, the “lemma” which may be added to every word form. In the case of Dzongkha, lemmatization has not been developed yet, therefore, we either make use of the word form itself, or a hyphen in the case that no lemma is available. As table (2) demonstrates, the lexicon contains the word forms in the first column, the second column contains the POS and the third a hyphen. In the case of ambiguous entries, one line may contain a sequence of tag-“lemma” pairs that follow the word form of the first column. The lexicon may not contain any spaces; all columns must be separated by exactly one tab(ulator). Because the lexicon is only made use of during the training phase of the tagger, any update must result in reproducing the parameter file.

Word	Pos tag	lemma	Pos tag	lemma
ཀ་	NN	ཀ་		
ཀ་ཀ་	NN	ཀ་ཀ		
ཀ་ཀུ་ཙ་	NNP	--		
ལས་	PP	ལས་	CC	ལས་

Table (2) Example entries of the lexicon

Open class tags: a file containing the list of open class tags, i.e. the productive classes (one entry per line), cf. Appendix A. In the upcoming version of the tagset, tags of the following fused forms will be added, like, e.g. NN+CG (combination of all forms nouns with genitive case CG), VB+CG (combination of all forms verb with genitive case CG), JJ+CG (combination of all forms of adjective with genitive case CG), RB+CG (combination of all adverb with genitive case CG), and same with the combination of Subordinate conjunction NN+SC, VB+SC, JJ+SC, RB+SC, just to name a few.

Tagged training data: a file that contains tagged training data. The data must be stored in one-token-per-line format. This means that each line contains one token and its respective tag, these are separated by one tabulator. The file should be cleaned from empty lines, no meta-information, like, e.g. SGML markup is allowed.

b) Tagging

Two files serve as input: the binary parameter file and a text file that is to be tagged.

Parameter file: the file that was generated by step a) above.

Text file: a file that is to be tagged; it is mandatory that the data in this file appears in a one-token-per line format.

TnT (Brants, 2000):The Trigram’s’n’Tags (TnT) tagger was developed by Thorsten Brants (Brants, 2000). It is language independent and has been used widely for a number of languages, often yielding an accuracy of +96% without utilizing an external lexicon or an open word class file. TnT is based on Markov models, and takes not only distributional data of tokens, but also the final sequences of characters of a token into account when guessing the POS of a word. It can use the same format for training data as the TreeTagger, therefore, in order to use TnT for comparison reasons, no additional preparations for tagging Dzongkha are necessary.

5 Validation and Results

5.1 k-fold cross validation and bootstrapping

When applying a tagset to training data for the first time, it is advisable to progress in steps and to validate each step separately: One begins with annotating a rather small portion of text that is then divided into k number of slices. Slices k-1 are then utilized to create a parameter file, the slice k is stripped of its annotations and annotated by the tagger using that parameter file. The same procedure is followed for all other slices (“k-fold cross validation”).

Afterwards, a comparison between the original tags with the tags assigned by the tagger will then help to judge upon a number of issues, like, e.g., whether the size of the training data is sufficient (quantitative review). Examining the most frequent (typical) assignment errors of the tagger will also support the enhancement of the tagset: if e.g. the distribution of two different tags is more or less identical, a probabilistic tagger will not succeed in making the right choices, here, one is to consider if using one tag would be acceptable from a linguistic point of view (qualitative review).

The knowledge gained here usually leads to updates in the tagset and/or to the necessity to add more amounts of texts containing constellations that were found as being problematic for probabilistic tagging for they occur too rarely in the texts. After such updates are done on the existing training texts and tagset respectively, the k-fold validation may be repeated and reviewed again.

Updating training data and tagset will be repeated until the tagging results are satisfying (such a progressing method is usually called “bootstrap-ping”).

5.2 TreeTagger results

The work on automatic part of speech tagging for Dzongkha began with the manual annotation of 20,000 tokens. Because a non-linguistic person performed the process manually, the language coordinator did thorough correction.

The 20,000 token training set, made use of 43 different single tags (of 47 provided by the tagset). The token-tag combinations from there were combined with an external lexicon produced from a dictionary; the resulting lexicon file thus contained all types.

The 10-fold cross validation resulted in an accuracy of around 78%. Result introspection lead to the knowledge that more data had to be added and that fused words will have to receive separate tags. It also showed that manual tokenization is an error-prone procedure, as a significant number of word and sentence borders had to be corrected in the data.

After updating tagset and training data, another 20,247 tokens were added to the training set and the lexicon was updated accordingly, except for the fused forms, where a final solution on how to tag them is not found yet. The tagset was extended to 66 tags (cf. Appendix A). With a full knowledge of the possible tag-token combinations, the Tree-Tagger achieved a median accuracy of 93.1%.

5.3 TnT results and comparison with the TreeTagger

Using the 40,247 tokens text segment, a 10-fold cross validation was also performed with the TnT tagger. It achieves a 91.5 % median accuracy when the tagset containing 47 tags is applied. Results for each slice and mean/median can be found in table (3) of both taggers for comparison reasons. TnT reports on the number of unknown tokens detected in each slice; the mean of 16.49 % (median 14.18%) of unknown tokens offers an explanation why TnT does not perform as good as the TreeTagger which was supplied with a complete lexicon thus not being faced with unknown tokens at all.

Tagger:	Tree-Tagger accuracy %	TNT accuracy %
slice 1	92.13	92.33
slice 2	84.61	89.73
slice 3	89.08	89.88
slice 4	90.17	90.43
slice 5	92.95	91.01
slice 6	93.32	91.35
slice 7	94.24	91.69
slice 8	93.32	92.03
slice 9	95.21	92.55
slice 10	94.56	92.60
Mean	91.96	91.36
Median	93.14	91.20

Table (3) 10-fold cross validation results for TreeTagger and TnT

A qualitative review of the results showed that usually it is the tag CC that is confused with others (NN, DT, NN, DT, PRL, etc.) by TnT, while the TreeTagger is rather confusing NN (with VB, NNP, PRL, CC).

However, a more thorough qualitative examination of these results is still to be done and may lead to further updates on the tagset.

6 Discussion, Conclusions and Future work

This paper describes the building of NLP resources for the national language of Bhutan, namely Dzongkha. We have designed and built an electronic corpus containing 570,247 tokens belonging to different text types and domains.

Automated word segmentation with a high precision/recall still remains a challenge. We have begun to examine statistical methods to find solutions for this and we plan to report on our progress in the near future.

We have developed a first version of a tag set on the basis of the Penn Tree tagset for English (cf. section 4.1) A training data set of 40,247 tokens has been tagged manually and thoroughly checked. Lastly, we have tagged the corpus with the TreeTagger (Schmid, 1994) using a full form lexicon achieving 93.1% and, for comparison reasons, with TnT (Brants, 2000), without a lexicon, achieving 91.5 %.

We have used the present output in the construction of an advance Dzongkha TTS (text to speech) using an HMM-based method which is developed by the Bhutan team in collaboration with HLT team at NECTEC, Thailand[7]

Loads of work still remains, we are still to examine the tagger results from a qualitative aspect in order to answer inter Alia the following questions: Are there any further updates on the tag set necessary, what is the best way to process fused forms. Quantitative aspects might also still play a role: It still might be necessary to add further training data containing part of speech constellations that rarely occur, so tagger results for those will enhance.

We also plan to increase our corpus collection from various ranges of domains. At present there are more media, e.g. newspapers available in the world wide web, we will be able to collect such texts easily. In Bhutan, there is an ongoing project on OCR (optical character recognition) of Dzongkha under the PAN project (www.PANL10n.net). Given the success of this project, we will be able to scan text from textbooks.

[7] http://www.nectec.or.th/en/

Acknowledgments

This research work carried out as part of PAN Localization Project (www.PAN10n.net) with the aid of grant from the International Development Research Center (IDRC), Ottawa, Canada, administered through the Center of Research in Urdu language Processing (CRULP), National University of Computer and Emerging Sciences (NUCES), Pakistan. The research team would also like to thank PD Dr. Helmut Schmid and Prof. Dr. Heid, Insitut für Maschinelle Sprachverarbeitung (NLP institute), Universität Stuttgart, Germany for their valuable support and contributions that made this research successful.

References

Brants, Thorsten. 2000. TnT - as statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA, USA, pages 224 – 231.

Hasan, Fahim Muhammad, Naushad UzZaman, and Mumit Khan. 2007. Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla. *PAN Localization Working Papers*, 2004-2007, pages 31-37.

Hassan, Sajjad and Helmut Schmid . 2009. Tagging Urdu Text with Parts of Speech: A Tagger Comparison, *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* . Athens, Greece, 2009.

Muaz, Ahmed, Aasim Ali, and Sarmad Hussain. 2009. *Analysis and Development of Urdu POS Tagged Corpus*. Association for Computational Linguistics. Morristown, NJ, USA. Retrieved December 1, 2009, from http://www.lancs.ac.uk

Schmid, Helmut. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, pages 44 – 49.

Sarkar, Asif Iqbal, Dewan Shahriar Hossain Pavel and Mumit Khan. 2007. *Automatic Bangla Corpus Creation*. BRAC University, Dhaka, Bangladesh.

PAN Localization Working Papers, 2004-2007. pages 22-26.

Taljard Elsabé, Faaß Gertrud, Heid Ulrich, and Daan J. Prinsloo. 2000. On the development of a tagset for Northern Sotho with special reference to standardization. *Literator 29(1)*, April 2008 (special edition on Human Language Technologies), South Africa, pages 111 – 137

APPENDIX A

The Dzongkha Tagset
as used for the validation tests

Type	**SubClass**	**Label**
Open classes:		
Noun	Common Noun	NN
	Honorific form	NNH
	Particular/Person	NNP
	Quantifier	NNQ
	Plural	NNS
Verb	Aspirational	VBAs
	Honorific	VBH
	Agentive	VBAt
	Non-Agentive	VBNa
	Auxiliary	VBAUX
	Imperative	VBI
	Modal	VBMD
	Past participle	VBN
	Verb	VB
Adjective	Characteristic	JJCt
	Periodic	JJP
	Comparative	JJR
	Superlative	JJS
	Adjective	JJ
Adverb	Behavioral	RBB
	Comparative	RBR
	Superlative	RBS
	Adverb	RB
Interjection		UH
Closed classes:		
Marker	Affirmative	AM
	Interrogative	IrM
	Tense	TM
Case marker	Ablative Case	CA
	Dative Case	CDt
	Genitive Case	CG
	Vocative Case	CV
Pronouns	Locative	PRL
	Differential	PRD
	Personal	PRP
	Reflexive	PRRF
Conjunction	Coordinate	CC
	Subordinate	SC
Number	Cardinal Number	CD
	Ordinal Number	OD
	Nominal Number	ND
Ad position	Post position	PP
Determiner	Definite	DT
	Possessive	DT$
	Indefinite	DTI
Negator		NEG
Punctuation		PUN
Combined tags:		
Noun+Genitive case(CG)	Common+CG	NNCG
	Particular+CG	NNPCG
	Quantifier+CG	NNQCG
	Plural+CG	NNSCG
Adjective+CG	Adjective+CG	JJCG
	Characteristic +CG	JJCtCG
	Periodic+CG	JJPCG
Verb+CG	Honorific+CG	VBHCG
	Agentive+CG	VBAtCG
	Verb+CG	VBCG
	Modal+CG	VBMDCG
DefiniteDeterminer+CG	Determiner+CG	DTCG
Locative Pronoun +CG	Locative+CG	PRLCG
Negator+CG	Negator+CG	NEGCG
Noun+ Subordinate Conjunction(SC)	Common Noun +SC	NNSC
Verb+SC	Verb+SC	VBSC
	Agentive+SC	VBAtSC
	Modal verb+SC	VBMDC
Affirmative +SC	Affirmative +SC	AMSC
Negator+SC	Negator+SC	NEGSC

Unaccusative/Unergative Distinction in Turkish: A Connectionist Approach

Cengiz Acartürk
Middle East Technical University
Ankara, Turkey
acarturk@acm.org

Deniz Zeyrek
Middle East Technical University
Ankara, Turkey
dezeyrek@metu.edu.tr

Abstract

This study presents a novel computational approach to the analysis of unaccusative/unergative distinction in Turkish by employing feed-forward artificial neural networks with a backpropagation algorithm. The findings of the study reveal correspondences between semantic notions and syntactic manifestations of unaccusative/unergative distinction in this language, thus presenting a computational analysis of the distinction at the syntax/semantics interface. The approach is applicable to other languages, particularly the ones which lack an explicit diagnostic such as auxiliary selection but has a number of diagnostics instead.

1 Introduction

Ever since Unaccusativity Hypothesis (UH, Perlmutter, 1978), it is widely recognized that there are two heterogeneous subclasses of intransitive verbs, namely unaccusatives and unergatives. The phenomenon of unaccusative/unergative distinction is wide-ranging and labeled in a variety of ways, including active, split *S*, and split intransitivity (SI). (cf. Mithun, 1991).[1]

Studies dealing with SI are numerous and recently, works taking auxiliary selection as the basis of this syntactic phenomenon have increased (cf. McFadden, 2007 and the references therein). However, SI in languages that lack explicit syntactic manifestations such as auxiliary selection has been less studied.[2] Computational approaches are even scarcer. The major goal of this study is to discuss the linguistic issues surrounding SI in Turkish and present a novel computational approach that decides which verbs are unaccusative and which verbs are unergative in this language. The computational approach may in turn be used to study the split in lesser-known languages, especially the ones lacking a clear diagnostic. It may also be used with well-known languages where the split is observed as a means to confirm earlier predictions made about SI.

2 Approaches to Split Intransitivity (SI)

Broadly speaking, approaches to the SI may be syntactic or semantic. Syntactic approaches divide intransitive verbs into two syntactically distinct classes. According to the seminal work of Perlmutter (1978), unaccusative and unergative verbs form two syntactically distinct classes of intransitive verbs. Within the context of Relational Grammar, Perlmutter (1978) proposed that unaccusative verbs have an underlying object promoted to the subject position, while unergative verbs have a base-generated subject. This hypothesis, known as the Unaccusativity Hypothesis (UH) maintains that the mapping of the sole argument of an intransitive verb onto syntax as subject or direct object is semantically predictable. The UH distinguishes active or activity clauses (i.e., unergative clauses) from unaccusative ones. Unergative clauses include

[1] In this paper, the terms *unaccusative/unergative distinction* and *split intransitivity (SI)* are used interchangeably.

[2] An exception is Japanese. For example see Kishimoto (1996), Hirakawa (1999), Oshita (1997), Sorace and Shomura (2001), and the references therein. Also see Richa (2008) for Hindi.

Proceedings of the 8th Workshop on Asian Language Resources, pages 111–119,
Beijing, China, 21-22 August 2010.

willed or volitional acts (*work*, *speak*) and certain involuntary bodily process predicates (*cough*, *sleep*); unaccusative clauses include predicates whose initial term is semantically patient (*fall*, *die*), predicates of existing and happening (*happen*, *vanish*), nonvoluntary emission predicates (*smell*, *shine*), aspectual predicates (*begin*, *cease*), and duratives (*remain*, *survive*).

From a Government and Binding perspective, Burzio (1986) differentiates between two intransitive classes by the verbs' theta-marking properties. In unaccusative verbs (labeled 'ergatives'), the sole argument is the same as the deep structure object; in unergative verbs, the sole argument is the same as the agent at the surface. The configuration of the two intransitive verb types may be represented simply as follows:

Unergatives:	NP [$_{VP}$ V]	John ran.
Unaccusatives:	[$_{VP}$ V NP]	John fell.

In its original formulation, the UH claimed that the determination of verbs as unaccusative or unergative somehow correlated with their semantics and since then, there has been so much theoretical discussion about how strong this connection is. It has also been noted that a strict binary division is actually not tenable because across languages, some verbs fail to behave consistently with respect to certain diagnostics. For example, it has been shown that, with standard diagnostics, certain verbs such as *last, stink, bleed, die,* etc can be classified as unaccusative in one language, unergative in a different language (Rosen, 1984; Zaenen, 1988, among many others). This situation is referred to as unaccusativity mismatches. New proposals that specifically focus on these problems have also been made (e.g., Sorace, 2000, below).

2.1 The Connection of Syntax and Semantics in SI

Following the initial theoretical discussions about the connection between syntactic diagnostics and their semantic underpinnings, various semantic factors were suggested. These involve directed change and internal/external causation (Levin & Rappaport-Hovav, 1995), inferable eventual position or state (Lieber & Baayen, 1997), telicity and controllability (Zaenen, 1993), and locomotion (see Randall, 2007; Alexiadou et al., 2004, and, Aranovich, 2007, and McFadden, 2007 for reviews). Some researchers have suggested that syntax has no role in SI. For example, van Valin (1990), focusing on Italian, Georgian, and Achenese, proposed that SI is best characterized in terms of Aksionsart and volitionality. Kishimoto (1996) suggested that volitionality is the semantic parameter that largely determines unaccusative/unergative distinction in Japanese.

Auxiliary selection is among the most reliable syntactic diagnostics proposed for SI. This refers to the auxiliary selection properties of languages that have two perfect auxiliaries corresponding to *be* and *have* in English. In Romance and Germanic languages such as Italian, Dutch, German, and to a lesser extent French, the equivalents of *be* (*essere*, *zijn*, *sein*, *etre*) tend to be selected by unaccusative predicates while the equivalents of *have* (*avere*, *haben*, *hebben*, *avoir*) tend to be selected by unergative predicates (Burzio, 1986; Zaenen, 1993; Keller, 2000; Legendre, 2007, among others). In (1a–b) the situation is illustrated in French (F), German (G) and Italian (I). (Examples are from Legendre, 2007).

(1) a. *Maria a travaillé (F)/hat gearbeitet (G)/ha lavorato (I).*
'Maria worked.'
b. *Maria est venue (F)/ist gekommen (G)/é venuta (I).*
'Maria came.'

Van Valin (1990) and Zaenen (1993) discuss auxiliary selection as a manifestation of the semantic property of telicity. Hence in Dutch, *zijn*-taking verbs are by and large telic, *hebben*-taking verbs are atelic.

Impersonal passivization is another diagnostic that seems applicable to a wide range of languages and used by a number of authors, e.g. Perlmutter (1978), Hoekstra and Mulder (1990), Keller (2000). This construction is predicted to be grammatical with unergative clauses but not with unaccusative clauses. Zaenen (1993) notes that impersonal passivization is controlled by the semantic notion of protagonist control in Dutch; therefore incompatibility of examples such as *bleed* with impersonal passivization is attributed to the fact that *bleed* is not a protagonist control verb. Levin and Rappaport-Hovav (1995:141) take impersonal passivization as an unaccusativity diagnostic but take its sensitivity to protagonist control as a necessary but an insufficient condition for unergative verbs to allow it. In other words, only unergative verbs will be found in this construction, though not all of them.

Refinements of the UH have also been proposed. Most notably, Sorace (2000) argued that the variation attested across languages (as well as within the dialects of a single language) is orderly, and that there are a number of cut-off points to which verb classes can be sensitive.

Sorace's work on (monadic) intransitive verbs is built on variation in the perfective auxiliary selection of verbs in Romance and Germanic languages and called Auxiliary Selection Hierarchy (ASH). She demonstrates that the variation is based on a hierarchy of thematic and aspectual specification of the verbs (viz., telicity and agentivity) and that it is a function of the position of a verb on the hierarchy. Verbs with a high degree of aspectual and thematic specification occupy the extreme ends; variable verbs occupy the middle position, reflecting the decreasing degree of aspectual specification. Both cross-dialectally and across languages, these verbs may be used either with unaccusative or unergative syntax.[3] The ASH therefore is a descriptive statement considering auxiliary selection as a property characterized by both syntax and semantics, as originally viewed by the UH.

We now turn to Turkish, which lacks perfective auxiliaries. A number of other syntactic diagnostics, reviewed below, have been proposed but unlike auxiliaries in other languages, these are not obligatory constructions in Turkish. In addition, the semantic properties underlying the proposed diagnostics have not been studied extensively. Therefore, Turkish presents a particular challenge for any study about SI.

3 Diagnostics for SI in Turkish

Just as other languages, intransitive verbs in Turkish are sensitive to a set of syntactic environments, summarized below.

3.1 The *–ArAk* Construction

One of the diagnostics is the *–ArAk* construction, which is an adverbial clause formed with the root verb plus the morpheme *–ArAk* (Özkaragöz, 1986). In a Turkish clause which involves the verbal suffix *–ArAk*, both the controller (the complement verb) and the target (the matrix verb) have to be either unaccusative or unergative. In addition, both the controlled and the target have to be the final (surface) subjects of the clause. The examples below contain sentences where both the controller and the target verbs are unaccusative (2) or both are unergative (3). The examples also contain ungrammatical sentences where the controller verb is unergative whereas the target verb is unaccusative (4), and those in which the controller verb is unaccusative whereas the target verb is unergative (5). (Examples are from Özkaragöz, 1986).

(2) *Hasan [kol-u kana -y -arak] acı çek -ti.*
arm-POSS bleed-GL-*ArAk* suffer -PST
'Hasan, while his arm bled, suffered.'

(3) *Kız [(top) oyna-y -arak] şarkı söyle-di.*
girl ball play-GL-*ArAk* sing -PST
'The girl, while playing (ball), sang.'

(4) * *Kız [(top) oyna-y -arak] kay-dı.*
girl ball play-GL-*ArAk* slip -PST
'The girl, while playing (ball), slipped.'

(5) * *Kız [kayak kay-arak] düş-tü.*
girl ski-*ArAk* fall-PST
'The girl, while skiing, fell.'

3.2 Double Causatives

Double construction is allowed with unaccusative verbs but not with unergatives, as shown in (6) and (7) below (Özkaragöz, 1986).

(6) *Sema Turhan-a çiçeğ-i sol- dur -t -tu.*
-DAT flower-ACC fade-CAUS-CAUS-PST

[3] The claim that the two notions of ASH lie within a single dimensional hierarchy has been questioned by Randall (2007). The ASH has also been criticized since it does not explain the reason why a certain language shows the pattern it does (McFadden, 2007).

‘Sema made Turhan cause the flower to fade.’

(7) ** Ben Turhan-a Sema-yı koş-tur -t -t -um*
I -DAT -ACC run-CAUS-CAUS-PST-1sg
‘I made Turhan make Sema run.’

3.3 Gerund Constructions

The gerund constructions *–Irken* ‘while’ and *–IncE* ‘when’ are further diagnostics. The former denotes simultaneous action and the latter denotes consecutive action. Unergative verbs are predicted to be compatible with the *–Irken* construction, whereas unaccusatives are predicted to be compatible with the *–IncE* construction, as shown in (8) and (9).[4]

(8) *Adam çalış-ırken esne-di.*
man work-*Irken* yawn-PAST.3per.sg
‘The man yawned while working.’

(9) *Atlet takıl-ınca düş-tü.*
athlete trip-*IncE* fall-PAST.3per.sg
‘The athlete when tripped fell.’

3.4 The Suffix *–Ik*

It has also been suggested that the derivational suffix *–Ik*, used for deriving adjectives from verbs, is compatible with unaccusatives but not with unergatives, as shown in (10) and (11).

(10) *bat-ık gemi*
sink-Ik ship
the sunk ship

(11) **çalış-ık adam*
work-Ik man
the worked man

3.5 The *–mIş* Participle

The past participle marker *–mIş*, which is used for deriving adjectives from verbs has been proposed as yet another diagnostic. The suffix *–mIş* forms participles with transitive and intransitive verbs, as well as passivized verbs. The basic requirement for the acceptability of the *–mIş* participle is the existence of an internal argument in the clause. In well-formed *–mIş* participles, the modified noun must be the external argument of a transitive verb (e.g., *anne* ‘mother’ in [12]), or the internal argument of a passivized verb (e.g., *borç* ‘debt’ in [13]). The internal argument of a transitive verb is not allowed as the modified noun as illustrated in (14).

(12) *Çocuğu-n-u bırak-mış anne*
Child-POSS-ACC leave-*mIş* mother
‘a/the mother who left her children’

(13) *Öde-n-miş borç*
pay-PASS-*mIş* debt
‘the paid debt’

(14) **Öde-miş borç*
pay-*mIş* debt
*‘the pay debt’

As expected, the adjectives formed by intransitive verbs and the *–mIş* participle is more acceptable with unaccusatives compared to unergatives, as shown in (15) and (16).

(15) *sol-muş/ karar-mış çiçek*
wilt/ blacken -*mIş* flower
‘The wilted/blackened flower’

(16) **sıçra-mış/ yüz-müş/ bağır-mış çocuk*
jump/ swim/ shout -*mIş* child
‘The jumped/ swum/ shouted child’

3.6 Impersonal Passivization

Impersonal passivization, used as a diagnostic to single out unergatives by some researchers, appears usable for Turkish as well. In Turkish, impersonal passives carry the phonologically conditioned passive suffix marker, *-Il*, accompanied by an indefinite human interpretation and a resistance to agentive by-phrases. It has been suggested that the tense in which the verb appears affects the acceptability of impersonal passives: when the verb is in the aorist, the implicit subject has an arbitrary interpretation, i.e. either a generic or existential interpretation. On the other hand, in those cases when the verb is in past tense, the implicit subject has a referential meaning, namely a first person plural reading. It was therefore suggested that impersonal passivization is a proper diagnostic environment only in the past tense, which was also adopted in the present study (Nakipoğlu-Demiralp, 2001, cf. Sezer, 1991). (17) and (18) exemplify

[4] Examples in sections 3.3 and 3.4 are from Nakipoğlu (1998).

impersonal passivization with the verb in the past tense.

(17) *Burada koşuldu.*
Here run-PASS-PST
'There was running here.' (existential interpretation)

(18) *??Bu yetimhanede büyündü.*
This orphanage-LOC grow-PASS-PST
'It was grown in this orphanage.'

The diagnostics summarized above do not always pick out the same verbs in Turkish. For example, most diagnostics will fare well with the verbs *düş-* 'fall', *gel-* 'come', *gir-* 'enter' (with a human subject) just as well as impersonal passivization. In other words, these verbs are unaccusative according to most diagnostics and unergative according to impersonal passivization. The opposite of this situation also holds. The stative verb *devam et-* 'continue' is bad or marginally acceptable with most diagnostics as well as impersonal passivization.

The conclusion is that in Turkish, acceptability judgments with the proposed diagnostic environments do not yield a clear distinction between unaccusative and unergative verbs. In addition, it is not clear which semantic properties these diagnostics are correlated with. The model described below is expected to provide some answers to these issues. It is based on native speaker judgments but it goes beyond them by computationally showing that there are correspondences between semantic notions and syntactic manifestations of SI in Turkish. The model is presented below.

4 The Model

This study employs feed-forward artificial neural networks with a backpropagation algorithm as computational models for the analysis of unaccusative/unergative distinction in Turkish.

4.1 Artificial Neural Networks and Learning Paradigms

An artificial neural network (ANN) is a computational model that can be used as a non-linear statistical data modeling tool. ANNs are generally used for deriving a function from observations, in applications where the data are complex and it is difficult to devise a relationship between observations and outputs by hand. ANNs are characterized by interconnected group of artificial neurons, namely nodes. An ANN generally has three major layers of nodes: a single input layer, a single or multiple hidden layers, and a single output layer. In a feedforward ANN, the outputs from all the nodes go to succeeding but not preceding layers.

There are three major learning paradigms that are used for training ANNs: supervised learning, unsupervised learning, and reinforcement learning. A backpropagation algorithm is a supervised learning method which is used for teaching a neural network how to perform a specific task. Accordingly, a feed-forward ANN with a backpropagation algorithm is a computational tool that models the relationship between observations and output by employing supervised learning method (see Hertz et al., 1991; Anderson & Rosenfeld, 1988, among many others for ANNs). The following section presents how such an ANN is used for analyzing unaccusative/unergative distinction in Turkish.

4.2 The Analysis

Two feed-forward ANNs with a backpropagation algorithm were developed for the analysis. Both models had a single input layer, a single hidden layer, and a single output layer of nodes. Both models had a single output node, which represents the binary status of a given verb as unaccusative (0) or unergative (1). The number of nodes in the hidden layer was variable (see below for a discussion of network parameters).

The difference between the two models was the design of the input layer. The first model (henceforth, the diagnostics model DIAG) took diagnostics as input nodes, whereas the second model (henceforth, the semantic parameters model SEMANP) took semantic parameters as input nodes, as presented in detail below.

The Diagnostics Model (DIAG): Binary acceptability values of the phrases or sentences formed by the syntactic diagnostics constituted the input nodes for the network (see above for the SI diagnostics). Each syntactic diagnostic provided a binary value (either 0 or 1) to one of the input nodes. For example, consider the *–mIş* participle as one of the syntactic diagnostics for SI in Turkish. As discussed above, the *–mIş* participle forms acceptable adjectival phrases with unaccusative verbs (e.g., *sol-* 'wilt') but not with the unergative verbs (e.g., *sıçra-* 'jump, leap'), as shown in (19) and (20) below.

(19) *sol-muş çiçek-ler*
wilt-*mIş* flower-PLU
'wilted flowers'
(20) **sıçra-mış sporcu-lar*
jump-*mIş* sportsman-PLU
'jumped sportsmen'

Accordingly, for the verb *sol-* 'wilt', the *–mIş* participle diagnostic provides the value *1* with one of the input nodes, whereas for the verb *sıçra-* 'jump, leap' it provides the value *0* with the corresponding input node. In this way, the syntactic diagnostics constituted an input pattern with eight members for each verb.[5] The construction of an input pattern is exemplified in (21) for the unergative verb *konuş-* 'talk'.

(21) *A sample input pattern for DIAG.*

a. **Adam konuşarak kızardı.* :0
The man talk-*ArAk* blush-PST
'The man blushed by talking.'
b. *Adam konuşarak yürüdü.* :1
The man talk-*Arak* walk-PST
'The man walked by talking.'
c. **Adam kadına çocuğu konuşturttu.* :0
'The man made the woman have the boy talked.'
d. *Adam konuşurken yürüdü.* :1
The man talk-*Irken* walk-PST
'The man walked while talking.'
e. *Adam konuşunca yürüdü.* :1
The man speak-*IncE* walk-PST
'The man walked when he talked.'
f. **Konuş-uk adam* :0
Talk-*Ik* man
'The talked man'
g. **Konuş-muş adam* :0
Talk-*mIş* man
'The talked man'
h. *Törende konuşuldu.* :1
Ceremony-LOC talk-PASS
'It was spoken in the ceremony.'

Accordingly, the input pattern for the verb *konuş-* 'talk' is schematically shown below.

0	1	0	1	1	0	0	1

The Semantic Parameters Model (SEMANP): The input nodes for the SEMANP network were constituted by four binary values that represented the status of four semantic parameters (telicity, volitionality, dynamicity, and directed motion) for each verb. Each semantic parameter provided a binary value (either *0* or *1*) to one of the input nodes. The value of the input nodes were determined by applying the following tests for the relevant semantic aspects: (1) *in/for an hour* test for telicity (e.g. the phrase *to talk *in/for an hour* shows that the verb *talk* is atelic whereas the phrase *to wilt in/*for an hour* shows that the verb *wilt* is telic), (2) *on purpose* test for volitionality, (3) *hala-* 'still' test for dynamicity, (4) and the dative test (i.e., acceptability of adding a dative term to the verb) for directed motion.

The construction of an input pattern for SEMANP is exemplified in (22) for the unaccusative verb *sol-* 'wilt'.

(22) *A sample input pattern for DIAG.*

a.	Telic	:1
b.	Non-volitional	:0
c.	Non-dynamic	:0
d.	No directed motion	:0

Accordingly, the input pattern for the verb *sol-* 'wilt' is schematically shown below.

1	0	0	0

4.3 The Training Phase

The network was trained by providing patterns for 52 verbs that are recognized as unaccusatives in the SI literature or placed closer to the unaccusative end rather than the unergative end of the Auxiliary Selection Hierarchy (ASH, Sorace, 2000); and 52 verbs that are recognized to be unaccusative in the SI literature or placed closer to the unergative end rather than the unaccusative end of the ASH. As a result, a total of 104 input patterns, each composed of eight nodes, were used to train the DIAG model and 104 input patterns, each composed of four nodes, were used to train the SEMANP model. The

[5] One of the syntactic diagnostics (the gerund suffix *–ArAk*) involves two verbs (i.e., the target and the matrix verb). Therefore, two sentences/phrases were formed–one with unaccusatives and the other with unergatives–which provided two binary values with the input pattern.

single output node was set to *0* if the verb with the given input pattern was unaccusative and it was set to *1* if the verb was unergative. Supervised learning method was used, as employed by the backpropagation algorithm.

One hidden layer with a variable number of hidden units was used (see below for the analysis of model parameters). Sigmoid activation function, shown in (23), was used for modeling the activation function.

(23) $$f(x) = \frac{1}{1+e^{-x}}$$

The number of maximum iterations per epoch was set to 20. The system sensitivity was defined by a global variable (ε=0.01) which decided whether the loops in the code converge or not.

4.4 The Test Phase

The DIAG and SEMANP models were tested by providing the following input patterns:

Group A: five verbs that are either recognized as unaccusatives in the SI literature or placed closer to the unaccusative end rather than the unergative end of the ASH.

Group B: Five verbs that are either recognized as unergatives in the SI literature or placed closer to the unergative end rather than the unaccusative end of the ASH.

Group C: Three verbs that are reported to exhibit variable behavior within the ASH.

After the training, the networks provided the binary outputs for the test verbs, which showed whether a test verb was unaccusative or unergative according to the models.

5 Results

The results are presented in the two sections below, separately for the DIAG model and for the SEMANP model.

5.1 The DIAG Model

After the training of the network and the optimization of the number of hidden units and the learning rate, the DIAG model classified all verbs in Group A as unaccusatives. The model also classified all Group-B verbs as unergatives. Finally, the model categorized three Group-C verbs that were reported to show variable behavior (*kana-* 'bleed', *parla-* 'shine' and *üşü-* 'be, feel cold') as unaccusative verbs in Turkish.

The distribution of weights after the training showed that the *–mIş* participle received the highest weight, which indicates that the *–mIş* participle is the most reliable diagnostics for analyzing unaccusative/unergative distinction in Turkish.

5.2 The SEMANP Model

The SEMANP model classified two of the Group-A verbs (namely, *gir-* 'enter' and *yetiş-* 'grow') as unaccusatives and the three remaining verbs (*dur-* 'remain, stay', *kal-* 'stall, stay, and *varol-* 'exist') as unergatives. The model also classified four of five Group-B verbs (*gül-* 'laugh', *sırıt-* 'grin', *söylen-* 'mutter', *yakın-* 'complain') as unergatives and the remaining verb (*yüz-* 'swim') as unaccusative. Finally, the model categorized three Group-C verbs (*kana-* 'bleed', *parla-* 'shine' and *üşü-* 'be, feel cold') as unaccusative verbs in Turkish.

The distribution of weights after the training showed that among the four semantic parameters that were selected in this study, telicity received the highest weight, which indicates that unaccusative and unergative verbs are most sensitive to the telicity aspect of the verb in Turkish.

5.3 Evaluation of Model Parameters

Four model design parameters, their initial values and acceptable ranges after optimization are discussed below.

The number of hidden units: The number of hidden layers was set to 1 as a non-variable design parameter of the network. The initial number of hidden units was set to 3. Keeping the learning rate (η=-0.25) and the momentum term (λ=0.25) constant, the number of hidden units was adjusted and the behavior of the network was observed. The analyses showed that the optimum range for the number of hidden units was between 2 and 6.

The learning rate: The learning rate was initially set to η=-0.25. Keeping the number of hidden units (hidden_size=3) and the momentum term (λ=0.25) constant, adjusting the learning rate between η=-0.005 and η=-0.9 did not have an effect on the results.

The momentum term: The momentum term was set to λ=0.25 initially. Keeping the number of hidden units (hidden_size=3) and the learning rate (η=-0.25) constant, adjusting the momentum term between λ=0.01 and λ=1.0 did not have an effect on the results. However, the system did not converge to a solution for the momentum term equal to and greater than λ=1.0.

6 Discussion

A major finding of the suggested model is that the predictions of the two models are compatible with the UH (Perlmutter, 1978) in that they divide most intransitive verbs into two, as expected. Furthermore, the differences between the decisions of the diagnostics-based DIAG model and the semantic-parameters-based SEMANP model reflect a reported finding in the unaccusativity literature, i.e., the tests used to differentiate between unaccusatives and unergatives do not uniformly delegate all verbs to the same classes (the solution of why such mismatches occur in Turkish is beyond the scope of this study, see Sorace, 2000; Randall, 2007, for some suggestions). More specifically, the three Group-A verbs that were predicted as unaccusative by the DIAG model and unergative by the SEMANP model (*dur*- 'remain, stay', *kal*- 'stall, stay, and *varol*- 'exist') are stative verbs, which are known to show inconsistent behavior in the literature and classified as variable-behavior verbs by Sorace (2000). An unexpected finding is the Group-B verb (*yüz*- 'swim'), which is predicted as unergative by the DIAG model and unaccusative by the SEMANP model. This seems to reflect the role of semantic parameters other than telicity (namely, dynamicity and directed motion) in Turkish. The remaining nine verbs of thirteen tested verbs were predicted to be of the same type (either unaccusative or unergative) by both models.

Another finding of the model is the alignment between the most weighted syntactic diagnostics for unaccusative/unergative distinction in Turkish, namely the *–mIş* participle which received the highest weight after the training, and the most weighted semantic parameter, namely telicity.

7 Conclusion and Future Research

This study contributes to our understanding of the distinction in several respects.

Firstly, it proposes a novel computational approach that tackles the unaccusative/unergative distinction in Turkish. The model confirms that a split between unaccusative and unergative verbs indeed exists in Turkish but that the division is not clear-cut. The model suggests that certain verbs (e.g., stative verbs) behave inconsistently, as mentioned in most accounts in the literature. Moreover, the model reflects a correspondence between syntactic diagnostics and semantics, which supports the view that unaccusativity is semantically determined and syntactically manifested (Permutter, 1978, Levin & Rappaport-Hovav, 1995). Since this approach uses relevant language-dependent features, it is particularly applicable to languages that lack explicit syntactic diagnostics of SI.

The computational approach is based on the connectionist paradigm which employs feed-forward artificial neural networks with a back-propagation algorithm. There are several dimensions in which the model will further be developed. First, the reliability of input node values will be strengthened by conducting acceptability judgment experiments with native speakers, and the training of the model will be improved by increasing the number of verbs used for training. Acceptability judgments are influenced not only by verbs but also by other constituents in clauses or sentences; therefore the input data will be improved to involve different senses of verbs under various sentential constructions. Second, alternative classifiers, such as decision trees and naïve Bayes, as well as the classifiers that use discretized weights may provide more informative accounts of the findings of SI in Turkish. These alternatives will be investigated in further studies.

Acknowledgements We thank Cem Bozşahin and three anonymous reviewers for their helpful comments and suggestions. All remaining errors are ours.

References

Alexiadou, Artemis, Anagnostopoulou, Elena, and Everaert, Martin (eds.). 2004. *The unaccusativity puzzle: Explorations of the syntax-lexicon interface.* Oxford University Press.

Anderson, James, and Rosenfeld, Edward. 1988. *Neurocomputing: Foundations of research.* MIT Press.

Aranovich, Raúl (ed.). 2007. *Split auxiliary systems.* Amsterdam/Philadelphia: John Benjamins.

Burzio, Luigi. 1986. *Italian syntax: A Government-Binding approach*. Dordrecht: Reidel.

Hertz, John, Krogh, Anders, and Palmer, Richard G. 1991. *Introduction to the theory of neural computation*. Addison-Wesley, Massachusetts.

Hirakawa, Makiko. 1999. L2 acquisition of Japanese unaccusative verbs by speakers of English and Chinese. In *The acquisition of japanese as a second language*, ed. Kazue Kanno, 89-113. Amsterdam: John Benjamins.

Hoekstra, Teun, and Mulder, René. 1990. Unergatives as copular verbs: Locational and existential predication. *The Linguistic Review* 7: 1-79.

Keller, Frank. 2000. Gradience in grammar: Experimental and computational aspects of degrees of grammaticality. University of Edinburgh dissertation.

Kishimoto, Hideki. 1996. Split intransitivity in Japanese and the Unaccusative Hypothesis. *Language* 72: 248-286.

Legendre, Géraldine. 2007. On the typology of auxiliary selection. *Lingua* 117: 1522-1540.

Levin, Beth, and Rappaport-Hovav, Malka. 1995. *Unaccusativity at the syntax-lexical semantics interface*. MIT Press.

Lieber, R., and Baayen, H. 1997. A semantic principle of auxiliary selection in Dutch. *Natural Language and Linguistic Theory* 15: 789-845.

McFadden, Thomas. 2007. Auxiliary selection. *Language and Linguistics Compass* 1/6: 674-708.

Mithun, Marianne. 1991. Active/agentive case marking and its motivations. *Language* 67: 510-546.

Nakipoğlu, Mine. 1998. Split intransitivity and the syntax-semantics interface in Turkish, Minneapolis: University of Minnesota dissertation.

Nakipoğlu-Demiralp, Mine. 2001. The referential properties of the implicit arguments of impersonal passive constructions. In *The verb in Turkish*, ed. E. Eser Taylan, 129-150, John Benjamins.

Oshita, Hiroyuki. 1997. The unaccusative trap: L2 acquisition of English intransitive verbs. University of Southern California dissertation.

Özkaragöz, İnci Z. 1986. The relational structure of Turkish syntax, San Diego: University of California dissertation.

Perlmutter, David M. 1978. Impersonal passives and the Unaccusative Hypothesis. In *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society*, ed. Farrell Ackerman et al., 157-189.

Randall, J. 2007. Parameterized auxiliary selection: A fine-grained interaction of features and linking rules. In *Split auxiliary systems*, ed. Raúl Aranovich, 207-236. John Benjamins.

Richa, Srishti. 2008. *Unaccusativity, unergativity and the causative alternation in Hindi: A minimalist analysis*. New Delhi: Jawaharlal Nehru University dissertation.

Rosen, Carol G. 1984. The interface between semantic roles and initial grammatical relations. In *Studies in relational grammar*, eds. Carol G. Rosen and David M. Perlmutter. University of Chicago Press.

Sezer, Engin. 1991. *Issues in Turkish syntax*. Cambridge: Harvard University dissertation.

Sorace, Antonella. 2000. Gradients in auxiliary selection with intransitive verbs. *Language* 76: 859-890.

Sorace, Antonella, and Shomura, Yoko. 2001. Lexical constraints on the acquisition of split intransitivity. Evidence from L2 Japanese. *Studies in Second Language Acquisition* 23: 247-278.

van Valin, Robert D., Jr. 1990. Semantic parameters of split transitivity. *Language* 66: 221-260.

Zaenen, Annie. 1988. Unaccusative verbs in Dutch and the syntax-semantics interface. CSLI Reports 88-123, CSLI, Stanford University.

Zaenen, Annie. 1993. Unaccusativity in Dutch: Integrating syntax and lexical semantics. In *Semantics and the lexicon*, ed. James Pustejovsky. Dordrecht: Kluwer Academic Publishers.

A Preliminary Work on Causative Verbs in Hindi

Rafiya Begum, Dipti Misra Sharma
Language Technologies Research Centre, IIIT.
rafiya@research.iiit.ac.in,
dipti@iiit.ac.in

Abstract

This paper introduces a preliminary work on Hindi causative verbs: their classification, a linguistic model for their classification and their verb frames. The main objective of this work is to come up with a classification of the Hindi causative verbs. In the classification we show how different types of Hindi verbs have different types of causative forms. It will be a linguistic resource for Hindi causative verbs which can be used in various NLP applications. This resource enriches the already available linguistic resource on Hindi verb frames (Begum et al., 2008b). This resource will be helpful in getting proper insight into Hindi verbs. In this paper, we present the morphology, semantics and syntax of the causative verbs. The morphology is captured by the word generation process; semantics is captured by the linguistic model followed for classifying the verbs and the syntax has been captured by the verb frames using relations given by Panini.

1 Introduction

Verbs play a major role in expressing the meaning of a sentence and its syntactic behavior. They decide the number of participants that will participate in the action. Semantically verbs are classified into action verbs, state verbs and process verbs. Syntactically they are classified into intransitives, transitives and ditransitives. The morphological, semantic and syntactic properties of verbs play an important role in deeper level analysis such as parsing.

Causative verbs are differently realized in different languages. These verbs have been an interesting area of study. The study of causative constructions is important as it involves the interaction of various components such as semantics, syntax and morphology (Comrie, 1981). This paper presents the preliminary work on Hindi causative verbs.

2 Causative Verbs

Causative verbs mean that *some actor makes somebody else do something or causes him to be in a certain state* (Agnihotri, 2007). The causal verb indicates the *causing of another to do something, instead of doing it oneself* (Greaves, 1983). Semantically causative verbs refer to a *causative situation* which has two components: (a) the causing situation or the antecedent; (b) the caused situation or the consequent. These two combine to make a *causative situation* (Nedyalkov and Silnitsky, 1973). There are different ways in which causation is indicated in different languages. There are three types of causatives: Morphological causatives, Periphrastic causatives and Lexical causatives (Comrie, 1981).

Morphological Causatives indicate causation with the help of verbal affixes. Sanskrit, Hindi/Urdu, Persian, Arabic, Hebrew, Japanese, Khmer and Finnish languages have morphological causatives. **Periphrastic causatives** indicate causation with the help of a verb which occurs along with the main verb. For example, in English in a sentence such as:

(1) *John made the child drink milk.*

In the above example the verb *make* is expressing causation which is occurring along with the verb *drink* which in turn is expressing

Proceedings of the 8th Workshop on Asian Language Resources, pages 120–128,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

the main action. English, German and French are some of the languages which have periphrastic causatives. **Lexical causatives** are those in which there is no morphological similarity between the base verb root and the causative verb form. A different lexical item is used to indicate causation. For example, the causative of English *eat* is *feed*. English and Japanese have lexical causatives. English has both **periphrastic** and **lexical causatives**.

3 Causative verbs in Hindi

Causatives in Hindi are realized through a morphological process. In Hindi, a base verb root changes to a causative verb when affixed by either an *'-aa'* or a *'-vaa'* suffix.

Base verb	**First causal**	**Second causal**
so	*sul-aa*	*sul-vaa*
'sleep'	'put to sleep'	'cause to put to sleep'

In each step of causative derivation there is an increase in the valency of the verb (Kachru, 2006; Comrie, 1981)

(2) *baccaa soyaa*
child sleep.Pst
'The child slept.'

(3) *aayaa ne bacce ko sulaayaa*
maid Erg. child Acc. sleep.Caus.Pst
'The maid put the child to sleep.'

(4) *maa.N ne aayaa se bacce ko*
mother Erg. maid by child Acc.
sulvaayaa
sleep.Caus.Pst
'Mother caused the maid to put the child to sleep.'

Hindi verbs are divided into two groups based on their behaviour in causative sentences: **affective verbs** and **effective verbs** (Kachru, 2006). The action of affective verbs benefits or affects the agent. **Affective verbs** will have both first and second causal forms. Verbs such as *ronaa* 'to cry' and *dau.Dnaa* 'to run' are affective intransitive verbs. Only verbs belonging to *khaanaa* 'to eat' class come under affective transitive verbs. The agent of the affective intransitive verb becomes the patient and the agent of the affective transitive verbs becomes the recipient in the first causal and they both will take a *ko* postpositon (Hindi case marker). **Effective verbs** and ditransitive verbs have only one causal form. The agent of the effective verb and ditransitive verb becomes the causative agent in the first causal. So this causative agent in the first causal takes a *se* 'with' postposition (Hindi case marker). Verbs belonging to *karnaa* 'to do' class come under the effective verbs.

The major studies in Hindi causatives: Kachru (1966), Kachru (1980), Kachru (2006), McGregor (1995), Greaves (1983), Kellogg (1972), Agnihotri (2007), Sahaay (2004), Sahaay (2007), Singh (1997) and Tripathi (1986). Kachru (1966) has given the classification of Hindi verbs based on their causativization behavior. The others have mostly talked about the derivation process of the causative verbs.

However the classification of causative verbs in Hindi remains an issue of discussion. Since they are morphologically related, the decision of what is the base verb form of these verbs remains a point of discussion. There are two approaches which are followed in deciding the base verb: (1) causative formation based only on morphology, (2) causative formation based on morphology and semantics.

I.Based on Morphology

Base verb →	**First causal** →	**Second causal**
(Intransitive)	(Transitive)	(Causative)
khul	*khol*	*khulvaa*
'open'	'open'	'cause to open'

II. Based on Morphology and Semantics

Intransitive ←	**Base verb** →	**Second causal**
(Intransitive)	(Transitive)	(Causative)
khul	*khol*	*khulvaa*
'open'	'open'	'cause to open'

In I, *khul* 'open' is taken as the base verb and *khol* 'open' and *khulvaa* 'cause to open' are derived from it by adding suffix *'-aa'* and *'-vaa'* respectively to the base verb (Kachru, 1966; Kachru, 1980). The arrow denotes the direction of derivation from base verb. Here, the forward arrow denotes the increment of the argument from base to the causal forms. On the other hand, in II, *khol* 'open' is taken as the base verb. Here, other than morphology, the semantics of the verbs is also taken into consideration. Here *khul* 'open' and *khulvaa* 'cause to open' are derived from the base verb *khol* 'open'. *khulvaa* 'cause to open' is a causative verb which is derived from the base verb by adding suffix *'-vaa'* to it. *khul* 'open' is a derived intransitive form. The agent of the base verb *khol* 'open' is not realized on the surface level of the derived intransitive verb *khul* 'open' though it is implied semantically. Here there is both forward and backward derivation. From base verb to the derived intransitive it is a backward derivation which means there is a reduction of one argument from base verb to the derived intransitive verb (Tripathi, 1986; Reinhart, 2005).

In this paper, we motivate our work by presenting our approach for classifying the causative verbs in Hindi.

4 Our Approach

4.1 Linguistic Model for Classifying Causative verbs

We have followed Paninian Grammatical framework in this model as the theoretical basis for our approach. The meaning of every *verbal root (dhaatu)* consists of: (a) *activity (vyaapaara)* and; (b) *result (phala). Activity* denotes the actions carried out by the various participants *(karakas)* involved in the action. *Result* denotes the state that is reached, when the action is completed (Bharati et al., 1995). The participants of the action expressed by the verb root are called *karakas.* There are six basic karakas, namely; *adhikarana* 'location', *apaadaan* 'source', *sampradaan* 'recipient', *karana* 'instrument', *karma* 'theme' and *karta* 'agent' (Begum et al., 2008a). Here the mapping between *karakas* and *theta roles* is a rough mapping.

The *karta karaka* is the locus of the *activity*. Similarly *karma karaka* is the locus of the *result*. The locus of the *activity* implied by the verbal root can be animate or inanimate. Sentence (2) given above, is the example where the locus of the activity is animate. Sentence (5) given below, is the example where the locus of the activity is inanimate.

(5)_*darvaazaa khulaa*
door open.Pst
'Door opened.'

(6) *raam ne darvaazaa kholaa*
ram Erg. door open.Pst
'Ram opened the door.'

(7) *maiM ne raam se darvaazaa*
I Erg. ram by door
khulvaayaa
open.Caus.Pst
'I made Ram open the door.'

When we come to the causatives, the notion of ***prayojak karta*** 'causer', ***prayojya karta*** 'causee' and ***madhyastha karta*** 'mediator causer' are introduced. *prayojak karta* 'causer' is the initiator of the action. *prayojya karta* 'causee' is the one who is made to do the action by the *prayojak karta* 'causer'. *madhyasta karta* 'mediator causer' is the causative agent of the action. The *karta* of the base verb becomes the *prayojya karta* of the causative verb and the *prayojak karta* of the first causative becomes the *madhyasta karta* of the second causative.

This model takes both semantics and morphology into consideration.

4.1.1 Semantics

(8) *caabii ne taalaa kholaa*
key Erg lock open.Pst
'The key opened the lock.'

(9)* *raam ne caabii se taalaa khulvaayaa*
ram Erg. key by lock open.Caus.Pst
'Ram caused the key to open the lock.'

(10)*raam ne mohan dvaaraa caabii se taalaa*
ram Erg. mohan by key with lock
khulvaayaa
open.Caus.Pst.
'Ram made Mohan open the lock with the key.'

In (8), *caabii* 'key' is the *karta* of the transitive verb *khol* 'open'. *caabii* 'key' is an inanimate *karta* so this sentence can't be causativized. (8) has been tried to causativize in (9) which is unacceptable. (9) is actually interpreted as (10) where an inanimate noun with a *se* 'with' postposition acts as an instrument and not as a *prayojya karta* 'causee'. So in (10), *caabii* ‘key’ is an inanimate noun and takes *se* ‘with’ postposition so *caabii se* ‘with the key’ acts as instrument and *mohan* ‘Mohan’ acts as the *prayojya karta* ‘causee’ (Kachru, 1966). It seems that only those verbs can be causativized which take an animate *karta*.

Out of the above two given approaches, we are following approach II where both morphology and semantics are considered. In our approach we are saying that only those base verbs can be causativized which take an animate *karta* and it should also have volitionality (Tripathi, 1986; Reinhart, 2005). Those base verbs which take an inanimate *karta* can’t be causativized. So in our approach the *prayojya karta* ‘causee’ in the causative sentence is always animate as the *karta* of the base verb becomes the *prayojya karta* ‘causee’ in the causative sentences. In our approach we have the notion of *karmakartri* which says an intransitive can be derived out of a basic transitive verb and the *karma* of the basic transitive verb becomes the *karta* of the derived intransitive verb. So the *karta* of the derived intransitive verb is called *karmakartri*. The derived intransitive verbs are like unaccusative verbs of English.

Whereas in approach I, the intransitive base verbs that take both animate and inanimate *karta* can be causativized. But in case of transitives, base verbs which take only animate *karta* can be causativized. Ditransitives can also be causativized. (Kachru 1966; Kachru 1980)

We follow the dependency tagging scheme proposed by Begum et al. (2008a) for the development of a dependency annotation for Indian Languages. In this scheme *prayojak karta* ‘causer’, *prayojya karta* ‘causee’ and *madhyastha karta* ‘mediator causer’ are represented as *pk1, jk1* and *mk1* respectively.

Some of the base verb forms and their causative sentences are given below with dependency relations marked in the brackets for the appropriate arguments:

(11) *raam ne**(k1)** seb**(k2)** khaayaa*
ram Erg. apple eat.Pst
‘Ram ate an apple.’

(12) *siitaa ne**(pk1)** raam ko**(jk1)** seb**(k2)***
sita Erg. ram Acc. apple
khilaayaa
eat.Caus.Pst
‘Sita fed Ram an apple.’

(13)*maa.N ne**(pk1)** siitaa se**(mk1)** raam ko**(jk1)***
mother Erg. sita by ram Acc.
*seb**(k2)** khilvaayaa*
apple eat.Caus.Pst.
‘Mother caused Sita to feed Ram an apple.’

(14) *naukar ne**(k1)** kaam**(k2)** kiyaa*
servant Erg. work do.Pst
‘The servant did the work.’

(15) *maalik ne**(pk1)** naukar se**(mk1)** kaam**(k2)***
master Erg. servant by work
karvaayaa
do.Caus.Pst
‘The master caused Ram to do the job.’

(16) *raam ne**(k1)** siitaa ko**(k4)** kitaab **(k2)** dii*
ram Erg. sita Dat. book give.Pst
‘Ram gave a book to Sita.’

(17)_*mohan ne**(pk1)** raam se**(jk1)** siitaa ko**(k4)***
mohan Erg. ram by sita Dat.
*kitaab**(k2)** dilaaii*
book give.Caus.Pst
‘Mohan made Ram give a book to Sita.’

(18) *mujhko**(k4a)** chaa.Nd**(k1)** dikhaa*
I.Dat. moon appear.Pst
‘The moon became visible to me.’

(19) *maiM ne**(k1)** chaa.Nd**(k2)** dekhaa*
I Erg. moon see.Pst
‘I saw the moon.’

(20) *maiM ne**(pk1)** raam ko**(jk1)** chaa.Nd**(k2)***
mother Erg. ram Dat. moon
dikhaayaa
see.Caus.Pst
‘Mother showed moon to Ram.’

(21) *maiM ne**(pk1)** mohan se**(mk1)***
I Erg. mohan by
*raam ko**(jk1)** chaa.Nd**(k2)** dikhlaayaa*
ram Dat. moon see.Caus.Pst
‘Mother made Mohan show moon to Ram.’

4.1.2 Morphology

In this section we have given the derivation process of the Hindi causative verbs. We have studied 160 Hindi verbs and have come up with certain number of rules for the derivation process of causative verbs.

When causative affixes are added to the base verb roots then some of the base verb roots change in form and some don't. Various causal affixes are added to each verb type to form causatives. An example of affix addition for each verb type is discussed below. The affixes that are added are given in bold. The changes in the base verb root are underlined and made bold in both root form and the causal form.

4.1.2.1 Type-1 and its causative forms:

Suffix **'-aa'** is added to the verb root to form the first causal and **'-vaa'** to form the second causal.

No Change in the Root:

Chip	⟶	*Chip-**aa***	⟶	*Chip-**vaa***
'hide'		'hide'		'cause to hide'

Change in the Root:

- *aa* → *a*

*n**aa**ch*	⟶	*n**a**ch-**aa***	⟶	*n**a**ch-**vaa***
'dance'		'make someone dance'		'cause to make someone dance'

4.1.2.2 Type-2 and its causative forms:

Suffix **'-aa'** is added to the verb root to form the first causal and **'-vaa'** to form the second causal.

No Change in the Root:

likh	⟶	*likh-**aa***	⟶	*likh-**vaa***
'write'		'dictate'		'cause to dictate'

Change in the root:

- *aa, ii* → *i;In addition, 'l' is inserted here between the root and the causative suffix.*

*kh**aa***	⟶	*kh**i-l-aa***	⟶	*kh**i-l-vaa***
'eat'		'feed'		'cause to feed'
*p**ii***	⟶	*p**i-l-aa***	⟶	*p**i-l-vaa***
'drink'		'make someone drink'		'cause to make someone drink'

4.1.2.3 Type-3 and its causative forms:

Suffix **'-vaa'** is added to the verb root to form the first causal.

No Change in the Root:

khariid	⟶	*khariid-**vaa***
'buy'		'cause to buy'

Change in the Root:

- *aa* → *a*

*g**aa***	⟶	*g**a**-**vaa***
'sing'		'cause to sing'

4.1.2.4 Type-4 and its causative forms:

Suffix **'-aa/-vaa'** is added to the verb root to form the first causal.

No Change in the Root:

paros	⟶	*paros-**vaa***
'serve'		'cause to serve'

Change in the root:

- *e* → *i; In addition, 'l' is inserted here between the root and the causative suffix.*

*d**e***	⟶	*d**i-l-aa** /d**i-l-vaa***
'give'		'cause to give'

In case type-5 and type-6 verbs, we can derive intransitive verbs out of transitive verbs. Here we have two types of word formations:

- causative formation,
- Derived intransitive verb formation

Causative derivation is the forward derivation and intransitive derivation is backward derivation.

4.1.2.5 Type-5 and its causative forms:

Suffix **'-aa'** is added to the verb root to form the first causal and **'vaa'** to form the second causal. In this verb type there is no example where the verb root form doesn't change.

Causative Formation: ***Change in the root***

- *e* → *i*

dekh → *dikh-**aa*** → *dikh-**vaa***
'see' 'show' 'cause to show'

Derived intransitive formation: ***Change in the above root*****:**

- *i* ← *e*

dikh ← *dekh*
'appear' 'see'

4.1.2.6 Type-6 and its causative forms:

Suffix **'-aa/-vaa'** is added to the transitive verb root to form the first causal.

Causative formation: ***No change in the root***

bhar → *bhar-**vaa** /bhar-**aa***
'fill' 'cause someone to fill'

Derived intransitive formation: ***No change in the root***

bhar ← *bhar*
'to fill' 'to fill'

Causative formation: Change in the root

- *o* → u; *In addition, 'l' is inserted here between the root and the causative suffix*

*dh**o*** → *dh**u-l-aa** /dh**u-l-vaa***
'wash' 'cause to wash'

Derived intransitive formation: Change in the above root:

- *u* ← o; *In addition, 'l' is inserted at the end of the root*

*dh**u-l*** ← *dh**o***
'be washed' 'cause to wash'

In the implementation of the causative verbs, the causative feature of a verb is reflected in the morph analysis. There are two possible ways to implement causative information: (i) All the causative verb roots are included in the root dictionary of the morph analyzer with an additional feature marking it a causative verb type. (ii) For all causative verbs the following information is marked; causative root, base root, verb type and causative suffix. In (i), the information of base verb root from which the causative root is derived is missing which is captured in (ii). In the above mentioned two ways the latter gives more detailed information than the former.

4.2 Methodology of the Work

For this work, 160 base verbs were taken, their causative forms were given and were classified. Rules for deriving causative verb forms from their base forms were made. Verb frames for base verbs and their causative forms were developed. Based on the analysis of the base verbs certain problem cases were logged and generalizations regarding causativization were made. In this paper, we briefly discuss about all the points mentioned above.

4.3 Classification of Hindi Causative Verbs

Here Hindi verbs have been classified into 6 types based on their causativization behavior:

- **Type-1:** Basic Intransitive verb

Basic intransitive verb has two causal forms i.e., first causal and second causal form. First causal of the basic intransitive verb functions as a transitive verb. The subject of the basic intransitive verb becomes the object of the transitive verb or the first causal form. The subject of the first causal form becomes the causative agent of the second causal form. Sentence (2) is the example of basic intransitive and sentences (3) and (4) are its causative forms.

- **Type-2:** Basic Transitive verb type-I (which is similar to *khaanaa* 'to eat' verb type given by Kachru (1966))

- **Type-3:** Basic Transitive verb type-II (which is similar to *karnaa* 'to eat' verb type given by Kachru (1966))

Type-2 and type-3 are transitive verbs which are divided into two types based on their causativization behavior. Basic transitive verbs of type-I, like *khaanaa* 'to eat' have two causal forms. First causal of *khaanaa* 'to eat' type verb also functions as ditransitive. Whereas transitive verbs of type-II, like *karnaa* ‘to do’ have one causal form. First causal of *karnaa* ‘to eat’ type verb functions as causative. Sentences (11-13) are examples for type-2 verb. Sentences (14-15) are examples for type-3 verb.

- **Type-4:** Basic Ditransitive verb

Ditransitive verbs also have one causal form. Sentences (16-17) are examples for type-4 verb.

- **Type-5:** Basic Transitive verb type-I, out of which intransitive verbs can be derived which takes a dative subject,

- **Type-6:** Basic Transitive verb type-II, out of which intransitive verbs can be derived.

Type-5 and type-6 are transitive verbs which have causal forms depending on whether it is type-I (*khaanaa* ‘to eat’) transitive or type-II (*karnaa* ‘to do’) transitive and in addition both have a derived intransitive form. Type-5 takes a dative subject in the base form. Sentences (18-21) are examples for type-5 verb. Sentences (5-7) are examples for type-6 verb. Other than the 4 classes classified by Kachru (1966), we have two more extra classes, i.e., type-5 and type-6.

An example for each verb type that goes into the classification is given below:

- **Type-1**

Base verb →	**First causal** →	**Second Causal**
so	*sulaa*	*sulvaa*
‘sleep’	‘put to sleep’	‘cause to put to sleep’

- **Type-2**

Base verb →	**First causal** →	**Second Causal**
khaa	*khilaa*	*khilvaa*
‘eat’	‘feed’	‘cause to feed’

- **Type-3**

Base verb →	**First causal**
kar	*karaa/karvaa*
‘do’	‘cause to do’

- **Type-4**

Base verb →	**First causal**
de	*dilaa/dilvaa*
‘give’	‘cause to give’

- **Type-5**

Intransitive ←	**Base verb**
dikh	*dekh*
‘appear’	‘see’

Base verb →	**First causal** →	**Second Causal**
dekh	*dikhaa*	*dikhvaa*
‘see’	‘show’	‘cause to show’

- **Type-6**

Intransitive ←	**Base verb**
khul	*khol*
‘open’	‘open’

Base verb →	**First causal**
khol	*khulvaa*
‘open’	‘cause to open’

5 Verb Frames

In this section we list out the syntactic frames for all the causative types discussed in the previous sections. Verb frame (Begum et al., 2008b) is given for the base form and for its first and second causal form. For ease of exposition, below we show only the relevant information of a verb frame. Components not necessary for the present discussion have been left out. Here the structure of a verb frame is given in terms of dependency relation, postposition (Hindi case marker) and TAM. We have taken past tense (*yaa* is the past tense marker) in the TAM. Refer the examples given above for each type of causatives for a better understanding of the frames.

I.Frame of Type-1 and its Causative Forms:

Relation-Postposition			**TAM**
(a)k1-0			yaa
(b)pk1-ne		jk1-ko	yaa
(c)pk1-ne	mk1-se	jk1- ko	yaa

II.Frame of Type-2 and its Causative Forms:

Relation-Postposition			**TAM**
(a)k1-ne		k2-0	yaa
(b)pk1-ne	jk1-ko	k2-0	yaa

(c)pk1-ne	mk1-se	jk1- ko	k2-0	yaa

III. Frame of Type-3 and its Causative Forms:

Relation-Postposition			**TAM**
(a)k1-ne		k2-0	yaa
(b)pk1-ne	jk1-se	k2-0	yaa

IV. Frame of Type-4 and its Causative Forms:

Relation-Postposition				**TAM**
(a) k1-ne		k4-ko	k2-0	yaa
(b) pk1-ne	jk1-se	k4-ko	k2-0	yaa

V. Frame of Type-5 and its Causative Forms:

Relation-Postposition				**TAM**
(a)k4a-ko		k1		aa
(b)k1-ne			k2-0	aa
(c)pk1-ne		jk1-ko	k2-0	yaa
(d)pk1-ne	mk1-se	jk1-ko	k2-0	yaa

VI. Frame of Type-6 and its Causative Forms:

Relation-Postposition			**TAM**
(a)k1-0			aa
(b)k1-ne		k2-0	aa
(c)pk1-ne	jk1-se	k2-0	yaa

6 Issues and Observations

There are some verbs which can't be causativized. Motion verbs like aa 'come' and jaa 'go' can' t be causativized. After analysing certain amount of corpus we have observed that not all motion verbs behave like the above verbs . aanaa 'to come' and jaanaa 'to go' verbs can't be causativized because these verbs always occur as main verbs and take the following verbs as manner adverbs: chalnaa, bhaagnaa, daudnaa. For instance, chalkar aayaa 'came running' and daudkar gayaa 'went running'. Those motion verbs which occur as manner adverbs and modify another motion verb can be causativized and those verbs which occur as main verbs and never occur as manner adverbs of another motion verb can't be causativized. Natural process verbs like khil 'blossom', garajnaa 'thunder' and ug 'rise' also can't be causativized.

There are three types of the verb *nikal* 'leave'. All the three are used as intransitives.

➢ *derived intransitive*: sense → drain out

(22) *paanii kamre se nikal gayaa*
water room from leave go.Pst.
'Water drained out of the room.'

➢ *Baisc Intransitive:* sense → walked out

(23) *raam kamre se baahar nikal gayaa*
ram room from out leave go.Pst.
'Ram walked out of the room.'

➢ *Baisc Intransitive which involves natural process.*

(24) *gangaa gangotrii se nikaltii*
ganga gangotri from emerge.Impf.
hai
be.Pres.
'Ganga emerges from Gangotri.'

The first type is a derived intransitive which is derived from the base transitive verb *nikaal* 'remove'. This base transitive verb root can be causativized. The second type is basic intransitive which can also be causativized. The third type which is natural process can't be causativized. This shows how important the property of animacy for making causatives is.

7 Conclusion and Future Work

In this work we flesh out the linguistic devices that work for causativization. In this paper we have introduced a preliminary work on Hindi causative verbs. We have given the classification of causative verbs and the linguistic model followed for their classification. We have also given the verb frames of the causative verbs. These insights have been incorporated in the Hindi dependency treebank (Bhatt et al., 2009). We also plan to use the verb frames in a Hindi dependency parser (Bharati et al., 2009) to improve its performance.

References

Agnihotri, Rama K. 2007. *Hindi, An Essential Grammar*. Routledge, London and New York, pp. 121-126.

Balachandran, Lakshmi B. 1973. *A Case Grammar of Hindi*. Central Institute of India, Agra.

Begum, Rafiya, Samar Husain, Arun Dhwaj, Dipti M. Sharma, Lakshmi Bai and Rajeev Sangal. 2008. Dependency Annotation Scheme for Indian Languages. *Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP),* Hyderabad, India.

Begum, Rafiya, Samar Husain, Dipti M. Sharma and Lakshmi Bai. 2008. Developing Verb Frames in Hindi. *In Proceedings of The Sixth International Conference on Language Resources and Evaluation (LREC).* Marrakech, Morocco.

Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing: A Paninian Perspective,* Prentice-Hall of India, New Delhi, pp. 65-106.

Bharati, Akshar, Samar Husain, Dipti Misra Sharma and Rajeev Sangal. 2009. Two stage constraint based hybrid approach to free word order language dependency parsing. *In Proceedings of the 11th International Conference on Parsing Technologies (IWPT09).* Paris.

Bhatt, Rajesh, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Fei Xia. 2009. A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. *In the Third Linguistic Annotation Workshop (The LAW III) in conjunction with ACL/IJCNLP 2009*. Singapore.

Comrie, Bernard. 1981. *Language Universals and Linguistic typology: Syntax and Morphology.* The University of Chicago Press.

Greaves, Edwin. 1983. *Hindi Grammar.* Asian Educational Services, New Delhi, pp. 301-313

Kachru, Yamuna. 1966. *Introduction To Hindi Syntax.* University of Illinois, Illionois, pp. 62-81

Kachru, Yamuna. 1980. *Aspects of Hindi Grammar*. Manohar Publications.

Kachru, Yamuna. 2006. *Hindi*. London Oriental and African Language Libtary.

Kale, M.R. 1960. *A Higher Sanskrit Grammar.* Motilal Banarasidass, Delhi.

Kellogg, S. H. 1972. *A Grammar of the Hindi Language*. Munshiram Manoharlal, New Delhi, pp. 253-257.

McGregor, R. S. 1995. *Outline of Hindi Grammar.* Oxford University Press, pp. 118-125.

Nedyalkov, Vladmir P., Georgij G. Silnitsky. 1973. The typology of Morphological and Lexical Causatives, in: F. Kiefer, (ed.), *Trends in Soviet Theoretical Linguistics,* Foundations of Language, Supplementary Series 18, Dordrecht, 1-32.

Reinhart, Tanya. 2005. *Causativization and Decausativization*: *Unpublished technical report.* LSA 2005.

Sahaay, Chaturbhuj. 2004. *Hindi ke Muul Vaakya Saanche,* Kumar Prakaashan, Agra, pp. 91-100.

Sahaay, Chaturbhuj. 2007. *Hindi Padvigyaan,* Kumar Prakaashan, Agra.

Shastri, Charu D. 1990. *Panini: Re-Interpreted.* Motilal Banarasidass, Delhi.

Singh, Rajendra and Rama K. Agnihotri. 1997. *Hindi Morphology: A Word-based Description.* Motilal Banarasidass, Delhi.

Tripathi, Acharya R. 1986. *Hindi Bhashaanushasan.* Bihar Granth Academy, Patna.

A Supervised Learning based Chunking in Thai using Categorial Grammar

Thepchai Supnithi, **Peerachet Porkaew, Taneth Ruangrajitpakorn, Kanokorn Trakultaweekool**
Human Language Technology,
National Electronics and Computer Technology Center

{thepchai.sup, peerachet.por, taneth.rua, kanokorn.tra}@nectec.or.th

Chanon Onman, Asanee Kawtrakul

Department of Computer Engineering, Kasetsart University and
National Electronics and Computer Technology Center

chanon.onman@gmail.com,
asanee.kaw@nectec.or.th

Abstract

One of the challenging problems in Thai NLP is to manage a problem on a syntactical analysis of a long sentence. This paper applies conditional random field and categorical grammar to develop a chunking method, which can group words into larger unit. Based on the experiment, we found the impressive results. We gain around 74.17% on sentence level chunking. Furthermore we got a more correct parsed tree based on our technique. Around 50% of tree can be added. Finally, we solved the problem on implicit sentential NP which is one of the difficult Thai language processing. 58.65% of sentential NP is correctly detected.

1 Introduction

Recently, many languages applied chunking, or shallow parsing, using supervised learning approaches. Basili (1999) utilized clause boundary recognition for shallow parsing. Osborne (2000) and McCallum et al. (2000) applied Maximum Entropy tagger for chunking. Lafferty (2001) proposed Conditional Random Fields for sequence labeling. CRF can be recognized as a generative model that is able to reach global optimum while other sequential classifiers focus on making the best local decision. Sha and Pereira (2003) compared CRF to other supervised learning in CoNLL task. They achieved results better than other approaches. Molina et al. (2002) improved the accuracy of HMM-based shallow parser by introducing the specialized HMMs.

In Thai language processing, many researches focus on fundamental level of NLP, such as word segmentation, POS tagging. For example, Kruengkrai et al. (2006) introduced CRF for word segmentation and POS tagging trained over Orchid corpus (Sornlertlamvanich et al., 1998.). However, the number of tagged texts in Orchid is specific on a technical report, which is difficult to be applied to other domains such as news, document, etc. Furthermore, very little researches on other fundamental tools, such as chunking, unknown word detection and parser, have been done. Pengphon et al. (2002) analyzed chunks of noun phrase in Thai for information retrieval task. All researches assume that sentence segmentation has been primarily done in corpus. Since Thai has no explicit sentence boundary, defining a concrete concept of sentence break is extremely difficult.

Most sentence segmentation researches concentrate on "space" and apply to Orchid corpus (Meknavin 1987, Pradit 2002). Because of ambiguities on using space, the accuracy is not impressive when we apply into a real application.

Let consider the following paragraph which is a practical usage from news:

Proceedings of the 8th Workshop on Asian Language Resources, pages 129–136,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

"สำหรับการวางกำลังของคนเสื้อแดง ได้มีการวางบังเกอร์โดยรอบพื้นที่ชุมนุม และใช้น้ำมันราด | รวมทั้งมียางรถยนต์ | ขณะการจราจรยังเปิดเป็นปกติ"

lit: "The red shirts have put bunkers around the assembly area and put oil and tires. The traffic is opened normally."

We found that three events are described in this paragraph. We found that both the first and second event do not contain a subject. The third event does not semantically relate to the previous two events. With a literal translation to English, the first and second can be combined into one sentence; however, the third events should be separated.

As we survey in BEST corpus (Kosawat 2009), a ten-million word Thai segmented corpus. It contains twelve genres. The number of word in sentence is varied from one word to 2,633 words and the average word per line is 40.07 words. Considering to a News domain, which is the most practical usage in BEST, we found that the number of words are ranged from one to 415 words, and the average word length in sentence is 53.20. It is obvious that there is a heavy burden load for parser when these long texts are applied.

Example 1:

คน	ขับ	รถแท็กซี่	พบ	กระเป๋าสตางค์
man(n)	drive(v)	taxi(n)	find(v)	wallet(n)

lit1: A man drove a taxi and found a wallet.

lit2: A taxi chauffeur found a wallet.

Example 2:

น่า	จะ	ต้อง	สามารถ	พัฒนา	ประเทศ
should	will	must	can	develop(v)	country(n)

lit: possibly have to develop country.

Figure 1. Examples of compounds in Thai

Two issues are raised in this paper. The first question is "How to separate a long paragraph into a larger unit than word effectively?" We are looking at the possibility of combining words into a larger grain size. It enables the system to understand the complicate structure in Thai as explained in the example. Chunking approach in this paper is closely similar to the work of Sha and Pereira (2003). Second question is "How to analyze the compound noun structure in Thai?"

Thai allows a compound construction for a noun and its structures can be either a sequence of nouns or a combination of nouns and verbs. The second structure is unique since the word order is as same as a word order of a sentence. We call this compound noun structure as a "sentential NP".

Let us exemplify some Thai examples related to compound word and serial construction problem in Figure 1. The example 1 shows a sentence which contains a combination of nouns and verbs. It can be ambiguously represented into two structures. The first alternative is that this sentence shows an evidence of a serial verb construction. The first word serves as a subject of the two following predicates. Another alternative is that the first three word can be formed together as a compound noun and they refer to "a taxi driver" which serve as a subject of the following verb and noun. The second alternative is more commonly used in practical language. However, to set the "N V N" pattern as a noun can be very ambiguous since in the example 1 can be formed a sentential NP from either the first three words or the last three words.

From the Example 2, an auxiliary verb serialization is represented. It is a combination of auxiliary verbs and verb. The word order is shown in Aux Aux Aux Aux V N sequence.

The given examples show complex cases that require chunking to reduce an ambiguity while Thai text is applied into a syntactical analysis such as parsing. Moreover, there is more chance to get a syntactically incorrect result from either rule-based parser or statistical parser with a high amount of word per input.

This paper is organized as follows. Section 2 explains Thai categorial grammar. Section 3 illustrates CRF, which is supervised method applied in this work. Section 4 explains the methodology and experiment framework. Section 5 shows experiments setting and result. Section 6 shows discussion. Conclusion and future work are illustrated in section 7.

2 Linguistic Knowledge

2.1 Categorial Grammar

Categorial grammar (Aka. CG or classical categorial grammar) (Ajdukiewicz, 1935; Bar-Hillel, 1953; Carpenter, 1992; Buszkowski, 1998; Steedman, 2000) is formalism in natural language syntax motivated by the principle of constitutionality and organized according to the syntactic elements. The syntactic elements are categorised in terms of their ability to combine with one another to form larger constituents as functions or according to a function-argument relationship. All syntactic categories in CG are distinguished by a syntactic category identifying them as one of the following two types:

1. Argument: this type is a basic category, such as s (sentence) and np (noun phrase).
2. Functor (or function category): this category type is a combination of argument and operator(s) '/' and '\'. Functor is marked to a complex constituent to assist argument to complete sentence such as s\np (intransitive verb) requires noun phrase from the left side to complete a sentence.

CG captures the same information by associating a functional type or category with all grammatical entities. The notation α/β is a rightward-combining functor over a domain of α into a range of β. The notation α\β is a leftward-combining functor over β into α. α and β are both argument syntactic categories (Hockenmaier and Steedman, 2002; Baldridge and Kruijff, 2003).

The basic concept is to find the core of the combination and replace the grammatical modifier and complement with set of categories based on the same concept with fractions. For example, intransitive verb is needed to combine with a subject to complete a sentence therefore intransitive verb is written as s\np which means it needs a noun phrase from the left side to complete a sentence. If there is a noun phrase exists on the left side, the rule of fraction cancellation is applied as np*s\np = s. With CG, each constituent is annotated with its own syntactic category as its function in text. Currently there are 79 categories in Thai. An example of CG derivation from Thai is shown in Figure 2.

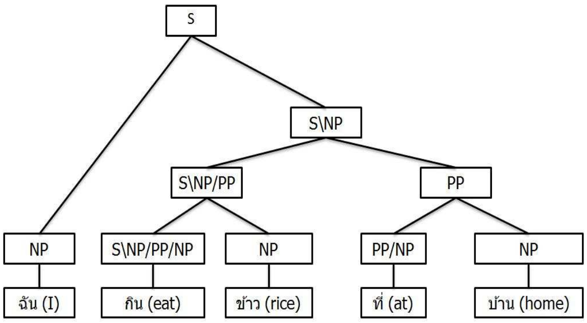


Figure 2 Example of Thai CG-parsed Tree.

2.2 CG-Set

CG-Set are used as a feature when no CG are tagged to the input. We aim to apply our chunker to a real world application. Therefore, in case that we have only sentence without CG tags, we will use CG-Set instead.

Cat-Set Index	Cat-Set	Member
0	np	คุณสมบัติ
2	s\np/pp,s\np/np,s\np/pp/np,s\np	เก็บ, กรอง
3	(np\np)/(np\np), ((s\np)\(s\np))/spnum, np, (np\np)\num,np\num, (np\np)/spnum, ((s\np)\(s\np))\num	วงจร, สัญญาณ
62	(s\np)\(s\np),s\s	มั้ย, มั้ง, ล่ะ
134	np/(s\np), np/((s\np)/np)	การ, ความ

Table 1 Example of CG-Set

The concept of CG-Set is to group words that their all possible CGs are equivalent to the other. Therefore every word will be assigned to only one CG-Set. By using CG-Set we use the lookup table for tagging the input. Table 1 shows examples of CG-set. Currently, there are 183 CG set.

3 Conditional Random Field (CRF)

CRF is an undirected graph model in which each vertex represents a random variable whose distribution is to be inferred, and edge represents a dependency between two random variables. It is a supervised framework for labeling a sequence data such as POS tagging and chunking. Let X is a random variable of observed input sequence, such as sequence of words, and Y is a random variable of label sequence corresponding to X , such as sequence of POS or CG. The most probable label sequence ($\hat{y}$) can be obtain by

$$\hat{y} = \arg\max p(y \mid x)$$

Where $x = x_1, x_2, ..., x_n$ and $y = y_1, y_2, ..., y_n$ $p(y \mid x)$ is the conditional probability distribution of a label sequence given by an input sequence. CRF defines $p(y \mid x)$ as

$$P(y \mid x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{n} F(y, x, i)\right)$$

where $Z = \sum_y \exp\left(\sum_{i=1}^{n} F(y, x, i)\right)$ is a normalization factor over all state sequences. $F(y, x, i)$ is the global feature vector of CRF for sequence x and y at position i . $F(y, x, i)$ can be calculated by using summation of local features.

$$F(y, x, i) = \sum_i \lambda_i f_i(y_{i-1}, y_i, t) + \sum_j \lambda_j g_j(x, y, t)$$

Each local feature consists of transition feature function $f_i(y_{i-1}, y_i, t)$ and per-state feature function $g_j(x, y, t)$. Where λ_i and λ_j are weight vectors of transition feature function and per-state feature function respectively.

The parameter of CRF can be calculated by maximizing the likelihood function on the training data. Viterbi algorithm is normally applied for searching the most suitable output.

4 Methodology

Figure 3 shows the methodology of our experiments. To prepare the training set, we start with our corpus annotated with CG tag. Then, each sentence in the corpus was parsed by

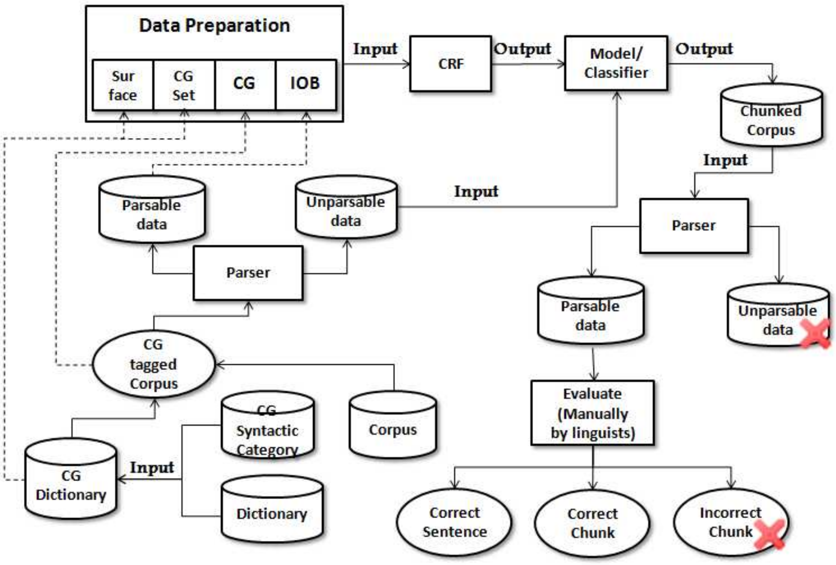


Figure 3 Experimental Framework

our Thai CG parser, developed by GLR techtechnique. However, not all sentences can be parsed successfully due to the complexity of the sentence. We kept parsable sentences and unparsable sentences separately. The parsable sentences were selected to be the training set.

There are four features – surface, CG, CG-set and chunk marker – in our experiments. CRF is applied using 5-fold cross validation over combination of these features. Accuracy in term of averaged precision and recall are reported.

We select the best model from the experiment to implement the chunker. To investigate performance of the chunker, we feed the unparsable sentences to the chunker and evaluate them manually.

After that, the sentences which are correctly chunked will be sent to our Thai CG parser. We calculate the number of successfully-parsed sentences and the number of correct chunks.

5 Experiment Settings and Results

5.1 Experiment on chunking

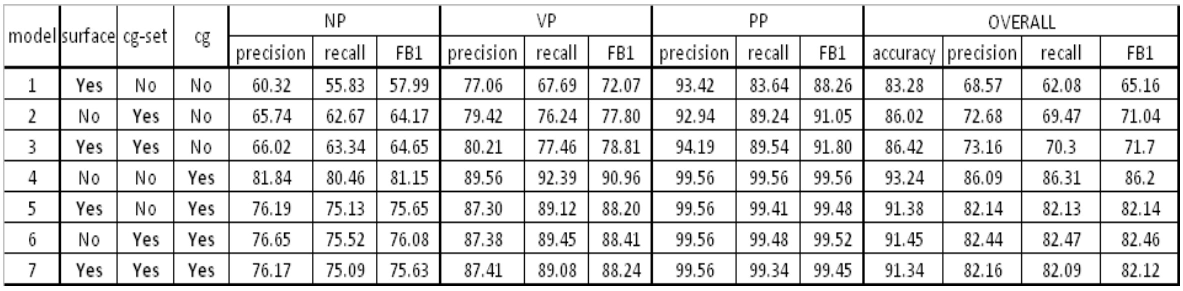
5.1.1 Experiment setting

To develop chunker, we apply CG Dictionary and CG tagged corpus as input. Four features are provided to CRF. Surface is a word surface. CG is a categorial grammar of the word. CG-set is a combination of CG of the word. IOB represents a method to mark chunk in a sentence. "I" means "inner" which represents the word within the chunk. "O" means "outside" which represents the word outside the chunk. "B" means "boundary" which represents the word as a boundary position. It accompanied with five chunk types. "NP" stands for noun phrase, "VP" stands for verb phrase, "PP" stands for preposition phrase, "ADVP" stands for adverb phrase and S-BAR stands for complementizer that link two phrases.

Surface and CG-set are developed from CG dictionary. CG is retrieved from CG tagged corpus. IOB is developed by parsing tree. We apply Thai CG parser to obtain the parsed tree. Figure 4 shows an example of our prepared data. We provide 4,201 sentences as a training data in CRF to obtain a chunked model. In this experiment, we use 5-fold cross validation to evaluation the model in term of F-measure.

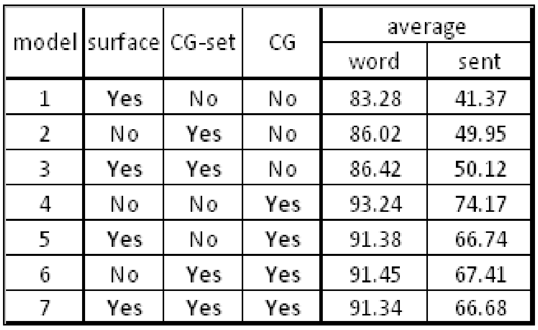
surface	cg_set	cg	chunk_label
ใน	74	s/s/np	B-ADVP
วัน	3	np	I-ADVP
ที่	180	(np\np)/(s\np)	I-ADVP
ไม่	54	(s\np)/(s\np)	I-ADVP
หนาว	7	s\np	I-ADVP
หรือ	130	((s/s)\(s/s))/(s/s)	I-ADVP
ใน	74	s/s/np	I-ADVP
ฤดูร้อน	0	np	I-ADVP
เขา	0	np	B-NP
สวม	8	s\np/np	B-VP
เสื้อ	0	np	B-NP
มา	148	(s\np)/(s\np)	B-VP
เข้าเฝ้า	2	s\np	I-VP

Figure 4 An example of prepared data



model	surface	cg-set	cg	NP			VP			PP			OVERALL			
				precision	recall	FB1	precision	recall	FB1	precision	recall	FB1	accuracy	precision	recall	FB1
1	Yes	No	No	60.32	55.83	57.99	77.06	67.69	72.07	93.42	83.64	88.26	83.28	68.57	62.08	65.16
2	No	Yes	No	65.74	62.67	64.17	79.42	76.24	77.80	92.94	89.24	91.05	86.02	72.68	69.47	71.04
3	Yes	Yes	No	66.02	63.34	64.65	80.21	77.46	78.81	94.19	89.54	91.80	86.42	73.16	70.3	71.7
4	No	No	Yes	81.84	80.46	81.15	89.56	92.39	90.96	99.56	99.56	99.56	93.24	86.09	86.31	86.2
5	Yes	No	Yes	76.19	75.13	75.65	87.30	89.12	88.20	99.56	99.41	99.48	91.38	82.14	82.13	82.14
6	No	Yes	Yes	76.65	75.52	76.08	87.38	89.45	88.41	99.56	99.48	99.52	91.45	82.44	82.47	82.46
7	Yes	Yes	Yes	76.17	75.09	75.63	87.41	89.08	88.24	99.56	99.34	99.45	91.34	82.16	82.09	82.12

Table 2 Chunking accuracy of each chunk



model	surface	CG-set	CG	average	
				word	sent
1	Yes	No	No	83.28	41.37
2	No	Yes	No	86.02	49.95
3	Yes	Yes	No	86.42	50.12
4	No	No	Yes	93.24	74.17
5	Yes	No	Yes	91.38	66.74
6	No	Yes	Yes	91.45	67.41
7	Yes	Yes	Yes	91.34	66.68

Table 3 Chunking accuracy based on word and sentence.

5.1.2 Experiment result

From Table 2, considering on chunk based level, we found that CG gives the best result among surface, CG-set, CG and their combination. The average on three types in terms of F-measure is 86.20. When we analyze information in detail, we found that NP, VP and PP show the same results. Using CG shows the F-measure for each of them, 81.15, 90.96 and 99.56 respectively.

From Table 3, considering in both word level and sentence level, we got the similar results, CG gives the best results. F-measure is 93.24 in word level and 74.17 in sentence level. This shows the evidence that CG plays an important role to improve the accuracy on chunking.

5.2 Experiment on parsing

5.2.1 Experiment setting

We investigate the improvement of parsing considering unparsable sentences. There are 14,885 unparsable sentences from our CG parser. These sentences are inputted in chunked model to obtain a chunked corpus. We manually evaluate the results by linguist. Linguists evaluate the chunked output in three types. 0 means incorrect chunk. 1 means correct chunk and 2 represents a special case for Thai NP, a sentential NP.

5.2.2 Experiment result

From the experiment, we got an impressive result. We found that 11,698 sentences (78.59%) are changed from unparsable to parsable sentence. Only 3,187 (21.41%) are unparsable. We manually evaluate the parsable sentence by randomly select 7,369 sentences. Linguists found 3,689 correct sentences (50.06%). In addition, we investigate the number of parsable chunk calculated from the parsable result and found 37,743 correct chunks from 47,718 chunks (78.47%). We also classified chunk into three types NN VP and PP and gain the accuracy in each type 79.14% ,74.66% and 92.57% respectively.

6 Discussion

6.1 Error analysis

From the experiment results, we found the following errors.

6.1.1 Chunking Type missing

Some chunk missing types are found in experiment results. For example, [PP บันทึก (record)][NP ตัวอักษรได้ประมาณ (character about)]. [PP

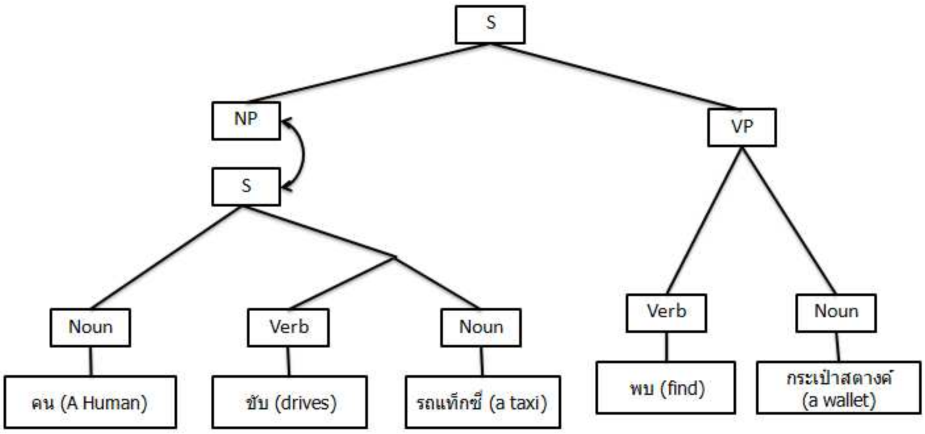


Figure 4 An Example of sentential NP

บันทึก (record)] should be defined as VP instead of PP.

6.1.2 Over-grouping

In the sentence “[VP ใช้ (Using)][NP (medicine)][VP รักษา (treat)][NP โรคแต่ละครั้งต้องเป็นไป (each disease have to)][PP ตาม (follow)][NP คำแนะนำของแพทย์ (doctor’s instruction)] “, we found that “NP โรคแต่ละครั้งต้องเป็นไป (each disease have to) “ has over-grouping. IT is necessary to breakdown to NP โรคแต่ละครั้ง(each disease) and VP ต้องเป็นไป(have to). The reason of this error is due to allow the sentential structure NP VP NP, and then NP and VP are combined.

6.1.3 Sentential NP

We investigated the number of sentential NP. If the number of chunk equal to 1, sentence should not be recognized as NP. Other cases are defined as NP. We found that 929 from 1,584 sentences (58.65 % of sentences) are correct sentential NP. This evidence shows the impressive results to solve implicit NP in Thai. Figure 4 shows an example of sentential NP.

6.1.4 CG-set

Since CG-set is another representation of word and can only detect from CG dictionary. It is very easy to develop a tag sequence using CG-set. We found that CG-set is more powerful than surface. It might be another alternative for less language resource situation.

6.2 The Effect of Linguistic Knowledge on chunking

Since CG is formalism in natural language syntax motivated by the principle of constitutionality and organised according to the syntactic elements, we would like to find out whether linguistic knowledge effects to the model. We grouped 89 categorial grammars into 17 groups, called CG-17. It is categorized into Noun, Prep, Noun Modifier, Number modifier for noun, Number modifier for verb, Number, Clause Marker, Verb with no argument, Verb with 1 argument, Verb with 2 or more arguments, Prefix noun, Prefix predicate, Prefix predicate modifier, Noun linker, Predicate Modification, Predicate linker, and Sentence Modifier.

We found that F-measure is slightly improved from 74.17% to 75.06%. This shows the evidence that if we carefully categorized data based on linguistics viewpoint, it may improve more accuracy.

7 Conclusions and Future Work

In this paper, we stated Thai language problems on the long sentence pattern and find the novel method to chunk sentence into smaller unit, which larger than word. We concluded that using CRF accompanied with categorical grammar show the impressive results. The accuracy of chunking in sentence level is 74.17%. We are possible to collect 50% more on correct tree. This technique enables us to solve the implicit sentential NP problem. With our technique, we found 58% of implicit sentential NP. In the future work, there are several issues to be improved. First, we have to trade-off between over-grouping problem and implicit sentential problem. Second, we plan to consider ADVP, SBAR, which has a very small size of data. It is not adequate to train for a good result. Finally, we plan to apply more linguistics knowledge to assist more accuracy.

References

Abney S., and Tenny C., editors, 1991. *Parsing by chunks, Priciple-based Parsing*. Kluwer Academic Publishers.

Awasthi P., Rao D., Ravindram B., 2006. *Part of Speech Tagging and Chunking with HMM and CRF*, Proceeding of the NLPAI Machine Learning Competition.

Basili R., Pazienza T., and Massio F., 1999. *Lexicalizing a shallow parser*, Proceedings of

Traitement Automatique du Langage Naturel 1999. Corgese, Corsica.

Charoenporn Thatsanee, Sornlertlamvanich Virach, and Isahara Hitoshi. 1997. Building A Large Thai Text Corpus - Part-Of-Speech Tagged Corpus: ORCHID. Proceedings of Natural Language Processing Pacific Rim Symposium.

Kosawat Krit, Boriboon Monthika, Chootrakool Patcharika, Chotimongkol Ananlada, Klaithin Supon, Kongyoung Sarawoot, Kriengket Kanyanut, Phaholphinyo Sitthaa, Purodakananda Sumonmas,Thanakulwarapas Tipraporn, and Wutiwiwatchai Chai. 2009. *BEST 2009: Thai Word Segmentation Software Contest*. The Eigth International Symposium on Natural Language Processing : 83-88.

Kruengkrai C., Sornlertlumvanich V., Isahara H, 2006. *A Conditional Random Field Framework for Thai Morphological Analysis*, Proceedings of 5th International Conference on Language Resources and Evaluation (LREC-2006).

Kudo T., and Matsumoto Y., 2001. *Chunking with support vector machines*, Proceeding of NAACL.

Lafferty J., McCallum A., and Pereira F., 2001. *Conditional Random Fields : Probabilistic models for segmenting and labeling sequence data*. In Proceeding of ICML-01, 282-289.

McCallum A., Freitag D., and Pereira F. 2000. *Maximum entropy markov model for information extraction and segmentation.* Proceedings of ICML.

Molina A., and Pla F., 2002. *Shallow Parsing using Specialized HMMs*, Journal of Machine Learning Research 2,595-613

Nguyen L. Minh, Nguyen H. Thao, and Nguyen P., Thai. 2009. *An Empirical Study of Vietnamese Noun Phrase Chunking with Discriminative Sequence Models*, Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009,9-16

Osborne M. 2000. *Shallow Parsing as Part-of-Speech Tagging.* Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal.

Pengphon N., Kawtrakul A., Suktarachan M., 2002. *Word Formation Approach to Noun Phrase Analysis for Thai*, Proceedings of SNLP2002.

Sha F. and Pereira F., 2003. *Shallow Parsing with Conditional Random Fields*, Proceeding of HLT-NAACL.

A hybrid approach to Urdu verb phrase chunking

Wajid Ali
National University of Computer
and Emergence Sciences,
Pakistan.
wajid.ali@msn.com

Sarmad Hussain
Center for Language Engineering,
Al-Khawarizmi Institute of Computer
Science, University of Engineering
and Technology, Pakistan
sarmad.hussain@kics.edu.pk

Abstract

A variety of verb phrases exist in Urdu including simple verb phrases, conjunct verb phrases and compound verb phrases. This paper explains the structure of Urdu verb phrases, and details a series of experiment to automatically tag them. Initially, a rule based model is developed using 21 linguistic rules for automatic VP chunking. A 100,000 word Urdu corpus is manually tagged with VP chunk tags. The corpus is then used to develop a hybrid approach using HMM based statistical chunking and correction rules. The technique is enhanced by changing chunking direction and merging chunk and POS tags. The automatically chunked data is compared with manually tagged held-out data to identify and analyze the errors. Based on the analysis, correction rules are extracted to address the errors. By applying these rules after statistical tagging, further improvement is achieved in chunking accuracy. The results of all experiments are reported with maximum overall accuracy of 98.44% achieved using hybrid approach with extended tagset.

1 Introduction

Urdu is an Indo-Aryan language, spoken by more than 100 million speakers across the world. It is the national language of Pakistan and state language of India. Urdu has free phrase-order, i.e. the phrases within a sentence can arbitrarily change order[1], but the words within a phrase have a fixed order. As the order of the phrases is variable, the case markers (CM), which are explicitly written in Urdu as separate words, help determine the role of each phrase in a sentence. Verb Phrase (VP) is the head of a sentence and licenses the number as well as role of the other phrases in a sentence, e.g. subject, object, etc. The number of arguments licensed depends on the valency of the verb, also categorized as intransitive, transitive and di-transitive. This information is normally encoded in the subcategorization frame of a verb, which lists the number and type of arguments the verb licenses. Determining these phrases within a sentence is very useful for a variety of applications, and the process which directly labels these phrases is called chunking. Chunking helps to identify phrases in a sentence, which are further used for the development of natural language processing (NLP) applications like parsing, searching, machine translation, question-answering and information extraction. The current work focuses on chunking VP in Urdu.

Relevant Urdu VP analysis is summarized in Section 2. Section 3 presents some relevant chunking related work. Section 4 contains the detail of the tagged corpus developed for this task. Methodology is discussed in Section 5. The results and discussion are presented in Section 6. Section 7 concludes the work.

2 Verb phrases in Urdu

Minimally, an Urdu VP is represented by a single verb. However, a typical Urdu verb phrase contains a verb followed by one or more auxiliary verbs (AUX) and verb tense markers (VBT). Each is represented by a separate word. Some of the tense and aspect information is also encoded within the verb morphology (Hussain 2004). An Urdu verb phrase can be categorized into a simple verb phrase or complex verb phrase. In a complex verb phrase, the verb is formed by a combination of nominal + verb (called conjunct verb) or a verb + verb (called compound verb). These complex verbs are also referred as complex predicates.

[1] Though the meaning does not change, the phrase-order may change the emphasis of the phrase within a sentence.

Proceedings of the 8th Workshop on Asian Language Resources, pages 137–143,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

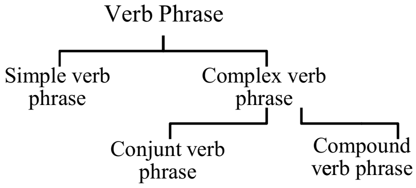


Figure 1: Categorization of Urdu verb phrase

The following subsections explain these types of verb phrases with examples.

2.1 Simple Verb Phrase

A simple verb phrase consists of a verb root followed by auxiliaries and tense verb, if any, as shown in (1).

(1)

علی سکول گیا[2]

Ali school *gaya*
Ali school went
"Ali went to school"

علی سکول جاتا ہے

Ali school *jata hai*
Ali school go *VBT-present*
"Ali goes to the school"

2.2 Complex Verb Phrase

A conjunct verb phrase consists of a nominal or an adjective and a verb, optionally followed by auxiliaries and tense verb, as shown in (2).

(2)

علی نے سبق یاد کیا

Ali ne sabaq *yaad kiya*
Ali *CM* lesson learn do-*past*
"Ali learnt the lesson"

علی نے کمرہ صاف کیا تھا

Ali ne kamraa *saaf kiya tha*
Ali *CM* room clean do-*past VBT-past*
"Ali cleaned the room"

A compound verb phrase consists of two verbs, optionally followed by auxiliaries and verb tense marker. The first verb is main verb and contributes the meaning of the sentence. The second verb adds additional information as shown in (3).

(3)

علی کام کربیٹھا ہے

Ali kam *kar baeTha hai*
Ali work do sit VBT-*present*
"Ali has done the work"

Detailed analysis of Urdu VP is not in the scope of this paper. For further discussion, see e.g. Butt (1995) and Chakrabarti et al. (2008).

3 Earlier work on Chunking

The work on chunking based on machine learning was introduced by Church (1988) for English. Abney (1991) proposed the idea of parsing by chunks, defining the chunks in English by assuming that a chunk has syntactic structure. Chunking was used to convert sentences into non-overlapping phrases, like VP and Noun Phrase (NP), to parse the sentence. Chen (1993) proposed a probabilistic chunker based on Abney (1991). Ramshaw et al. (1995) used transformation based learning using a large annotated corpus for English. They proposed chunking as an IOB tagging task, where I marks the words which are *Inside* a chunk, O marks the words which are *Outside* the chunk and B marks the words which are at the *Beginning* of a chunk. Overall recall and precision achieved by this approach is about 88%.

Zhou et al. (2000) use standard HMM based tagging methods to model the chunking process, and achieved an accuracy of 91.99% precision and 92.25% recall using a contextual lexicon. Veenstra et al. (2000) use memory based phrase chunking with accuracy of 91.05% precision and 92.03% recall for English. Kudo et al. (2001) use support vector machines for chunking with 93.48% accuracy for English. Park et al. (2003) described a hybrid approach using rule based and memory based learning to chunk the phrases of Korean language. First, the rule based chunker is applied to chunk the phrases then memory based learning technique is used for the correction of errors which were not handled by rule based chunker. Grover et al. (2007) proposed rule based chunking using XML. They reported 90.18% precision and 92.49% recall for verb group chunker for English.

Singh et al. (2005) presented HMM based chunk tagger for Hindi. They divided chunk tagging into two main tasks: one was identification of chunk boundaries and the other was labeling of chunks. The Hindi annotated corpus of

[2] Urdu sentences are written from right to left.

200,000 words was used in their work. The data of 150,000 words used to train different HMM representations and 50,000 words data was kept aside as unseen data. The chunker was tested on 20,000 words and chunker achieved 92% precision with 100% recall for chunk boundaries by the HMM based chunker. Dalal et al. (2006) presented a maximum entropy based statistical approach to POS tagger and chunk tagger for Hindi. The model uses multiple features simultaneously to predict the tag for a word. The feature set is broadly classified as context-based features, word features, dictionary features and corpus-based features. The annotated corpus contained almost 35,000 words for training and testing. The reported accuracy was 87.4%. Agarwal et al. (2006) used Conditional Random Field for POS tagging and chunking Hindi text. Various experiments were carried out with various sets and combinations of features to mark a gradual increase in the performance of the system throughout the building process. A data of 21,000 words used for the training. The chunker gave 90.89% accuracy on the data for CONLL 2000.

4 Corpus and Tagset

For the current work, Part of speech (POS) tagged corpus containing 4,585 sentences and 101,414 words is used (from Muaz et al. 2009). Complex phrase is composed of a nominal, adjective or a verb combined with a *light verb.* In the POS tagged corpus used, light verbs are not tagged separately. However, tagging such verbs as light verbs helps determine whether the preceding word is part of verb phrase. So, we customized the tagset by introducing light verb tag (VBL) and infinitive light verb tag (VBLI) to better address the compound and conjunct verb phrases, following Sajjad (2007). The example demonstrates the light verb tag and chunk boundary of complex phrase.

<JJ> صاف <NN> کمرہ <PP> نے <NNP> علی
<VBT> تھا <VBL> کیا

<O><NN>کمرہ <O><PP>نے <O><NNP>علی
I><VBT> تھا <I><VBL> کیا <B><JJ> صاف

Ali ne kamraa *saaf kiya tha*
Ali *CM* room clean do-*past VBT-past*
“Ali cleaned the room”

The IOB tagset is used to prepare chunk annotated data. The data of 3,650 sentences containing 81,430 words is for training, 530 sentences containing 9,985 words are used for analysis during the implementation of methodologies (as held-out data) and the remaining 405 sentences with 9,999 words are used for testing.

5 Methodology

A hybrid approach is used for VP chunking. First, a rule based chunker is developed for baseline. Then HMM based statistical approach is used. Finally, error correction rules are identified for further correction. The methodology is described below.

5.1 Rule Based Chunking

Initially, a set of 21 hand crafted rules are derived, based on experience through manual tagging, for VP chunking. These rules are incrementally built and applied using the training corpus.

5.2 Statistical Chunking

A statistical model for automatic tagging is also developed. Given a sequence of *n* words, there are corresponding t_1 to t_n POS tags and c_1 to c_n chunk tags. The aim is to find the most probable chunk sequence for given the POS tags.

$$\hat{C} = arg \max P(t_1^n/c_1^n) \,.\, P(c_1^n)$$

We assume that the probability of a POS tag depends on its own chunk tag and the probability of a chunk tag is dependent only on the previous two chunk tags. Using chain rule, problem is reduced to the following equation.

$$\hat{C} = arg \max \prod_{i=1}^{n} P(t_i/c_i).P(c_i/c_{i-1}, c_{i-2})$$

TnT tagger (Brants 2000) is used for training and testing, which is based on this model. All the experiments are executed using its default option of second order HMM (trigram model, as presented).

5.3 Error Correction Rules

The statistical tagger is run on the held out data and errors are analyzed to derive rules to fix them as part of the post-processing module. Based on error analysis, twelve rules are identified. Here we discuss some errors and corresponding rules for correction. The complete list of error correction rules is included in the appendix. The most frequent error was assigning I tag to VBT, when it was not preceded by a verb, as it is itself the verb in this case. In this case, it should have been assigned B tag, as it is the beginning of the verb phrase. For example, see the tags underlined below.

<O> <PP> کی <O> <NN>بچوں <O> <JJ>زیرتعلیم
<I> <VBT> ہے <O> <CD>۹۱۹<O> <CD> تعداد

Zair-e-taleem bachoon ki tadaad 919 hai
Under-education children’s number 919 is
“Number of children under education is 919”

The following simple rule makes the needed correction.

- If $POS(w_i) = VBT$ and $POS(w_{i-1})\,! = \{VB, VBI, VBL, VBLI, AUX\}$, then chunk tag for w_i is B.

Another error is to assign O tag to JJ while it precedes the light verb and follows NN. Here it should be the part of the verb phrase.

<O> <NN> جیلیں <O> <PP> کی <O> <NN> بچوں
<I> <VBT> گی <B> <VBL> ہوں <O> <JJ> علیحدہ

Bachon ki jailain alaidha hon gi
Children’s jails separate be *VBT-future*
“Children’s jails will be separate”

The following rule makes the correction.

- If $POS(w_i) = VBL$, $POS(w_{i-1}) = JJ$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.

B tag is assigned to VBL while it follows the CVRP and CVRP follows VB. Here it should be the part of same verb phrase starting from VB.

کر <B> <VB> لکھ <O> <NN> خط <O> <PRP> وہ
<I> <VBT> گیا <B> <VBL> آ <I> <CVRP>

Wo khaat likh ker aa gya
He letter write do come *VBT-past*
“He came after a writing letter”

The following rule makes the correction.

- If$POS(w_i) = VB$, $POS(w_{i-1}) = CVRP$ and $POS(w_{i-2}) = VBL$, Then chunk tag for w_i is I.

One more error is to assign O tag to WALA while it follows VBLI. Here it should be the part of the verb phrase.

والی <I> <VBLI> کرنے <B> <NN> کام
<O><PP> نے <O><NN> خاتون <O><WALA>
<B><VB> بتایا

kaam karnay wali khaton ne batayaa
Work doing WALA[3] women CM told
“Working woman told”

The following rule makes the correction.

- If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for tag for w_i is also I.

5.4 Architecture of VP Chunker

A POS tagged sentence is the input of the VP chunker. The input data is prepared in a specific format and each line contains only a POS tag corresponding to the word in the sentence. TnT Tagger outputs appropriate chunk tag against each POS tag using HMM model. Then post processing is performed on the output of the statistical chunker to enhance the accuracy by applying the error correction rules. Figure 2 shows the architecture of this VP chunker.

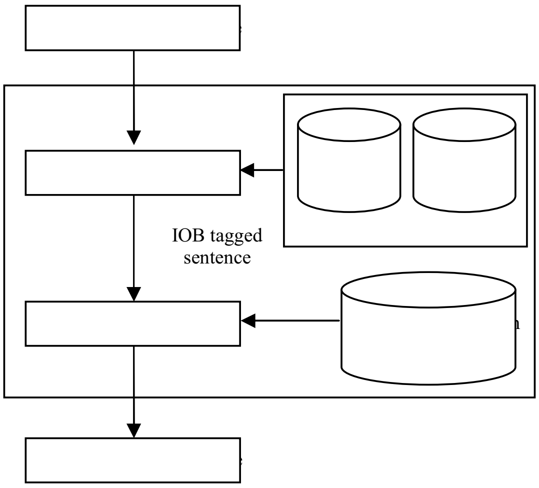


Figure 2: Architecture of VP Chunker

[3] There is no easy translation for WALA. See Muaz and Khan (2009).

5.5 Experiments

VP chunking system is evaluated by conducting a series of experiments. The data is trained and tested using TnT tagger. Three additional factors are used. First, if one scans the sentence in reverse order, one may be able to better predict phrase boundary, as CM comes at the end of a NP. Thus, both Right to Left (default for Urdu) and Left to Right (reverse) directions are explored for scanning and tagging. Second, IOB tagging scheme is further fine-grained by merging it with POS tagset, as an alternate system. Thus B-NN and B-VB are used as different tags, instead of just using B. Third, only statistical vs. hybrid methodologies are used. So, a total nine experiments including rule based model (as baseline) are performed which are listed in Table 1.

Table 1: Scheme for VP chunking experiments

No.	Tagset	Model	Scanning
1.	IOB	Rule Base	Right to left
2.	IOB	Statistical	Right to left
3.	IOB	Hybrid	Right to left
4.	IOB Extended	Statistical	Right to left
5.	IOB Extended	Hybrid	Right to left
6.	IOB	Statistical	Left to right
7.	IOB	Hybrid	Left to right
8.	IOB Extended	Statistical	Left to right
9.	IOB Extended	Hybrid	Left to right

6 Results and Discussion

6.1 Results

There are a total of nine experiments which are performed. First Rule based method is executed on testing data using 21 handcrafted linguistic rules for automatic VP chunking and 93.23% accuracy is achieved. Then statistical experiment is executed on same testing data with simple IOB tagset scheme, and right to left scanning direction. The precision and recall for I, O and B tags are calculated separately. The overall accuracy is 95.14%. By applying error correction rules on this output of statistical chunker, we obtain an overall accuracy of 98.14%. This is given in Table 2 below.

Experiments are also performed using extended tagset by merging IOB tag with POS tag. The overall accuracy of experiment is improved to 95.95%. Error correction rules are applied on output of the statistical chunker, and accuracy is improved to 98.44%.

When the scanning direction is changed to Left to Right, the overall accuracy of statistical approach with simple tagset is 95.06% and 98.02% overall accuracy is obtained using hybrid approach. When extended tagset is used with statistical and hybrid approaches in this scanning direction, the overall accuracy of 95.86% and 98.29% is achieved respectively.

Table 2: Results of VP chunking Experiments

No.	Methodology	Over all result (%)	B-tag		I-tag		O-tag	
			Precision	Recall	Precision	Recall	Precision	Recall
1.	Rule Base (all rules) and RTL scanning	93.23	94.93	56.78	92.64	82.95	99.94	92.99
2.	Statistical using IOB tagset and RTL scanning	95.14	82.52	75.08	88.45	85.17	97.54	99.22
3.	Hybrid using IOB tagset and RTL scanning	98.14	96.21	87.54	92.00	96.62	99.42	99.72
4.	Statistical using Extended tagset and RTL scanning	95.95	83.98	79.13	88.88	86.41	98.27	99.51
5.	Hybrid using Extended tagset and RTL scanning	98.44	97.16	90.07	93.10	96.62	99.45	99.77
6.	Statistical using IOB tagset and LTR scanning	95.06	81.64	74.77	88.20	84.93	97.62	99.22
7.	Hybrid using IOB tagset and LTR scanning	98.02	95.95	86.32	91.57	96.62	99.28	99.69
8.	Statistical using Extended tagset and LTR scanning	95.86	83.24	78.01	88.45	85.67	98.78	99.51
9.	Hybrid using Extended tagset and LTR scanning	98.29	96.80	88.75	91.78	96.62	99.42	99.74

6.2 Discussion

The aim of this research has been to develop an automatic verb phrase chunker for Urdu. To get maximum accuracy different experiments have been conducted using rule base, statistical and hybrid approaches. The intention has been to identify the factors which are important for high accuracy. The experiments show that statistical technique performs better than the rule based system, though the accuracy of the rule based system may be increased further by adding more rules to the repository, which is a tedious process. It is also observed that a few simple error correction rules give a significant 3% improvement in accuracy. Moreover, merging POS tag with IOB tag gives minor improvement in accuracy but reversing scanning direction decreases accuracy.

These results are comparable, even a bit better than the work reported for English. The results are also comparable, perhaps a little better, than Hindi, as reported in the literature, even though Hindi is same as Urdu as spoken. Though the difference in results from English can be attributed to the grammatical differences, it is interesting to note the differences with Hindi. Future work should explore how much of the difference can be attributed to the difference in data used for training, and how much of this difference is caused due to a slight morpho-syntactic difference between the two languages, where in Hindi the case markers are written with the noun as single word in Devanagari script, but are written as separate words from nouns in Urdu using Arabic script.

7 Conclusion

In this paper, we have proposed a hybrid approach to learn verb phrase chunking for Urdu using HMM based statistical chunking and rule based correction afterwards. We have performed different experiments to get maximum accuracy and found the scheme based on hybrid approach with extended tagset and right to left scanning gives the best accuracy of 98.44%.

Appendix

The rules for verb phrase chunking are as following:

1. If $POS(w_i) = VBT$ and $POS(w_{i-1}) != \{VB, VBI, VBL, VBLI, AUX\}$, Then chunk tag for w_i is B.
2. If $POS(w_i) = VB$ and $POS(w_{i-1}) = \{VB, VBI, VBL\}$, Then chunk tag for w_i is I.
3. If$POS(w_i) = VB$, $POS(w_{i-1}) = CVRP$ and $POS(w_{i-2}) = VBL$, Then chunk tag for w_i is I.
4. If $POS(w_i) = VBL$, $POS(w_{i-1}) = JJ$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.
5. If $POS(w_i) = VBL$, $POS(w_{i-1}) = NN$ and $POS(w_{i-2}) = JJ$, Then chunk tag for w_{i-1} is B and w_i is I.
6. If $POS(w_i) = VBLI$, $POS(w_{i-1}) = NNP$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.
7. If$POS(w_i) = VBLI$, $POS(w_{i-1}) = NN$ and $POS(w_{i-2}) = NNP$, Then chunk tag for w_{i-1} is B and w_i is I.
8. If $POS(w_i) = WALA$, $POS(w_{i+1}) = NN$ and $POS(w_{i+2}) = NN$, Then chunk tag for w_{i+1} is I.
9. If$POS(w_i) = WALA$, $POS(w_{i+1}) = JJ$ and $POS(w_{i+2}) = NN$, Then chunk tag for w_{i+1} is I.
10. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBI$, Then chunk tag for w_i is I.
11. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for w_i is I.
12. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for tag for w_i is also I.

References

Abney S. 1991. *Parsing by Chunks: Principle based parsing*. Kluwer Academic Publishers, Dordrecht.

Agarwal H. and Mani A. 2006. *Part of Speech Tagging and Chunking with Conditional Random Fields*. In proceedings of NLPAI Machine Learning Context, Mumbai, India.

Butt M. 1995. *The Structure of Complex Predicates in Urdu.* Stanford, CA: CSLI Publications.

Brant T. 2000. *TnT: a statistical part of speech tager*. In proceeding of the sixth conference on applied natural language processing, Seattle, Washington: 224–231.

Chakrabarti D., Mandalia H., Priya R., Sarma V., and Bhattacharyya P. 2008. *Hindi Compound Verbs and their Automatic Extraction*. Computational Linguistics (COLING08), Manchester, UK.

Chen Kuang-Hua and Chen Hsin-His. 1993. *A Probablistic Chunker*. In proceedings of ROCLING VI.

Church K. 1988. *A stochastic parts program and noun phrase parser for unrestricted text*. In proceedings of Second Conference on Applied Natural Language Processing: 136–143.

Dalal A., Nagaraj K., Sawant U. and Shelke S. 2006. *Hindi Part-of-speech tagging and chunking: A Maximum Entropy Approach*. In proceedings of NLPAI Machine Learning Context, Mumbai, India.

Grover C. and Tobin R. 2007. *Rule Based Chunking and Reusability*. In proceedings of the Fifth international conference on Language Resources.

Hussain, S. 2004. *Finite-State Morphological Analyzer for Urdu.* Unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan. Available at www.crulp.org.

Kudo T. and Matsumoto, Y. 2001. Chunking with Support Vector Machines. Proceedings of NAACL 2001: 1013–1015.

Muaz A., Ali A. and Hussain S. 2009. *Analysis and Development of Urdu POS Tagged Corpora.* In proceedings of the 7th Workshop on Asian Language Resources, IJCNLP'09, Suntec City, Singapore.

Muaz, A., Khan, A. 2009. "The Morphosyntactic Behavior of 'Wala' in Urdu Language", In the *Proceedings of 28th Annual Meeting of the South Asian Language Analysis Roundtable, SALA'09*, University of North Texas, USA. Available at http://www.crulp.org.

Park S. and Zhang B. 2003. *Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning*. In proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1: 497–504.

Ramshaw L. A. and Marcus M. P. 1995. *Text chunking using transformation based learning*. In proceedings of the third ACL workshop on Very Large Corpora, Somerset, NJ: 82–94.

Sajjad, H. 2007. *Statistical Part of Speech Tagger for Urdu.* Unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan. Available at www.crulp.org.

Singh A., Bendre S. and Sangal R. 2005. *HMM Based Chunker for Hindi*. In the Proceedings of IJCNLP-05: The Second International Joint Conference on Natural Language Processing.

Veenstra, J. and van den Bosch, A. 2000. *Single-Classifier Memory-Based Phrase Chunking*. In proceedings of CoNLL-2000 and LLL-2000: 157–159.

Zhou, G., Su, J. and Tey, T. 2000. *Hybrid Text Chunking*. In proceedings of CoNLL- 2000 and LLL-2000: 163–165.

Development of the Korean Resource Grammar: Towards Grammar Customization

Sanghoun Song
Dept. of Linguistics
Univ. of Washington
sanghoun@uw.edu

Jong-Bok Kim
School of English
Kyung Hee Univ.
jongbok@khu.ac.kr

Francis Bond
Linguistics and Multilingual Studies
Nanyang Technological Univ.
bond@ieee.org

Jaehyung Yang
Computer Engineering
Kangnam Univ.
jhyang@kangnam.ac.kr

Abstract

The Korean Resource Grammar (KRG) is a computational open-source grammar of Korean (Kim and Yang, 2003) that has been constructed within the DELPH-IN consortium since 2003. This paper reports the second phase of the KRG development that moves from a phenomena-based approach to grammar customization using the LinGO Grammar Matrix. This new phase of development not only improves the parsing efficiency but also adds generation capacity, which is necessary for many NLP applications.

1 Introduction

The Korean Resource Grammar (KRG) has been under development since 2003 (Kim and Yang, 2003) with the aim of building an open source grammar of Korean. The grammatical framework for the KRG is Head-driven Phrase Structure Grammar (HPSG: (Pollard and Sag, 1994; Sag et al., 2003)), a non-derivational, constraint-based, and surface-oriented grammatical architecture. The grammar models human languages as systems of constraints on typed feature structures. This enables the extension of grammar in a systematic and efficient way, resulting in linguistically precise and theoretically motivated descriptions of languages.

The initial stage of the KRG (hereafter, KRG1) has covered a large part of the Korean grammar with fine-grained analyses of HPSG. However, this version, focusing on linguistic data with theory-oriented approaches, is unable to yield efficient parsing or generation. The additional limit of the KRG1 is its unattested parsing efficiency with a large scale of naturally occurring data, which is a prerequisite to the practical uses of the developed grammar in the area of MT.

Such weak points have motivated us to move the development of KRG to a data-driven approach from a theory-based one upon which the KRG1 is couched. In particular, this second phase of the KRG (henceforth, KRG2) also starts with two methods: shared grammar libraries (the Grammar Matrix (Bender et al., 2002; Bender et al., 2010)) and data-driven expansion (using the Korean portions of multilingual texts).

Next, we introduce the resources we used (§ 2). this is followed by more detailed motivation for our extensions (§ 3). We then detail how we use the grammar libraries from the Grammar Matrix to enable generation (§ 2) and then expand the coverage based on a corpus study (§ 5).

2 Background

2.1 Open Source NLP with HPSG

The Deep Linguistic Processing with HPSG Initiative (DELPH-IN: `www.delph-in.net`) provides an open-source collection of tools and grammars for deep linguistic processing of human language within the HPSG and MRS (Minimal Recursion Semantics (Copestake et al., 2005)) framework. The resources include software packages, such as the LKB for parsing and generation, PET (Callmeier, 2000) for parsing, and a profiling tool [incr_tsdb()] (Oepen, 2001). There are also several grammars: e.g. ERG; the

Proceedings of the 8th Workshop on Asian Language Resources, pages 144–152,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

English Resource Grammar (Flickinger, 2000), Jacy; a Japanese Grammar (Siegel and Bender, 2002), the Spanish grammar, and so forth. These along with some pre-compiled versions of preprocessing or experimental tools are packaged in the LOGON distribution.[1] Most resources are under the MIT license, with some parts under other open licenses such as the LGPL.[2] The KRG has been constructed within this open-source infrastructure, and is released under the MIT license[3].

2.2 The Grammar Matrix

The Grammar Matrix (Bender et al., 2002; Bender et al., 2010) offers a well-structured environment for the development of precision-based grammars. This framework plays a role in building a HPSG/MRS-based grammar in a short time, and improving it continuously. The Grammar Matrix covers quite a few linguistic phenomena constructed from a typological view. There is also a starter-kit, the Grammar Matrix customization system which can build the backbone of a computational grammar from a linguistic description.

2.3 A Data-driven Approach

Normally speaking, building up a computational grammar is painstaking work, because it costs too much time and effort to develop a grammar by hand only. An alternative way is a data-driven approach which ensures ‘cheap, fast, and easy’ development. However, this does not mean that one is better than the other. Each of these two approaches has its own merits. To achieve the best or highest performance of parsing and generation, each needs to complement the other.

3 Directions for Improvement

3.1 Generation for MT

HPSG/MRS-based MT architecture consists of parsing, transfer, and generation, as assumed in Figure 1 (Bond et al., 2005). As noted earlier, the KRG1 with no generation function is limited only to the Source Analysis in Figure 1. In addition, since its first aim was to develop a Korean grammar that reflects its individual properties in detail, the KRG1 lacks compatible semantic representations with other grammars such as ERG and Jacy. The mismatches between the components of the KRG1 and those of other grammars make it difficult to adopt the Korean grammar for an MT system. To take a representative example, the KRG1 treats tense information as a feature type of HEAD, while other grammars incorporate it into the semantics; thus, during the transfer process in Figure 1, some information will be missing. In addition, KRG1 used default inheritance, which makes the grammar more compact, but means it could not used with the faster PET parser. We will discuss this issue in more detail in Section 4.1.

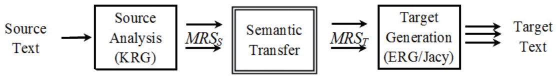


Figure 1: HPSG/MRS-based MT Architecture

Another main issue in the KRG1 is that some of the defined types and rules in the grammar are inefficient in generation. Because the declared types and rules are defined with theoretical motivations, the run-time for generating any parsing units within the system takes more than expected and further causes memory overflow errors to crop up almost invariably, even though the input is quite simple. This problem is partially due to the complex morphological inflection system in the KRG1. Section 4.2 discusses how KRG2, solves this problem.

Third it is better “to be robust for parsing and strict for generation” (Bond et al., 2008). That means robust rules will apply in parsing, though the input sentence does not sound perfect, but not in generation. For example, the sentence (1b), the colloquial form of the formal, standard sentence (1a), is used more frequently in colloquial context:

(1) a. ney-ka cham yeppu-ney.
you-NOM really pretty-DECL
‘You are really pretty.’

b. ni-ka cham ippu-ney

The grammar needs to parse both (1a) and

[1] wiki.delph-in.net/moin/LogonTop

[2] www.opensource.org/licenses/

[3] It allows people “. . . without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies” so long as “The above copyright notice and this permission notice shall be included . . . ”

(1b) and needs to yield the same MRS because both sentences convey the same truth-conditional meaning. However, the KRG1 handles only the legitimate sentence (1a), excluding (1b). The KRG1 is thus not sophisticated enough to distinguish these two stylistic different sentences. Therefore we need to develop the generation procedures that can choose a proper sentence style. Section 4.3 proposes the 'STYLE' feature structure as the choice device.

3.2 Exploiting Corpora

One of the main motivations for our grammar improvement is to achieve more balance between linguistic motivation and practical purpose. We have first evaluated the coverage and performance of the KRG1 using a large size of data to track down the KRG1's problems that may cause parsing inefficiencies and generating clog. In other words, referring to the experimental results, we patterned the problematic parts in the current version. According to the error pattern, on the one hand, we expanded lexicon from occurring texts in our generalization. On the other hand, we fixed the previous rules and sometimes introduced new rules with reference to the occurrence in texts.

3.3 How to Improve

In developing the KRG, we have employed two strategies for improvement; (i) shared grammar libraries and (ii) exploiting large text corpora.

We share grammar libraries with the Grammar Matrix in the grammar (Bender et al., 2002) as the foundation of KRG2. The Grammar Matrix provides types and constraints that assist the grammar in producing well-formed MRS representations. The Grammar Matrix customization system provides with a linguistically-motivated broad coverage grammar for Korean as well as the basis for multilingual grammar engineering. In addition, we exploit naturally occurring texts as the generalization corpus. We chose as our corpora Korean texts that have translations available in English or Japanese, because they can be the baseline of multilingual MT. Since the data-driven approach is influenced by data type, multilingual texts help us make the grammar more suitable for MT in the long term. In developing the grammar in the next phrase, we assumed the following principles:

(2) a. The Grammar Matrix will apply when a judgment about structure (e.g. semantic representation) is needed.

b. The KRG will apply when a judgment about Korean is needed.

c. The resulting grammar has to run on both PET and LKB without any problems.

d. Parsing needs to be accomplished as robustly as possible, and generation needs to be done as strictly as possible.

4 Generation

It is hard to alter the structure of the KRG1 from top to bottom in a relatively short time, mainly because the difficulties arise from converting each grammar module (optimized only for parsing) into something applicable to generation, and further from making the grammar run separately for parsing and generation.

Therefore, we first rebuilt the basic schema of the KRG1 on the Grammar Matrix customization system, and then imported each grammar module from KRG1 to the matrix-based frame (§4.1). In addition, we reformed the inflectional hierarchy assumed in the KRG1, so that the grammar does not impede generation any longer (§ 4.2). Finally, we introduced the STYLE feature structure for sentence choice in accordance with our principles (2c-d) (§4.3).

4.1 Modifying the Modular Structure

The root folder `krg` contains the basic type definition language files (`*.tdl`. In the KRG2, we subdivided the `types.tdl` into: `matrix.tdl` file which corresponds to general principles; `korean.tdl` with language particular rules; `types-lex.tdl` for lexical types and `types-ph.tdl` for phrasal types. In addition, we reorganized the KRG1's `lexicons.tdl` file into the `lex` folder consisting of several sub-files in accordance with the POS values (e.g.; `lex-v.tdl` for verbs).

The next step is to revise grammar modules in order to use the Grammar Matrix to a full extent. In this process, when inconsistencies arise between KRG1 and KRG2, we followed (2a-b).

We further transplanted each previous module into the KRG2, while checking the attested test items used in the KRG1. The test items, consisting of 6,180 grammatical sentences, 118 ungrammatical sentences, were divided into subgroups according to the related phenomena (e.g. light verb constructions).

4.2 Simplifying the Inflectional Hierarchy

Korean has rigid ordering restrictions in the morphological paradigm for verbs, as shown in (3).

(3) a. V-base + HON + TNS + MOOD + COMP

b. ka-si-ess-ta-ko 'go-HON-PST-DC-COMP'

KRG1 dealt with this ordering of suffixes by using a type hierarchy that represents a chain of inflectional slots (Figure 2: Kim and Yang (2004)).

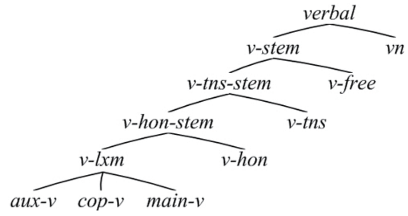


Figure 2: Korean Verbal Hierarchy

This hierarchy has its own merits, but it is not so effective for generating sentences. This is because the hierarchy requires a large number of calculations in the generation process. Figure 3 and Table 1 explains the difference in computational complexity according to each structure.In Figure 3, (a) is similar to Figure 2, while (b) is on the traditional template approach. Let us compare each complexity to get the target node D. For convenience' sake, let us assume that each node has ten constraints to be satisfied. In (a), since there are three parents nodes (i.e. A, B, and C) on top of D, D cannot be generated until A, B, and C are checked previously. Hence, it costs at least 10,000 (10[A] ×10[B] ×10[C] ×10[D]) calculations. In contrast, in (b), only 100 (10[A] ×10[D]) calculations is enough to generate node D. That means, the deeper the hierarchy is, the more the complexity increases. Table 1 shows (a) requires more than 52 times as much complexity as (b), though they have the same number of nodes.

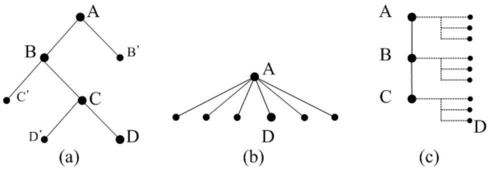


Figure 3: Calculating Complexity

Table 1: Complexity of (a) and (b)

	(a)		(b)
B'	10[A]×10[B']	100	10[A]×10[B']
C'	10[A]×10[B]×10[C']	1,000	10[A]×10[C']
D'	10[A]×10[B]×10[C]×10[D']	10,000	10[A]×10[D']
D	10[A]×10[B]×10[C]×10[D]	10,000	10[A]×10[D]
Σ		21,100	400

When generation is processed by LKB, all potential inflectional nodes are made before syntactic configurations according to the given MRS. Thus, if the hierarchy becomes deeper and contains more nodes, complexity of (a)-styled hierarchy grows almost by geometric progression. This makes generation virtually impossible, causing memory overflow errors to the generation within the KRG1.

A fully flat structure (b) is not always superior to (a). First of all, the flat approach ignores the fact that Korean is an agglutinative language. Korean morphological paradigm can yield a wide variety of forms; therefore, to enumerate all potential forms is not only undesirable but also even impossible.

The KRG2 thus follows a hybrid approach (c) that takes each advantage of (a) and (b). (c) is more flattened than (a), which lessens computational complexity. On the other hand, in (c), the depth of the inflectional hierarchy is fixed as two, and the skeleton looks like a unary form, though each major node (marked as a bigger circle) has its own subtypes (marked as dotted lines). Even though the depth has been diminished, the hierarchy is not a perfectly flat structure; therefore, it can partially represent the austere suffix ordering in Korean. The hierarchy (c), hereby, curtails the cost of generation.

In this context, we sought to use the minimum number of possible inflectional slots for Korean. We need at least three: root + semantic slot(s) + syntactic slot(s). That is, a series of suffixes

Table 2: Complexity of (a-c)

	Depth	Complexity
(a)	n ≥ 3	≥ 10,000
(b)	n = 1	100
(c)	n = 2	10,000

that denote semantic information attaches to the second slot, and a series of suffixes, likewise, attaches to the third slot. Since semantic suffixes are, almost invariably, followed by syntactic ones in Korean, this ordering is convincing, granting that it does not fully represent that there is also an ordering among semantic forms or syntactic ones. (4) is an example from hierarchy (c). There are three slots; root *ka* 'go', semantic suffixes *si-ess*, and syntactic ones *ta-ko*.

(4) a. V-base + (HON+TNS) + (MOOD+COMP)
b. ka-si+ess-ta+ko 'go-HON+PST-DC+COMP'

Assuming there are ten constraints on each node, the complexity to generate D in (c) is just 10,000. The measure, of course, is bigger than that of (b), but the number never increases any more. That means, all forms at the same depth have equal complexity, and it is fully predictable. Table 2 compares the complexity from (a) to (c). By converting (a) to (c), we made it possible to generate with KRG2.

4.3 Choosing a Sentence Style

The choice between formal or informal (colloquial) sentence styles depends on context. A robust parser should cover both styles, but we generally want a consistent style when generating.

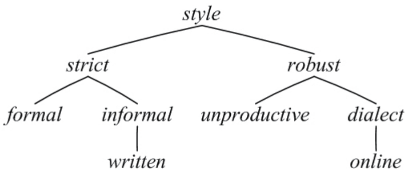


Figure 4: Type Hierarchy of STYLE

In such a case, the grammar resorts to STYLE to filter out the infelicitous results. The type hierarchy is sketched out in Figure 4. *strict* is near to school grammar (e.g. *written* is a style of newspapers). On the other hand, some variant forms that stem from the corresponding canonical forms falls under *robust* in Figure 4. For instance, if the text domain for generation is newspaper, we can select only *written* as our sentence choice, which excludes other styled sentences from our result.

Let us see (1a-b) again. *ni* 'you' in (1b) is a dialect form of *ney*, but it has been used more productively than its canonical form in daily speech. In that case, we can specify STYLE of *ni* as *dialect* as given below. In contrast, the neutral form *ney* has an unspecified STYLE feature:

```
ni := n-pn-2nd-non-pl &
[ STEM < ''ni'' >, STYLE dialect ].
ney := n-pn-2nd-non-pl &
[ STEM < ''ney'' > ].
```

Likewise, since the predicate in (1b) *ippu* 'pretty' stems from *yeppu* in (1a), they share the predicate name '*_yeppu_a_1_rel*' (i.e. the RMRS standard for predicate names such as '_lemma_pos_sense_rel'), but differ in each STYLE feature. That means (1a-b) share the same MRS structure (given below). KRG hereby can parse (1b) into the same MRS as (1a) and generate (1a) from it.

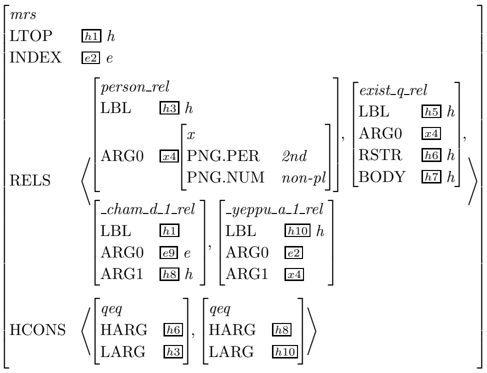


Figure 5: MRS of (1a-b)

These kinds of stylistic differences can take place at the level of (i) lexicon, (ii) morphological combination, and (iii) syntactic configuration. The KRG2 revised each rule with reference to its style type; therefore, we obtained totally 96 robust rules. As a welcoming result, we could manipulate our generation, which was successful respect to (2c-d). Let us call the version reconstructed so far '**base**'.

5 Exploiting Corpora

5.1 Resources

This study uses two multilingual corpora; one is the *Sejong Bilingual Corpora*: SBC (Kim and Cho, 2001), and the other is the *Basic Travel Expression Corpus*: BTEC (Kikui et al., 2003). We exploited the Korean parts in each corpus, taking them as our generalization corpus. Table 3 represents the configuration of two resources (KoEn: Korean-English, KoJa: Korean-Japanese):

Table 3: Generalization Corpora

	SBC	**BTEC**
Type	Bilingual	Multilingual
Domain	Balanced Corpus	Tourism
Words	KoEn : 243,788 KoJa : 276.152	914,199
T/T ratio	KoEn : 27.63 KoJa : 20.28	92.7
Avr length	KoEn : 16.30 KoJa : 23.30	8.46

We also make use of nine test suites sorted by three types (Each test suite includes 500 sentences). As the first type, we used three test sets covering overall sentence structures in Korean; Korean Phrase Structure Grammar (**kpsg**; Kim (2004)), Information-based Korean Grammar (**ibkg**; Chang (1995)), and the SERI test set (**seri**; Sung and Jang (1997)).

Second, we randomly extracted sentences from each corpus, separately from our generalization corpus; two suites were taken from the Korean-English and Korean-Japanese pair in SBC (**sj-ke** and **sj-kj**, respectively). The other two suites are from the BTEC-KTEXT (**b-k**), and the BTEC-CSTAR (**b-c**); the former consists of relatively plain sentences, while the latter is composed of spoken ones.

Third, we obtained two test suites from sample sentences in two dictionaries; Korean-English (**dic-ke**), and Korean-Japanese (**dic-kj**). These suites assume to have at least two advantages with respect to our evaluation; (i) the sentence length is longer than that of BTEC as well as shorter than that of SBC, (ii) the sample sentences on dictionaries are normally made up of useful expressions for translation.

5.2 Methods

We tried to do experiments and improve the KRG, following the three steps repeatedly: (i) evaluating, (ii) identifying, and (iii) exploiting. In each of the first step, we tried to parse the nine test suites and generate sentences with the MRS structures obtained from the parsing results, and measured their coverage and performance. Here, 'coverage' means how many sentences can be parsed or generated, and 'performance' represents how many seconds it takes on average. In the second step, we identified the most serious problems. In the third step, we sought to exploit our generalization corpora in order to remedy the drawbacks. After that, we repeated the procedures until we obtain the desired results.

5.3 Experiments

So far, we have got two versions; **KRG1** and **base**. Our further experiments consist of four phases; **lex**, **MRS**, **irules**, and **KRG2**.

Expanding the lexicon: To begin with, in order to broaden our coverage, we expanded our lexical entries with reference to our generalization corpus and previous literature. Verbal items are taken from Song (2007) and Song and Choe (2008), which classify argument structures of Korean verbal lexicon into subtypes within the HPSG framework in a semi-automatic way. The reason why we do not use our corpus here is that verbal lexicon commonly requires subcategorization frames, but we cannot induce them so easily only using corpora. For other word classes, we extracted lexical items from the POS tagged SBC and BTEC corpora. Table 4 explains how many items we extracted from our generalization corpus. Let us call this version '**lex**'.

Table 4: Expansion of Lexical Items

verbal nouns	4,474
verbs and adjectives	1,216
common nouns	11,752
proper nouns	7,799
adverbs	1,757
numeral words	1,172

MRS: Generation in LKB, as shown in Figure 1, deploys MRS as the input, which means our generation performance hinges on the well-formedness of MRS. In other words, if our MRS is broken somewhere or constructed inefficiently, generation results is directly affected. For instance, if the semantic representation does not scope, we will not generate correctly. We were able to identify such sentences by parsing the corpora, storing the semantic representations and then using the semantic well formedness checkers in the LKB. We identified all rules and lexical items that produced ill-formed MRSs using a small script and fixed them by hand. This had an immediate and positive effect on coverage as well as performance in generation. We refer to these changes as '**MRS**'.

Different inflectional forms for sentence styles: Texts in our daily life are actually composed of various styles. For example, spoken forms are normally more or less different from written ones. The difference between them in Korean is so big that the current version of KRG can hardly parse spoken forms. Besides, Korean has lots of compound nouns and derived words. Therefore, we included these forms into our inflectional rules and expanded lexical entries again (3,860 compound nouns, 2,791 derived words). This greatly increased parsing coverage. We call this version '**irules**'.

Grammaticalized and Lexicalized Forms: There are still remaining problems, because our test suites contain some considerable forms. First, Korean has quite a few grammaticalized forms; for instance, *kupwun* is composed of a definite determiner *ku* and a classifier for human *pwun* "the person", but it functions like a single word (i.e. a third singular personal pronoun). In a similar vein, there are not a few lexicalized forms as well; for example, a verbal lexeme *kkamek-* is composed of *kka-* "peel" and *mek-* "eat", but it conveys a sense of "forget", rather than "peel and eat". In addition, we also need to cover idiomatic expressions (e.g. "thanks") for robust parsing. Exploiting our corpus, we added 1,720 grammaticalized or lexicalized forms and 352 idioms. Now, we call this '**KRG2**'.

Table 5 compares KRG2 with KRG1, and Figure 6 shows how many lexical items we have covered so far.

Table 5: Comparison between KRG1 and KRG2

	KRG1	KRG2
# of default types	121	159
# of lexical types	289	593
# of phrasal types	58	106
# of inflectional rules	86	244
# of syntactic rules	36	96
# of lexicon	2,297	39,190

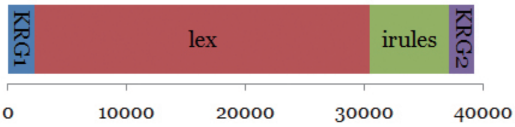


Figure 6: Size of Lexicon

5.4 Evaluation

Table 6 shows the evaluation measure of this study. 'p' and 'g' stand for 'parsing' and 'generation', respectively. '+' represents the difference compared to KRG1. Since KRG1 does not generate, there is no 'g+'.

Table 6: Evaluation

	coverage (%)				ambiguity	
	p	p+	g	s	p	g
kpsg	77.0	-5.5	55.2	42.5	174.9	144.4
ibkg	61.2	41.8	68.3	41.8	990.5	303.5
seri	71.3	-0.8	65.7	46.8	289.1	128.4
b-k	43.0	32.6	62.8	27.0	1769.4	90.0
b-c	52.2	45.8	59.4	31.0	1175.8	160.6
sj-ke	35.4	31.2	58.2	20.6	358.3	170.3
sj-kj	23.0	19.6	52.2	12.0	585.9	294.9
dic-ke	40.4	31.0	42.6	17.2	1392.7	215.9
dic-kj	34.8	25.2	67.8	23.6	789.3	277.9
avr	**48.7**	**24.5**	**59.1**	**28.8**	**836.2**	**198.4**

On average, the parsing coverage increases **24.5%**. The reason why there are negative values in 'p+' of **kpsg** and **seri** is that we discarded some modules that run counter efficient processing (e.g., the grammar module for handling floating quantifiers sometimes produces too many ambiguities.). Since KRG1 has been constructed largely around the test sets, we expected it to perform well here. If we measure the parsing coverage again, after excluding the results of **kpsg** and **seri**, it accounts for **32.5%**.[4] The generation coverage of KRG2 accounts for almost **60%** per parsed sentence on average. Note that KRG1 could not parse at all. 's' (short for 'success') means the portion of both parsed and generated sentences (i.e. 'p'×'g'), which accounts

[4] The running times, meanwhile, becomes slower as we would expect for a grammar with greater coverage. However, we can make up for it using the PET parser, as shown in Figure 9.

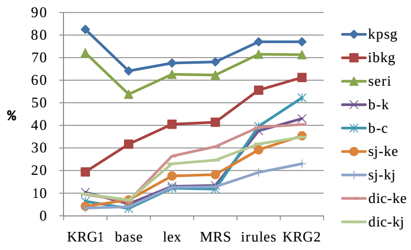


Figure 7: Parsing Coverage (%)

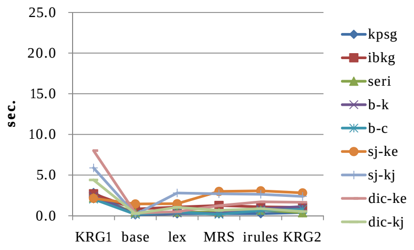


Figure 9: Parsing Performance (s)

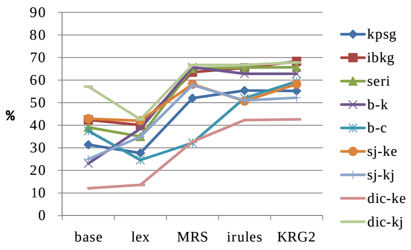


Figure 8: Generation Coverage (%)

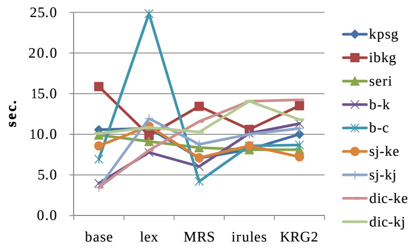


Figure 10: Generation Performance (s)

for about **29%**. Ambiguity means '# of parses/# of sentences' for parsing and '# of realizations/# of MRSes' for generation. The numbers look rather big, which should be narrowed down in our future study.

In addition, we can find out in Table 6 that there is a coverage ordering with respect to the type of test suites; 'test sets > BTEC > dic > SBC'. It is influenced by three factors; (i) lexical variety, (ii) sentence length, and (iii) text domain. This difference implies that it is highly necessary to use variegated texts in order to improve grammar in a comprehensive way.

Figure 7 to 10 represent how much each experiment in §5.3 contributes to improvement. First, let us see Figure 7 and 8. As we anticipated, **lex** and **irules** contribute greatly to the growth of parsing coverage. In particular, the line of **b-c** in Figure 8, which mostly consists of spoken forms, rises rapidly in **irules** and **KRG2**. That implies Korean parsing largely depends on richness of lexical rules. On the other hand, as we also expected, **MRS** makes a great contribution to generation coverage (Figure 8). In **MRS**, the growth accounts for **22%** on average. That implies testing with large corpora must take precedence in order for coverage to grow.

Figure 9 and 10 shows performance in parsing and generation, respectively. Comparing to **KRG1**, our Matrix-based grammars (from **base** to **KRG2**) yields fairly good performance. It is mainly because we deployed the PET parser that runs fast, whereas KRG1 runs only on LKB. Figure 10, on the other hand, shows that the revision of MRS also does much to enhance generation performance, in common with coverage mentioned before. It decreases the running times by about **3.1** seconds on average.

6 Conclusion

The newly developed KRG2 has been successfully included in the LOGON repository since July, 2009; thus, it is readily available. In future research, we plan to apply the grammar in an MT system (for which we already have a prototype). In order to achieve this goal, we need to construct multilingual treebanks; Korean (KRG), English (ERG), and Japanese (Jacy).

Acknowledgments

We thank Emily M. Bender, Dan Flickinger, Jae-Woong Choe, Kiyotaka Uchimoto, Eric Nichols, Darren Scott Appling, and Stephan Oepen for comments and suggestions at various stages. Parts of this research was conducted while the first and third authors were at the National Institute for Information and Communications Technologies (NICT), Japan; we thank NICT for their support. Our thanks also go to three anonymous reviewers for helpful feedback.

References

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Michael Wayne Goodman, Daniel P. Mills, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar Prototyping and Testing with the LinGO Grammar Matrix Customization System. In *Proceedings of ACL 2010 Software Demonstrations*.

Bond, Francis, Stephan Oepen, Melanie Siegel, Ann Copestake, and Dan Flickinger. 2005. Open Source Machine Translation with DELPH-IN. In *Proceedings of Open-Source Machine Translation: Workshop at MT Summit X*.

Bond, Francis, Eric Nichols, Darren Scott Appling, and Michael Paul. 2008. Improving Statistical Machine Translation by Paraphrasing the Training Data. In *Proceedings of the 5th International Workshop on Spoken Languaeg Translation*.

Callmeier, Ulrich. 2000. PET–a Platform for Experimentation with Efficient HPSG Processing Techniques. *Natural Language Engineering*, 6(1):99–107.

Chang, Suk-Jin. 1995. *Information-based Korean Grammar*. Hanshin Publishing, Seoul.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(4):281–332.

Flickinger, Dan. 2000. On Building a More Efficient Grammar by Exploiting Types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28.

Kikui, Genichiro, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. 2003. Creating corpora for speech-to-speech translation. In *Proc. of the EUROSPEECH03*, pages 381–384, Geneve, Switzerland.

Kim, Se-jung and Nam-ho Cho. 2001. The progress and prospect of the 21st century Sejong project. In *ICCPOL-2001*, pages 9–12, Seoul.

Kim, Jong-Bok and Jaehyung Yang. 2003. Korean Phrase Structure Grammar and Its Implementations into the LKB System. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*.

Kim, Jong-Bok and Jaehyung Yang. 2004. Projections from Morphology to Syntax in the Korean Resource Grammar: Implementing Typed Feature Structures. In *Lecture Notes in Computer Science*, volume 2945, pages 13–24. Springer-Verlag.

Kim, Jong-Bok. 2004. *Korean Phrase Structure Grammar*. Hankwuk Publishing, Seoul.

Oepen, Stephan. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, IL.

Sag, Ivan A., Thomas Wasow, , and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA.

Siegel, Melanie and Emily M. Bender. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*.

Song, Sanghoun and Jae-Woong Choe. 2008. Automatic Construction of Korean Verbal Type Hierarchy using Treebank. In *Proceedings of HPSG2008*.

Song, Sanghoun. 2007. A Constraint-based Analysis of Passive Constructions in Korean. Master's thesis, Korea University, Department of Linguistics.

Sung, Won-Kyung and Myung-Gil Jang. 1997. SERI Test Suites '95. In *Proceedings of the Conference on Hanguel and Korean Language Information Processing*.

An Open Source Urdu Resource Grammar

Shafqat M Virk
Department of Applied IT
University of Gothenburg
virk@chalmers.se

Muhammad Humayoun
Laboratory of Mathmatics
University of Savoie
mhuma@univ-savoie.fr

Aarne Ranta
Department of CS & Eng
University of Gothenburg
aarne@chalmers.se

Abstract

We develop a grammar for Urdu in Grammatical Framework (GF). GF is a programming language for defining multilingual grammar applications. GF resource grammar library currently supports 16 languages. These grammars follow an Interlingua approach and consist of morphology and syntax modules that cover a wide range of features of a language. In this paper we explore different syntactic features of the Urdu language, and show how to fit them in the multilingual framework of GF. We also discuss how we cover some of the distinguishing features of Urdu such as, ergativity in verb agreement (see Sec 4.2). The main purpose of GF resource grammar library is to provide an easy way to write natural language applications without knowing the details of syntax, morphology and lexicon. To demonstrate it, we use Urdu resource grammar to add support for Urdu in the work reported in (Angelov and Ranta, 2010) which is an implementation of Attempto (Attempto 2008) in GF.

1. Introduction

Urdu is an Indo-European language of the Indo-Aryan family, widely spoken in south Asia. It is a national language of Pakistan and one of the official languages of India. It is written in a modified Perso-Arabic script from right to left. As regards vocabulary, it has a strong influence of Arabic and Persian along with some borrowing from Turkish and English. Urdu is an SOV language having fairly free word order. It is closely related to Hindi as both originated from the dialect of Delhi region called khari boli (Masica, 1991).

We develop a grammar for Urdu that addresses problems related to automated text translation using an Interlingua approach and provide a way to precisely translate text. This is described in Section 2. Then we describe different levels of grammar development including morphology (Section 3) and syntax (Section 4). In Section 6, we discuss an application in which a semantics-driven translation system is built upon these components.

2. GF (Grammatical Framework)

GF (Grammatical Framework, Ranta 2004) is a tool for working with grammars, implementing a programming language for writing grammars which in term is based on a mathematical theory about languages and grammars[1]. Many multilingual dialog and text generation applications have been built using GF. GF grammars have two levels *the abstract* and *the concrete syntax*[2]. The abstract syntax is language independent and is common to all languages in GF grammar library. It is based on common syntactic or semantic constructions, which work for all the involved languages on an appropriate level of abstraction. The concrete syntax is language dependent and defines a mapping from abstract to actual textual representation in a specific language[2]. GF uses the term ‘category’ to model different parts of speech (e.g verbs, nouns adjectives etc.). An abstract syntax defines a set of categories, as well as a set of tree building functions. Concrete syntax contains rules telling how these trees are linearized. Separating the tree building rules (abstract syntax) from linearization rules (concrete syntax) makes it possible to have multiple concrete syntaxes for one abstract. This

[1] http://www.grammaticalframework.org

[2] In given example code ’fun’ and ’cat’ belongs to abstract syntax, ’lin’ and ’lincat’ belongs to concrete syntax

Proceedings of the 8th Workshop on Asian Language Resources, pages 153–160,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

makes it possible to parse text in one language and translate it to multiple languages.

Grammars in GF can be roughly classified into two kinds: resource grammars and application grammars. Resource grammars are general purpose grammars (Ranta, 2009a) that try to cover the general aspects of a language linguistically and whose abstract syntax encodes syntactic structures. Application grammars, on the other hand, encode semantic structures, but in order to be accurate they are typically limited to specific domains. However, they are not written from scratch for each domain, but they use resource grammars as libraries (Ranta 2009b).

Previously GF had resource grammars for 16 languages: English, Italian, Spanish, French, Catalan, Swedish, Norwegian, Danish, Finish, Russian, Bulgarian, German, Interlingua (an artificial language), Polish, Romanian and Dutch. Most of these languages are European languages. We developed resource grammar for Urdu making it the 17th in total and the first south Asian language. Resource grammars for several other languages (e.g. Arabic, Turkish, Persian, Maltese and Swahili) are under construction.

3. Morphology

In GF resource grammars a test lexicon of 350 words is provided for each language. These words are built through lexical functions. The rules for defining Urdu morphology are borrowed from (Humayoun et el., 2006), in which Urdu morphology was developed in the Functional Morphology toolkit (Forsberg and Ranta, 2004). Although it is possible to automatically generate equivalent GF code from it, we wrote the rules of morphology from scratch in GF, to receive better abstractions than are possible in generated code. Furthermore, we extend this work by including compound words. However, the details of morphology are beyond the scope of this paper, and its focus is on syntax.

4. Syntax

While morphological analysis deals with the formation and inflection of individual words, syntax shows how these words (parts of speech) are grouped together to build well formed phrases. In this section we show how this works and is implemented for Urdu.

4.1 Noun Phrases (NP)

When nouns are to be used in sentences as part of speech, then there are several linguistic details which need to be considered. For example other words can modify a noun, and nouns have characteristics such as gender, number etc. When all such required details are grouped together with the noun, the resulting structure is known as noun phrase (NP). The basic structure of Urdu noun phrase is, "(M) H (M)" according to (Butt M., 1995), where (M) is a modifier and (H) is the head of a NP. Head is the word which is compulsory and modifiers can or cannot be there. In Urdu modifiers are of two types pre-modifiers i.e modifiers that come before the head for instance (کالی بلی kali: bli: "black cat"), and post-modifiers which come after the head for instance (تم سب tm sb "you all"). In GF resource library we represent NP as a record

```
lincat NP : Type = {s : NPCase => Str ; a : Agr} ;
```

where

```
NPCase = NPC Case | NPErg | NPAbl
         |NPIns|NPLoc1NPLoc2
         |NPDat;|NPAcc
Case = Dir | Obl | Voc ;
Agr = Ag Gender Number UPerson ;
Gender = Masc | Fem ;
UPerson = Pers1| Pers2_Casual
           | Pers2_Familiar | Pers2_Respect
           | Pers3_Near | Pers3_Distant;
Number = Sg | Pl ;
```

Thus NP is a record with two fields, 's' and 'a'. 's' is an inflection table and stores different forms of a noun.

The Urdu NP has a system of syntactic cases which is partly different from the morphological cases of the category noun (N). The case markers that follow nouns in the form of post-positions cannot be handled at lexical level through morphological suffixes and are thus handled at syntactic level (Butt et el., 2002). Here we create different forms of a noun phrase to handle case markers for Urdu nouns. Here is a short description of the different cases of NP :

- NPC Case: this is used to retain the original case of Noun
- NPErg: Ergative case with case marker ‘ne: نے’
- NPAbl: Ablative with case marker ‘se: سے’
- NPIns: Instrumental case with case marker ‘se: سے’
- NPLoc1: Locative case with case marker ‘mi: ɳ میں’
- NPLoc2: Locative case with case marker ‘pr پر’
- NPDat: Dative case with case marker ‘kʊ کو’
- NPAcc: Accusative case with case marker ‘kʊ کو’

And ‘a’ (Agr in the code sample given in previous column) is the agreement feature of the the noun that is used for selecting the appropriate form of other categories that agree with nouns.

A noun is converted to an intermediate category common noun (CN; also known as N-Bar) which is then converted to NP category. CN deals with nouns and their modifiers. As an example consider adjectival modification:

```
fun AdjCN   : AP -> CN  -> CN ;

lin  AdjCN ap cn = {
     s = \\n,c =>
         ap.s ! n ! cn.g ! c ! Posit ++ cn.s ! n ! c ;
     g = cn.g
     } ;
```

The linearization of AdjCN gives us common nouns such as (پانی ٹھنڈا ʈ^{h}n ɖa pani: “cold water”) where a CN (پانی pani: “water”) is modified by an AP (ٹھنڈا , ʈ^{h}n ɖa “cold”).

Since Urdu adjectives also inflect in number, gender, case and degree, we need to concatenate the appropriate form of adjective that agrees with common noun. This is ensured by selecting the corresponding forms of adjective and common noun from their inflection tables using selection operator (‘!’). Since CN does not inflect in degree but the adjective does, we fix the degree to be positive (Posit) in this construction. Other modifiers include possibly adverbs, relative clauses, and appositional attributes.

A CN can be converted to a NP using different functions: common nouns with determiners; proper names; pronouns; and bare nouns as mass terms:

```
 fun DetCN   : Det -> CN -> NP  (e.g the boy)
fun UsePN   : PN -> NP (e.g John)
fun UsePron : Pron -> NP  (e.g he)
fun MassNP     : CN -> NP (e.g milk)
```

These different ways of building NP’s, which are common in different languages, are defined in the abstract syntax of the resource grammar, but the linearization of these functions is language dependent and is therefore defined in the concrete syntaxes.

4.2 Verb Phrases (VP)

A verb phrase is a single or a group of words that act as a predicate. In our construction Urdu verb phrase has following structure

```
lincat VP = {
     s   : VPHForm => {fin, inf: Str} ;
     obj  : {s : Str ; a : Agr} ;
     vType : VType ;
     comp : Agr => Str;
     embComp : Str ;
     ad  : Str  } ;
```

where

```
VPHForm =
   VPTense VPPTense Agr
   | VPReq HLevel | VPStem
```

and

```
  VPPTense = VPPres |VPPast |VPFutr;
 HLevel = Tu |Tum |Ap |Neutr
```

In GF representation a VP is a record with different fields. The most important field is ‘s’ which is an inflectional table and stores different forms of Verb.

At VP level we define Urdu tenses by using a simplified tense system, which has only three tenses, named VPPres, VPPast, VPFutr. In case of VPTense for every possible combination of VPPTense and agreement (gender, number, person) a tuple of two string values {fin, inf : Str} is created. ‘fin’ stores the coupla (auxiliary verb) , and ‘inf’ stores corresponding form of verb. VPStem is a special tense which stores the root form of verb. This form is used to create the full set of Urdu tenses at clause level (tenses in which the root form of verb is used, i.e. perfective and progressive tenses). Handling tenses at clause level rather than at verb phrase level simplifies the VP and results in a more efficient grammar.

The resource grammar has a common API which has a much simplified tense system, which is close to Germanic languages. It is divided into tense and anteriority. There are only four tenses named as present, past, future and conditional, and two possibilities of anteriority (Simul , Anter). This means it creates 8 combinations. This abstract tense system does not cover all the tenses in Urdu. We have covered the rest of tenses at clause level, even though these tenses are not accessible by the common API, but still can be used in language specific modules.

Other forms for verb phrases include request form (VPReq), imperative form (VPImp). There are four levels of requests in Urdu. Three of them correspond to (tʊ تو, tm تم , a:p آپ) honor levels and the fourth is neutral with respect to honorific levels. .

The Urdu VP is a complex structure that has different parts: the main part is a verb and then there are other auxiliaries attached to verb. For example an adverb can be attached to a verb as a modifier. We have a special field ‘ad’ in our VP representation. It is a simple string that can be attached with the verb to build a modified verb. In Urdu the complement of a verb precedes the actual verb e.g (وہ دوڑنا چاہتی ہے ʊo dʊɽna tʃahti: he: “she want to run”), here (چاہنا tʃahna “want”) is complement of verb (دوڑنا dʊɽna “run”), except in the case where, a sentence or a question is the complement of the verb. In that case complement of the verb comes at the very end of clause e.g (ʊo khta he: kh ʊo dʊɽti: he: وہ کہتا ہے کہ وہ دوڑتی ہے “he says that she runs”). We have two different fields named ‘compl’ and ‘embCompl’ in the VP to deal with these different situations.

‘vType’ field is used to store information about type of a verb. In Urdu a verb can be transitive, intransitive or double-transitive (Schmidt R. L., 1999). This information is important when dealing with ergativity in verb agreement. The information about the object of the verb is stored in ‘obj’ field. All this information that a VP carries is used when a VP is used in the construction of a clause.

A distinguishing feature of Urdu verb agreement is ‘ergativity’. Urdu is one of those languages that shows split ergativity at verb level. Final verb agreement is with direct subjective except in the transitive perfective tense. In transitive perfective tense verb agreement is with direct object. In this case the subject takes the ergative construction (subject with addition of ergative case marker (ne: نے).

However, in the case of the simple past tense, verb shows ergative behavior, but in case of other perfective tenses (e.g immediate past, remote past etc) there are two different approaches, in first one auxiliary verb (tʃka چکا) is used to make clauses. If (tʃka چکا) is used, verb does not show ergative behavior and final verb agreement is with direct subjective. Consider the following example

لڑکا کتاب خرید چکا ہے
lɽka $_{\text{Direct}}$ ktab $_{\text{Direct}}$ xri:d $_{\text{Root}}$ tʃka $_{\text{aux\_verb}}$ he:
The boy has bought a book

The second way to make the same clause is

لڑکے نے کتاب خریدی ہے
lɽke: ne: $_{\text{Erg}}$ ktab $_{\text{Direct\_Fem}}$ xri:di: $_{\text{Direct\_Fem}}$ he:
The boy has bought a book

In the first case the subject (lɽka, لڑکا “boy”) is in direct case and auxiliary verb agrees to subject, but in second case verb is in agreement with object and ergative case of subject is used. However, in the current implementation we follow the first approach.

In the concrete syntax we ensure this ergative behavior through the following code segment in GF. However the code given here is just a segment of the code that is relevant.

```
case vt of {
  VPPast => case vp.vType of {
    (Vtrans| VTransPost) => <NPErg, vp.obj.a>
    _                    => <NPC Dir, np.a>
                 } ;
  _ => <NPC Dir, np.a>
        } ;
```

e.g in case of simple past tense if verb is transitive then ergative case of noun is used and agreement is with object of verb. In all other cases direct case of noun is used and agreement is with subject of verb.

A VP is constructed in different ways; the simplest is

```
fun UseV    : V   -> VP ;
```

where V is the morphological category and VP is the syntactic category. There are other ways to make a VP from other categories, or combinations of categories. For example

```
fun AdvVP    : VP -> Adv -> VP ;
```

An adverb can be attached to a VP to make an adverbial modified VP. For example (i:haɳ یہاں سونا)

4.3 Adjective Phrases (AP)

Adjectives (A) are converted into the much richer category adjectival phrases (AP) at syntax level. The simplest function to convert is

```
fun PositA  : A  -> AP ;
```

Its linearization is very simple, since in our case AP is similar to A e.g.

```
fun PositA a = a ;
```

There are other ways of making AP for example

```
fun ComparA : A  -> NP -> AP ;
```

When a comparative AP is created from an adjective and a NP, constant "se: سے" is used between oblique form of noun and adjective. For example linearization of above function is

```
lin ComparA a np = {
   s = \\n,g,c,d => np.s ! NPC Obl ++ "se:"
      ++ a.s ! n ! g ! c ! d ;
     } ;
```

4.4 Clauses

A clause is a syntactic category that has variable tense, polarity and order. Predication of a NP and VP gives simplest clause

```
fun PredVP    : NP -> VP -> Cl ;
```

The subject-verb agreement is insured through agreement feature of NP which is passed to verb as inherent feature. A clause is of following type

```
lincat Clause : Type = {s : VPHTense =>
Polarity => Order => Str} ;
```

Here VPHTense represents different tenses in Urdu. Even though current abstract level of common API does not cover all tenses of Urdu, we cover them at clause level and can be accessed through language specific module. So, VPHTense is of following type

```
 VPHTense =  VPGenPres | VPPastSimple
             | VPFut | VPContPres
             | VPContPast | VPContFut
             | VPPerfPres  | VPPerfPast
             | VPPerfFut    | VPPerfPresCont
             | VPPerfPastCont
             | VPPerfFutCont | VPSubj
```

Polarity is used to make positive and negative sentences; Order is used to make simple and interrogative sentences. These parameters are of following forms

```
Polarity  = Pos | Neg
Order     = ODir | OQuest
```

PredVP function will create clauses with variable tense, polarity and order which are fixed at sentence level by different functions, one is.

fun UseCl : Temp -> Pol -> Cl -> S

Here Temp is syntactic category which is in the form of a record having field for Tense and Anteriority. Tense in the Temp category refers to abstract level Tense and we just map it to Urdu tenses by selecting the appropriate clause. This will create simple declarative sentence, other forms of sentences (e.g Question sentences) are handled in Questions categories of GF which follows next.

4.5 Question Clauses and Question Sentences

Common API provides different ways to create question clauses. The simplest way is to create from simple clause

fun QuestCl : Cl -> QCl ;

In Urdu simple interrogative sentences are created by just adding "ki:a کیا" at the start of a direct clause that already have been created at clause level. Hence, the linearization of above function simply selects appropriate form of clause and adds "ki:a کیا" at the start. However this clause still has variable tense and polarity which is fixed at sentence level e.g

fun UseQCl : Temp -> Pol -> QCl -> QS

Other forms of question clauses include clauses made with interrogative pronouns (IP), interrogative adverbs (IAdv), and interrogative determiners (IDet), categories. Some of the functions for creating question clauses are

fun QuestVP : IP -> VP -> QCl (e.g who walks)
fun QuestIAdv : IAdv -> Cl -> QCl (e.g why does he walk)

IP, IAdv, IDet etc are built at morphological level and can also be created with following functions.

fun AdvIP : IP -> Adv -> IP
fun IdetQuant : IQuant -> Num -> IDet ;
fun PrepIP : Prep -> IP -> IAdv ;

5. Example

As an example consider the translation of following sentence from English to Urdu, to see how our proposed system works at different levels.

He drinks hot milk.

Figure 1 shows the parse tree for this sentence. As a resource grammar developer our goal is to provide correct concrete level linearization of this tree for Urdu.

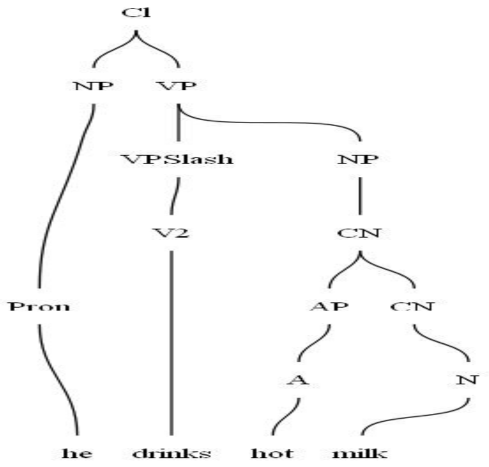


Figure 1. Parse tree of an example sentence

The nodes in this tree represent different categories and its branching shows how a particular category is built from other categories and/or leaves (words from lexicon). In GF notation these are the syntactic rules which are declared at abstract level. For example category CN can be built from an AP (adjectival phrase) and a CN. So in GF representation it has following type signature.

fun AdjCN : AP -> CN -> CN ;

A correct implementation of this rule in Urdu concrete syntax ensures correct formation of a common noun (گرم دودھ grm dʊd^{ɦ} "hot milk") from a CN (دودھ dʊd^{ɦ} "milk") modified by an Adjective (گرم , grm "hot").

A NP is constructed from this CN by one of the NP construction rules (see section 4.1 for details). A VPSlash (object missing VP) is build from a two place verb (پیتا pi:ta “drinks”). This VPSlash is then converted to VP through function

fun ComplSlash : VPSlash -> NP -> VP ;

Resulting VP and NP are grouped together to make a VP (گرم دودھ پیتا ہے ʈgrm dʊd^{ɦ} pi:ta he: “drinks hot milk”). Finally clause (گرم دودھ پیتا ہے وہ ʊh grm dʊd^{ɦ} pi:ta he: “he drinks hot milk”) is build from NP (وہ ʊh “he”) which is build from pronoun (وہ ʊh “he”) and VP (گرم دودھ پیتا ہے grm dʊd^{ɦ} pi:ta he: “drinks hot milk”). Language dependent concrete syntax assures that correct forms of words are selected from lexicon and word order is according to rules of that specific language. While, morphology makes sure that correct forms of words are built during lexicon development.

6. An application: Attempto

An experiment of implementing Controlled languages in GF is reported in (Angelov and Ranta, 2010). In this experiment, a grammar for Attempto Controlled English (Attempto, 2008) is implemented and then ported to six languages (English, Finnish, French, German, Italian, and Swedish) using the GF resource library. To demonstrate the usefulness of our grammar and to check its correctness, we have added Urdu to this set. Now, we can translate Attempto documents between all of these seven languages. The implementation followed the general recipe for how new languages can be added (Angelov and Ranta, 2009) and created no surprises. However the details of this implementation are beyond the scope of this paper.

7. Related Work

A suite of Urdu resources were reported in (Humayoun et el., 2006) including a fairly complete open-source Urdu morphology and a small fragment of syntax in GF. In this sense, it is a predecessor of Urdu resource grammar, implemented in a different but related formalism.

Like the GF resource library, Pargram project (Butt et el., 2007) aims at building a set of parallel grammars including Urdu. The grammars in Pargram are connected with each other by transfer functions, rather than a common representation. Further, the Urdu grammar is still one of the least implemented grammars in Pargram at the moment. This project is based on the theoretical framework of lexical functional grammar (LFG).

Other than Pargram, most work is based on LFG and translation is unidirectional i.e. from English to Urdu only. For instance, English to Urdu MT System is developed under the Urdu Localization Project (Hussain, 2004), (Sarfraz and Naseem, 2007) and (Khalid et el., 2009).

Similarly, (Zafer and Masood, 2009) reports another English-Urdu MT system developed with example based approach. On the other hand, (Sinha and Mahesh, 2009) presents a strategy for deriving Urdu sentences from English-Hindi MT system. However, it seems to be a partial solution to the problem.

8. Future Work

The common resource grammar API does not cover all the aspects of Urdu language, and non-generalizable language-specific features are supposed to be handled in language-specific modules. In our current implementation of Urdu resource grammar we have not covered those features. For example in Urdu it is possible to build a VP from only VPSlash (VPSlash category represents object missing VP) e.g (کھاتا ہے k^{h}ata he:) without adding the object. This rule is not present in the common API. One direction for future work is to cover such language specific features.

Another direction for future work could be to include the causative forms of verb which are not included in the current implementation due to efficiency issues.

9. Conclusion

The resource grammar we develop consists of 44 categories and 190 functions[3] which cover a fair enough part of language and is enough for

[3] http://www.grammaticalframework.org/lib/doc/synopsis.html

building domain specific application grammars including multilingual dialogue systems, controlled language translation, software localization etc. Since a common API for multiple languages is provided, this grammar is useful in applications where we need to parse and translate the text from one to many other languages.

However our approach of common abstract syntax has its limitations and does not cover all aspects of Urdu language. This is why it is not possible to use our grammar for arbitrary text parsing and generation.

10. References

Angelov K. and Ranta A. 2010. *Implementing controlled Languages in GF.* Controlled Natural Language (CNL) 2009, LNCS/LNAI Vol. 5972 (To appear)

Attempto 2008. Project Homepage. attempto.ifi.uzh.ch/site/

Butt M., 1995. *The Structures of Complex Predicate in Hindi* Stanford: CSLI Publications

Butt M., Dyvik H., King T. H., Masuichi H., and Rohrer C. 2002. *The Parallel Grammar Project.* In Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation. pp. 1-7.

Butt, M. and King, T. H. 2007. *Urdu in a Parallel Grammar Development Environment'.* In T. Takenobu and C.-R. Huang (eds.) *Language Resources and Evaluation*: Special Issue on Asian Language Processing: State of the Art Resources and Processing 41:191-207.

Forsberg M., and Ranta A., 2004. *Functional Morphology.* Proceedings of the Ninth ACM SIGPLAN International Conference of Functional Programming, Snowbird, Utah.

Humayoun M., Hammarström H., and Ranta A. *Urdu Morphology, Orthography and Lexicon Extraction. CAASL-2:* The Second Workshop on Computational Approaches to Arabic Script-based Languages, July 21-22, 2007, LSA 2007 Linguistic Institute, Stanford University. 2007

Hussain, S. 2004. *Urdu Localization Project. COLING*:WORKSHOP ON Computational Approaches to Arabic Script-based Languages, Geneva. pp. 80-81

Khalid, U., Karamat, N., Iqbal, S. and Hussain, S. 2009. *Semi-Automatic Lexical Functional Grammar Development.* Proceedings of the Conference on Language & Technology 2009.

Masica C., 1991. *The Indo-Aryan Languages, Cambridge,* Cambridge University Press, ISBN 9780521299442.

Ranta A., *Grammatical Framework: A Type-Theoretical Grammar Formalism.* The Journal of Functional Programming 14(2) (2004) 145–189.

Ranta A. *The GF Resource Grammar Library A systematic presentation of the library from the linguistic point of view.* to appear in the on-line journal Linguistics in Language Technology, 2009a.

Ranta A. *Grammars as Software Libraries. From Semantics to Computer Science,* Cambridge University Press, Cambridge, pp. 281-308, 2009b.

Rizvi, S. M. J. 2007. *Development of Algorithms and Computational Grammar of Urdu.* Department of Computer & Information Sciences/ Pakistan Institute of Engineering and Applied Sciences Nilore Islamabad. Pakistan.

Sarfraz H. and Naseem T., 2007. *Sentence Segmentation and Segment Re-Ordering for English to Urdu Machine Translation.* In Proceedings of the Conference on Language and Technology, August 07-11, 2007, University of Peshawar, Pakistan.

Schmidt R. L., 1999. *Urdu an Essential Grammar,*Routledge Grammars.

Sinha R., and Mahesh K., 2009. *Developing English-Urdu Machine Translation Via Hind.,* Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3) in conjunction with The twelfth Machine Translation Summit. Ottawa, Ontario, Canada.

Zafar M. and Masood A., 2009. *Interactive English to Urdu Machine Translation using Example-Based Approach.* International Journal on Computer Science and Engineering Vol.1(3), 2009, pp 275-282.

A Current Status of Thai Categorial Grammars and Their Applications

Taneth Ruangrajitpakorn and Thepchai Supnithi
Human Language Technology Laboratory
National Electronics and Computer Technology Center

{taneth.ruangrajitpakorn,thepchai.supnithi}@nectec.or.th

Abstract

This paper presents a current status of Thai resources and tools for CG development. We also proposed a Thai categorial dependency grammar (CDG), an extended version of CG which includes dependency analysis into CG notation. Beside, an idea of how to group a word that has the same functions are presented to gain a certain type of category per word. We also discuss about a difficulty of building treebank and mention a toolkit for assisting on a Thai CGs tree building and a tree format representations. In this paper, we also give a summary of applications related to Thai CGs.

1 Introduction

Recently, CG formalism was applied to several Thai NLP applications such as syntactic information for Thai to English RBMT (Ruangrajitpakorn et al., 2007), a CG treebank (Ruangrajitpakorn et al., 2009), and an automatic CG tagger (Supnithi et al., 2010). CG shows promises to handle Thai syntax expeditiously since it can widely control utilisations of function words which are the main grammatical expression of Thai.

In the previous research, CG was employed as a feature for an English to Thai SMT and it resulted better accuracy in term of BLEU score for 1.05% (Porkaew and Supnithi, 2009). CG was also used in a research of translation of noun phrase from English to Thai using phrase-based SMT with CG reordering rules, and it returned 75% of better and smoother translation from human evaluation (Porkaew et al., 2009).

Though CG has a high potential in immediate constituency analysis for Thai, it sill lacks of a dependency analysis which is also important in syntactical parsing. In this paper, we propose a category dependency grammar which is an upgraded version of CG to express a dependency relation alongside an immediate constituency bracketing. Moreover, some Thai dependency banks such as NAIST dependency bank (Satayamas and Kawtrakul, 2004) have been developed. It is better to be able to interchange data between a Thai CG treebank and a Thai dependency bank in order to increase an amount of data since building treebank from scratch has high cost.

In the point of resources and applications, Thai CG and CDG still have a few number of supported tools. Our CG treebank still contains insufficient data and they are syntactically simple and do not reflect a natural Thai usage. To add complex Thai tree, we found that Thai practical usage such as news domain contains a number of word and very complex.

An example of natural Thai text from news, which contains 25 words including nine underlined function words, is instanced with translation in Figure 1.

สำหรับ|การ|วาง|กำลัง|ของ|คน|เสื้อ|แดง| |ได้|มี|การ|วาง|บังเกอร์|รอบ|พื้นที่|ชุมนุม| |เอา|น้ำมัน|ราด| |รวมทั้ง|ยาง|รถยนต์|ที่|เสื่อม|สภาพ|แล้ว

lit: The red-shirts have put bunkers around the assembly area and poured oil and worn-out tires.

Figure 1. An example of Thai usage in natural language

We parsed the example in Figure 1 with CG and our parser returned 1,469 trees. The result is in a large number because many Thai structural issues in a syntactic level cause ambiguity.

The first issue is many Thai words can have multiple functions including employing grammatical usage and representing a meaning. For instance, a word "ที่" /tee/ can be a noun, a relative clause marker, a classifier, a preposition, and an adjective marker. A word "คน" /kon/ can refer to a person, a classifier of human being and it can denote an action. A word "กำลัง" /kumlung/ can serve as an auxiliary verb to express progressive aspect and also refers a meaning as a noun. A function word is a main grammatical representation and it hints an analyser to clarify an overall context structure. Regretfully, it is difficult for system to instantly indicate the Thai function words by focusing on the lexical surface and their surrounding lexicons. This circumstance is stimulates an over generation of many improper trees.

The second issue is a problem of Thai verb utilisations. Thai ordinarily allows to omit either a subject or an object of a verb. Moreover, a Thai intransitive verb is occasionally attached its indirect object without a preposition. Furthermore, Thai adjective allows to perform as a predicate without a marker. With an allowance of verb serialisation, these complexify linguists to design a category into well-crafted category set for verb. Therefore, many Thai verbs contain several syntactic categories to serve their many functions.

The last issues is a lack of an explicit boundary for a word, a phrase and a sentence in Thai. A Thai word and phrase boundary is implicit and a space is not significantly signified a boundary in the context. In addition, most of modifiers are attached after a core element. This leads to ambiguity of finding an ending of a subject with an attached adjective and relative clause since the verbs in attachment can be serialised and consequently placed with following main verb phrase (which is likely to be serialised either) without a signified indicator.

With these issues, a parser with only syntactic information merely returns a large number of all possible trees. It becomes difficulty and time consuming for linguists to select the correct one among them. Moreover, with many lexical elements, using a statistical parser has a very low chance to generate a correct tree and a manual tree construction is also required as a gold standard. Thus, we recently implemented an assistant toolkit for tree construction and tree representation to reduce linguists' work load and time consumption.

This paper aims to explain the current status of resource and tool for CG and CDG development for Thai language. We also listed open tools and applications that relate to CGs in this paper.

The rest of the paper is organised as follows. Section 2 presents a Thai categorial grammar and its related formalism. Section 3 explains status of CGs resources including syntactic dictionary and treebank. Section 4 shows details of a toolkit which assists linguist to manage and construct CGs derivation tree and tree representations. Section 5 provides information of applications that involve Thai CGs. Lastly, Section 6 concludes this paper and lists future works.

2 Thai Categorial Grammars

2.1 Categorial Grammar

Categorial grammar (Aka. CG or classical categorial grammar) (Ajdukiewicz, 1935; Carpenter, 1992; Buszkowski, 1998) is a formalism in natural language syntax motivated by the principle of constitutionality and organised according to the syntactic elements. The syntactic elements are categorised in terms of their ability to combine with one another to form larger constituents as functions or according to a function-argument relationship.

CG captures the same information by associating a functional type or category with all grammatical entities. Each word is assigned with at least one *syntactic category*, denoted by an argument symbol (such as *np* and *num*) or a functional symbol *X/Y* and *X\Y* that require *Y* from the right and the left respectively to form *X*.

The basic concept is to find the core of the combination and replace the grammatical modifier and complement with set of categories based on the same concept of the rule of fraction cancellation as follow:

$$\mathrm{np} \times \frac{\mathrm{s}}{\mathrm{np}} = \mathrm{s}$$

Upon applying to Thai, we have modified argument set and designed eight arguments shown in Table 1.

From the last version, two arguments were additionally designed. "*ut*" argument was added to denote utterance that is followed after a word "ว่า". The word "ว่า" has a special function to let the word after it perform as an exemplified utterance and ignore its appropriate category as it is signified an example in context. Comparing to "ws" argument, the word "ว่า" is functioned in a different sense which is used to denote a beginner of subordinate clause.

For "X" category, it is used for punctuation or symbol which takes the same categories from the left or right sides and produces the taken category. For instance, "ๆ" is a marker to denote after many types of content word. In details, this symbol signifies plurality while it is after noun but it intensifies a degree of meaning while it is placed after adjective.

Upon CG design, we allowed only binary bracketing of two immediate constituents. To handle serial construction in Thai including serial verb construction, we permitted the exactly same categories which are consequent to be combined. For example, Thai noun phrase 'มติ(np)|คณะรัฐมนตรี(np)' (lit: a consensus of the government) contains two consequent nouns without a joint word to form a noun phrase. Unfortunately, there still remain limits of syntactic parsing in CG that can not handle long dependency and word omission in this state.

argu-ment category	**definition**	**example**
np	a noun phrase	ช้าง (elephant), ผม (I, me)
num	a digit and a spelled-out number	หนึ่ง (one), 2 (two)
spnum	a number which is succeeding to classifier	นึง (one), เดียว (one)
pp	a prepositional phrase	ในรถ (in car), บนโต๊ะ (on table)
s	a sentence	ช้างกินกล้วย (an elephant eats a banana)
ws	a specific category for Thai which is assigned to a sentence or a phrase that begins with Thai word ว่า (that : sub-ordinate clause marker).	* ว่าเขาจะมาสาย 'that he will come late' * ว่าจะมาสาย 'that (he) will come late'
ut	an utterance using to exemplify a specific word after a word ว่า	คำ ว่า ดี 'the word "good"'
X	an undefined category that takes the same categories from the left or right sides and produces the taken category.	เด็ก ๆ (plural marker) สะอาด ๆ (intensifier)

Table 1. A list of Thai CDG arguments

2.2 Categorial Dependency Grammar

Categorial dependency grammar (CDG) is an extension of CG. CDG differs from CG in that a dependency direction motivated by Collins (1999) is additionally annotated to each slash notation in syntactic category. The derivation rules of CDG are listed as follow:

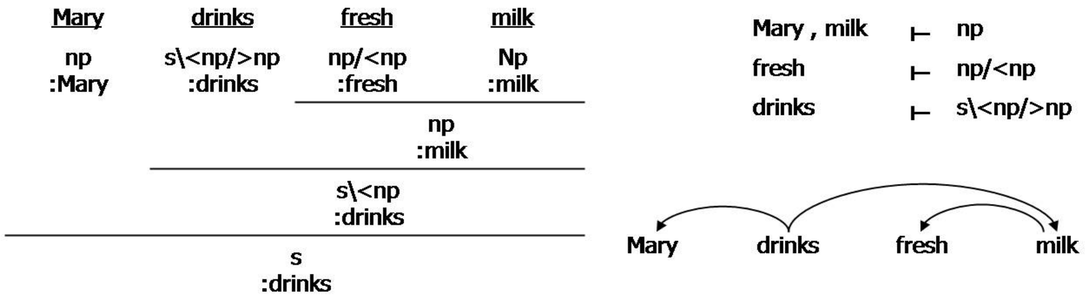
$$X/{<}Y : d_1 \; Y : d_2 \Rightarrow X : h(d_1) \rightarrow h(d_2)$$
$$X/{>}Y : d_1 \; Y : d_2 \Rightarrow X : h(d_1) \leftarrow h(d_2)$$
$$Y : d_1 \; X\backslash{<}Y : d_2 \Rightarrow X : h(d_1) \rightarrow h(d_2)$$
$$Y : d_1 \; X\backslash{>}Y : d_2 \Rightarrow X : h(d_1) \leftarrow h(d_2)$$

where the notations $h(d_1) \rightarrow h(d_2)$ and $h(d_1) \leftarrow h(d_2)$ mean a dependency linking from the head of the dependency structure d_1 to the head of d_2, and that linking from the head of d_2 to the head of d_1, respectively. Throughout this paper, a constituent type of the syntactic category *c* and the dependency structure *d* is represented by *c:d*.

Let us exemplify a dependency driven derivation of CDG of sentence 'Mary drinks fresh milk' in Figure 2. In Figure 2(a), each pair of constituents is combined to form a larger constituent with its head word. Figure 2(b) shows a dependency structure equivalent to the derivation in Figure 2(a).

Comparing to PF-CCG (Koller and Kuhlmann, 2009), there is different in that their PF-CCG dependency markers are fixed to the direction of slashes while CDG dependency markers are customised based on behaviour of a constituent.

CDG offers an efficient way to represent dependency structures alongside syntactic derivations. Apart from immediate constituency analysis, we can also investigate the correspondence between the syntactic derivations and the dependency structures. It benefits linguists in details a grammar for a specific language be-



(a) dependency driven derivation

(b) Equivalent dependency structure

Figure 2. Syntactic derivation of 'Mary drinks fresh milk' based on CDG

cause we can restrain the grammar in lexical level.

In this paper, our Thai CG was applied to CDG. For the case of serial construction, we set the left word as a head of dependency since Thai modifiers and dependents are ordinarily attached on right side.

2.3 Categorial Set

A categorial set is a group of lexicons that exactly contains the same function(s) in terms of their category amount and all their same syntactic categories. With a specific surface, each word certainly is in one categorial set. For example, suppose that we have following words and categories:

word	category	POS
ภูมิทัศน์,ขโมย,ล้อ,เกาะ	├ np	noun
ล้อ,เกาะ,ขโมย,กิน,ซื้อ	├ s\np/np	verb
ล้อ,เกาะ,ขวบ	├ np\np/num	classifier

We can group the given words into five groups based on the concept of categorial set shown in Table 2.

Set-index	Category member	Word member
1	np	ภูมิทัศน์
2	s\np/np	กิน,ซื้อ
3	np s\np/np	ขโมย
4	np s\np/np np\np/num	ล้อ,เกาะ
5	np\np/num	ขวบ

Table 2. An example of categorial set

For current status, we attain 183 categorial sets in total and the maximum amount of category member in a categorial set is 22 categories.

3 Categorial Grammars Resources

To apply categorial grammars to Thai NLP, syntactic dictionary and treebank are a mandatory.

3.1 Categorial Grammars Dictionary

For using in other work and researches, we collected all CGs information into one syntactic dictionary. An example of CGs dictionary is shown in Table 3. In a summary, our Thai CGs dictionary currently contain 70,193 lexical entries with 82 categories for both CG and CDG and 183 categorial sets.

Lexicon	CG	CDG	Cset no.
สมุด	np	np	0
เกาะ	np,s\np/np,np\np/num	np,s\<np/>np,np\>np/<num	15
กิน	s\np/np,s\np	s\<np/>np,s\<np	13
ถ้า	s\s/s,s/s/s	s\<s/>s,s/>s/>s	43
พูด	s\np/pp,s\np,s\np/ws	s\<np/>pp,s\<np,s\<np/>ws	19
เขียว	np\np,s\np	np\>np,s\<np	3
วิ่ง	s\np	s\<np	1
กล้าหาญ	np\np,s\np	np\>np,s\<np	3
นอน	s\np	s\<np	1
ขาย	s\np/np,s\np	s\<np/>np,s\<np	13
เสื้อ	np	np	0
ว่า	s\np/np,s\np/ws,np\np/ut	s\<np/>np,s\<np/>ws,np\>np/>ut	136
เพราะ	s\s/s,s/s/s	s\<s/>s,s/>s/>s	43

Table 3. An example of Thai CGs dictionary

3.2 Thai CDGTreebank

Our CG treebank was recently transformed into dependency-driven derivation tree with CDG. An example of derivation tree of sentence |การ|

ล่า|เสือ|เป็น|การ|ผจญภัย| 'lit: Tiger hunting is an adventure' comparing between CG and CDG is illustrated in Figure 3.

```
s                         s
 (np                       (np
  (np/(s\np)[การ]            (np/>(s\<np)[การ]
  s\np(                     s\<np(
   (s\np)/np[ล่า]              (s\<np)/>np[ล่า]
   np[เสือ]                    np[เสือ]
  )                         )
 )                         )
 s\np(                     s\<np(
  (s\np)/np[เป็น]             (s\<np)/>np[เป็น]
  np(                       np(
   np/(s\np)[การ]             np/>(s\<np)[การ]
   s\np[ผจญภัย]               s\<np[ผจญภัย]
  )                         )
 )                         )
)                         )
 (a) CG derivation tree    (b) CDG derivation tree
```

Figure 3. An example of a derivation tree in treebank comparing between CG and CDG

A status of transformed CDG treebank is 30,340 text lines which include 14,744 sentences, 9,651 verb phrases or subject-omitted sentences and 5,945 noun phrases. However, the average word amount of this treebank is 3.41 words per tree which is obviously short.

Upon an attempt to increase a number of trees, we considered that the existing trees are simple and not signify all utilisations of natural Thai text. Therefore, news domain of BEST (Kosawat et al., 2009) corpus was chosen to fulfil such issues because of its practical usage. From our observation, we found that most of data are ranged from 25 to 68 words and the longest line contains 415 words which is extremely long for parser to handle it efficiently.

After a prior experiment, we found that our GLR parser with CDG information resulted 514.62 tree alternatives in average from the range of three to fifteen words per sentence from news domain in BEST. With problems from ambiguous syntax of Thai, to automatically select a correct tree is extremely difficult since several resulted trees are grammatically correct and semantically sound but they are not proper for their context. It becomes difficulty for linguists to select an appropriate one among them. In order to solve that problem, we implemented a toolkit to assist linguists on constructing treebank with such a long and complicated sentence. The manual annotated tree will be used as a gold standard and confidentially apply for statistical parser development.

4 CGs Tree Supported Tool

Building a resource is a laboured work especially a treebank construction. For Thai language which uses several function words to express grammatical function in context, an immediate constituency analysis and a dependency analysis become difficult since many word pair can cause ambiguity and complexity among them. Additionally, a representation of a derivation tree in textual format is excessively complex to be analysed or approved. To reduce a burden of linguists, we developed a toolkit to help a linguists with graphical user-interface in manual tree construction.

4.1 CGs Toolkit

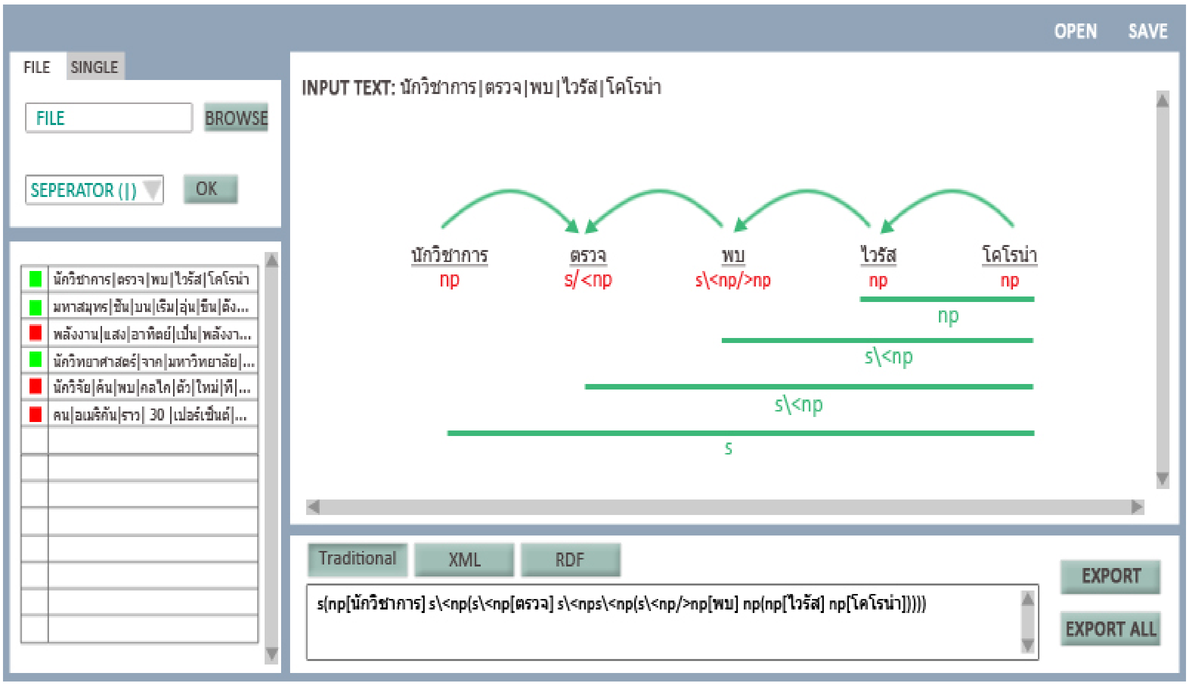
The proposed toolkit supports multi-tasks which are annotating CG tag to a word, bracketing intermediate constituents, generating dependency-driven derivation tree in multiple formats, and visualising graphical tree.

4.1.1 Category Annotator

Category annotator supports users to select an appropriate CDG category for each word. The system takes word-segmented input text. It starts with checking possible categories with the given CDG dictionary and lists all of them to each word. Users only select a correct category for each. Unless the word is known or the required category for the word is present, user has to add a new category for the word and the system contiguously updates the dictionary with the given data for further usage.

4.1.2 Dependency-driven Derivation Tree Generator

This system is implemented for manual annotating tree information and dependency relation to a text that is difficult for parser to generate tree such as a text with multiple serial verb constructions, a complex head-dependent relation word pairs, etc. A captured picture of user-interface



INPUT TEXT: นักวิชาการ|ตรวจ|พบ|ไวรัส|โคโรน่า

🟩	นักวิชาการ|ตรวจ|พบ|ไวรัส|โคโรน่า
🟩	มหาสมุทร|ชั้น|บน|เริ่ม|อุ่น|ขึ้น|ตั้ง...
🟥	พลังงาน|แสง|อาทิตย์|เป็น|พลังงา...
🟩	นักวิทยาศาสตร์|จาก|มหาวิทยาลัย|...
🟥	นักวิจัย|ค้น|พบ|กลไก|ตัว|ใหม่|ที่|...
🟥	คน|อเมริกัน|ราว| 30 |เปอร์เซ็นต์|...

s(np[นักวิชาการ] s\<np(s\<np[ตรวจ] s\<nps\<np(s\<np/>np[พบ] np(np[ไวรัส] np[โคโรน่า]))))

Figure 4. A snapshot of dependency-driven derivation tree generator

working on immediate constituency and dependency annotation is illustrated in Figure 4.

We provide a user-interface for linguists and experts to easily annotate brackets covering. Users begin a process by selecting a pair of words that are a terminal of leaf node. The system apparently shows only categories of the word that can be possibly combined within the bracket for selecting. After choosing categories of those two constituents, the system automatically generates a result category. Next, users will continue the process for other constituents until one top result category is left.

After users finish the bracketing process, dependency relation will be generated from annotated dependency marker within categories without manual assignment.

4.1.3 Tree Visualiser

The system includes a function to create a graphical tree from a file in textual formats. It provides a function to modify a tree by editing a word spelling and its syntactic category and shifting a branch of syntactic tree to another.

4.2 Tree Representation

The CGs toolkit allows users to export a tree output in two representations; traditional textual tree format and XML format.

Throughout all tree format examples, we exemplify a Thai sentence 'นักวิชาการ ตรวจ พบ ไวรัส โคโรน่า' (lit: an expert discovers corona virus.) with the following categories:

Word	CDG category
นักวิชาการ (expert) ไวรัส (virus) โคโรน่า (corona)	├ np
ตรวจ (diagnose)	├ s\<np
พบ (discover)	├ s\<np/>np

4.2.1 Traditional Textual Tree Format

A traditional textual tree format represents a terminal (*w*) with its category (*c*) in form of *c[w]*. The brackets are enclosed two constituents split by space with parentheses and the result category (c_r) is placed before the open parenthesis in format *c_r(c[w] c[w])*. Figure 5 shows an example of a traditional textual tree format.

s(np[นักวิชาการ] s\<np(s\<np[ตรวจ] s\<np(s\<np(s\<np/>np[พบ] np(np[ไวรัส] np[โคโรน่า])))

Figure 5. An example of a traditional textual tree format of 'นักวิชาการ ตรวจ พบ ไวรัส โคโรน่า'

4.2.2 XML Tree Format

For XML tree format, we design three tag sets, i.e., *word* tag, *tree* tag and *input* tag. The *word* tag bounds a terminal to mark a word. In a start-tag of *word* tag, there are two attributes which are *cat* to assign a category in a value and text to assign a given text in a value. For *tree* tag, it marks a combination of either *word* tags or *tree* tags to form another result category. It contains two previous attributes with an additional attribute, i.e., a *head* attribute to fill in a notation that which word has a head-outward relation value where '0' value indicates head from left constituent and '1' value indicates head from right constituent. The input tag shows a boundary of all input and it has attributes to show line number, raw input text and status of tree building process. Figure 6 illustrates an XML tree representation.

5 Thai CGs Related Applications

Several applications related to Thai CGs or used Thai CGs as their syntactic information have been implemented recently. Below is a summary of their methodology and result.

5.1 CG AutoTagger for Thai

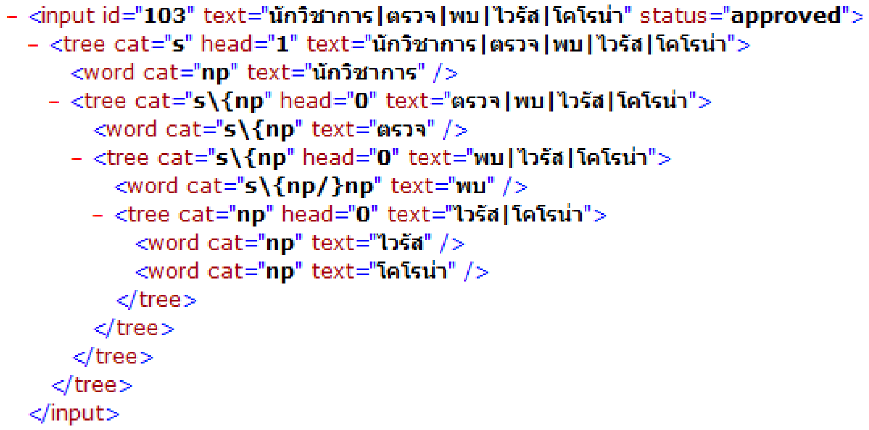
To reduce an amount of trees generated from a parser with all possible categories, an automatic syntactic category tagger (Supnithi et al., 2010) was developed to disambiguate unappropriated combinations of impossible categories. The system was developed based on CRF and Statistical Alignment Model based on information theory (SAM) algorithm. The accuracy 89.25% in word level was acquired. This system also has a function to predict a syntactic category for an unknown word and 79.67% of unknown word are predicted correctly.

5.2 Chunker

With a problem of a long sentence in Thai, chunker was implemented to group a consequent of words to larger unit in order to reduce a difficulty on parsing too many lexical elements. CRD method with syntactic information from CG and categorial set was applied in the system to chunk a text into noun phrase, verb phrase, prepositional phrase, and adverbial phrase. Moreover, the system also attempts to handle a compound word that has a form like sentence. The result was impressive as it improved 74.17% of accuracy on sentence level chunking and 58.65% on sentence-form like compound noun.

5.3 GLR parser for Thai CG and CDG

Our implemented LALR parser (Aho and Johnson, 1974) was improved to GLR parser for syntactically parse Thai text. This parser was developed to return all possible trees form input to show a baseline that covers all syntactic possibilities. For our GLR parser, a grammar rule is not manually determined, but it is automatically produced by any given syntactic notations aligned with lexicons in a dictionary. Hence, this GLR parser has a coverage including CG and CDG formalism parsing. Furthermore, our GLR parser accepts a sentence, a noun phrase, a verb phrase and prepositional phrase. However, the parser does not only return the best first tree, but also all parsable trees to gather all ambiguous trees since Thai language tends to be ambiguous because of lacking explicit sentence, phrase and word boundary. This parser includes a pre-process to handle named-entities, numerical expression and time expression.



```
- <input id="103" text="นักวิชาการ|ตรวจ|พบ|ไวรัส|โคโรน่า" status="approved">
  - <tree cat="s" head="1" text="นักวิชาการ|ตรวจ|พบ|ไวรัส|โคโรน่า">
      <word cat="np" text="นักวิชาการ" />
    - <tree cat="s\{np" head="0" text="ตรวจ|พบ|ไวรัส|โคโรน่า">
        <word cat="s\{np" text="ตรวจ" />
      - <tree cat="s\{np" head="0" text="พบ|ไวรัส|โคโรน่า">
          <word cat="s\{np/}np" text="พบ" />
        - <tree cat="np" head="0" text="ไวรัส|โคโรน่า">
            <word cat="np" text="ไวรัส" />
            <word cat="np" text="โคโรน่า" />
          </tree>
        </tree>
      </tree>
    </tree>
  </input>
```

Figure 6. An example of XML tree format of 'นักวิชาการ ตรวจ พบ ไวรัส โคโรน่า'

6 Conclusion and Future Work

In this paper, we update our Thai CG information and a status of its resources. We also propose CDG for Thai, an extended version of CG. CDG offers an efficient way to represent dependency structures with syntactic derivations. It benefits linguists in terms of they can restrain Thai grammar in lexical level. With CDG dependency-driven derivation tree, both bracketing information and dependency relation are annotated to every lexical units. In the current state, we transformed our CG dictionary and CG treebank into CDG formalism.

With an attempt to increase an amount of our treebank with a complex text, CDG tree toolkit was developed for linguists to manual managing a derivation tree. This toolkit includes a CDG category tagger tool, dependency-driven derivation tree generator, and tree visualiser. This toolkit can generate an output in two formats which are traditional textual tree and XML tree. The XML tree format is an option for standardised format or further usage such as applying tree for ontology.

We also summarised CGs related works and their accuracy. They included an automatic CG tagger and a Thai phrase chunker.

In the future, we plan to increase an amount of CGs derivation trees of complex sentence and practical language. Moreover, we will implement a system to transform an existing Thai dependency bank to CDG format to gain more number of trees. We also plan to include semantic meaning into derivation tree and represent such trees in an RDF format. In addition, statistical parser will be implemented based on the CDG derivation trees.

References

Ajdukiewicz Kazimierz. 1935. Die Syntaktische Konnexitat, Polish Logic.

Aho Alfred, and Johnson Stephen. 1974. LR Parsing, Proceedings of Computing Surveys, Vol. 6, No. 2.

Bar-Hillel Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. 29(1): 47-58.

Carpenter Bob. 1992. Categorial Grammars, Lexical Rules,and the English Predicative, In R. Levine, ed., Formal Grammar: Theory and Implementation. OUP.

Collins Micheal. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. Thesis, University of Pennsylvania.

Koller Alexander, and Kuhlmann Marco. 2009. Dependency trees and the strong generative capacity of ccg, Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: 460-468.

Kosawat Krit, Boriboon Monthika, Chootrakool Patcharika, Chotimongkol Ananlada, Klaithin Supon, Kongyoung Sarawoot, Kriengket Kanyanut, Phaholphinyo Sitthaa, Purodakananda Sumonmas, Thanakulwarapas Tipraporn, and Wutiwiwatchai Chai. 2009. BEST 2009: Thai Word Segmentation Software Contest. The 8th International Symposium on Natural Language Processing: 83-88.

Porkaew Peerachet, Ruangrajitpakorn Taneth, Trakultaweekoon Kanokorn, and Supnithi Thepchai.. 2009. Translation of Noun Phrase from English to Thai using Phrase-based SMT with CCG Reordering Rules, Proceedings of the 11th conference of the Pacific Association for Computational Linguistics (PACLING).

Porkaew Peerachet, and Supnithi Thepchai. 2009. Factored Translation Model in English-to-Thai Translation, Proceedings of the 8th International Symposium on Natural Language Processing.

Ruangrajitpakorn Taneth, Na Chai Wasan , Boonkwan Prachya, Boriboon Monthika, Supnithi Thepchai. 2007. The Design of Lexical Information for Thai to English MT, Proceedings of the 7th International Symposium on Natural Language Processing.

Ruangrajitpakorn Taneth, Trakultaweekoon Kanokorn, and Supnithi Thepchai. 2009. A Syntactic Resource for Thai: CG Treebank, Proceedings of the 7th Workshop on Asian Language Resources, (ACL-IJCNLP): 96–102.

Satayamas Vee, and Kawtrakul Asanee . 2004. Wide-Coverage Grammar Extraction from Thai Treebank. Proceedings of Papillon 2004 Workshops on Multilingual Lexical Databases, Grenoble, France.

Supnithi Thepchai, Ruangrajitpakorn Taneth, Trakultaweekoon Kanokorn, and Porkaew Peerachet. 2010. AutoTagTCG : A Framework for Automatic Thai CG Tagging, Proceedings of the 7th international conference on Language Resources and Evaluation (LREC).

Chained Machine Translation Using Morphemes as Pivot Language

Wen Li
Institute of Intelligent Machines, Chinese Academy of Sciences, University of Science and Technology of China
xtliwen@mail.ustc.edu.cn

Lei Chen
Institute of Intelligent Machines, Chinese Academy of Sciences
alan.cl@163.com

Wudabala
Institute of Intelligent Machines, Chinese Academy of Sciences
hwdbl@126.com

Miao Li
Institute of Intelligent Machines, Chinese Academy of Sciences
mli@iim.ac.cn

Abstract

As the smallest meaning-bearing elements of the languages which have rich morphology information, morphemes are often integrated into state-of-the-art statistical machine translation to improve translation quality. The paper proposes an approach which novelly uses morphemes as pivot language in a chained machine translation system. A machine translation based method is used therein to find the mapping relations between morphemes and words. Experiments show the effectiveness of our approach, achieving 18.6 percent increase in BLEU score over the baseline phrase-based machine translation system.

1 Introduction

Recently, most evaluations of machine translation systems (Callison-Burch et al., 2009) indicate that the performance of corpus-based statistical machine translation (SMT) has come up to the traditional rule-based method. In the corpus-based SMT, it is difficult to exactly select the correct inflections (word-endings) if the target language is highly inflected. This problem will be more severe if the source language is an isolated language with non-morphology (eg. Chinese) and the target language is an agglutinative language with productive derivational and inflectional morphology (eg. Mongolian: a minority language of China). In addition, the lack of large-scale parallel corpus may cause the sparse data problem, which will be more severe if one of the source language and the target language is highly inflected. As the smallest meaning-bearing elements of the languages which have rich morphology information, morphemes are the compact representation of words. Using morphemes as the semantic units in the parallel corpus can not only help choose the correct inflections, but also alleviate the data sparseness problem partially.

Many strategies of integrating morphology information into state-of-the-art SMT systems in different stages have been proposed. (Ramanathan et al., 2009) proposed a preprocessing approach for incorporating syntactic and Morphological information within a phrase-based English-Hindi SMT system. (Watanabe et al., 2006) proposed a method which uses Porter stems and even 4-letter prefixes for word alignment. (Koehn et al., 2007) proposed the factored translation models which combine feature functions to handle syntactic, morphological, and other linguistic information in a log-linear model during training. (Minkov et al., 2007) made use of the information of morphological structure and source language in postprocessing to improve SMT quality. (de Gispert et al., 2009) adopted the Minimum Bayes Risk decoding strategy to combine output from identical SMT system, which is trained on alternative morphological decompositions of the source language.

Meanwhile, the SMT-based methods are widely used in the area of natural language processing. (Quirk et al., 2004) applied SMT to generate novel paraphrases. (Riezler et al., 2007) adopted an SMT-based

Proceedings of the 8th Workshop on Asian Language Resources, pages 169–177,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

method to query expansion in answer retrieval. (Jiang and Zhou, 2008) used SMT to generate the second sentence of the Chinese couplets.

As opposed to the above strategies, the paper proposes an approach that uses morphemes as pivot language in a chained SMT system, for translating Chinese into Mongolian, which consists of two SMT systems. First, Chinese sentences are translated into Mongolian morphemes instead of Mongolian words in the Chinese-Morphemes SMT (SMT_1). Then Mongolian words are generated from morphemes in the Morphemes-Mongolian SMT (SMT_2). The essential part of the chained SMT system is how to find the mapping relations between the morphemes and words, which is considered as a procedure of machine translation in our approach. More concretely, the first challenge of this approach is to investigate some effective strategies to segment the Mongolian corpus in the Chinese-Mongolian parallel corpus. And the second challenge is how to efficiently generate Mongolian words from morphemes. Additionally, on the one hand Mongolian words may have multiple kinds of morphological segmentations. On the other hand there is also the ambiguity of word boundaries in the processing of generating Mongolian words from morphemes. In order to solve these ambiguities, a SMT-based method is applied in that word context and morphemes context can be taken into account in this method.

The remainder of the paper is organized as follows. Section 2 introduces two methods of morphological segmentation. Section 3 presents the details of chained SMT system. Section 4 describes the experiment results and evaluation. Section 5 gives concluding remarks.

2 Morphological segmentation

Mongolian is a highly agglutinative language with a rich set of affixes. Mongolian contains about 30,000 stems, 297 distinct affixes. A big growth in the number of possible word forms may occur due to the inflectional and derivational productions. An inflectional suffix is a terminal affix that does not change the parts of speech of the root during concatenation, which is added to maintain the syntactic environment of the root. For instance, the Mongolian word “YABVGSAN” (walking) in the present continuous tense syntactic environment consists of the root “YABV” (walk) and the suffix “GSAN” (ing). Whereas, when a verb root “UILED” (do) concatenates a noun derivational suffix “BURI”, it changes to a noun “UILEDBURI” (factory). According to that whether linguistic lemmatization (the reduction to base form) is considered or not, the paper proposes two methods of morphological segmentation. The two methods are tested on the same training databases.

The root lemmatization is concerned in the first method, which is called the SMT-based morphological segmentation (SMT-MS) in this paper. Given the Mongolian-morphemes parallel corpus, this method trains a Mongolian-morphemes SMT to segment Mongolian words. The root lemmatization is considered in the original morphological pre-segmented training corpus. So the SMT-based method can also deal with root lemmatization when it segments a Mongolian word. For instance, the Mongolian word “BAYIG_A” exhibits the change of spelling during the concatenation of the morphemes “BAI” and “G_A”. We also investigate whether it is effective if those roots are identical to the original word forms. In other words, the root lemmatization is ignored in the second method, which takes the gold standard morphological segmentation corpus as a trained model of Morfessor (Creutz and Lagus, 2007) and uses the Viterbi decoding algorithm to segment new words. Therefore, this method is called the Morfessor-based morphological segmentation (Mor-MS). For instance, the word “BAYIG_A” will be segmented to “BAYI” and “G_A” instead of “BAI” and “G_A”.

The mathematical description of SMT-MS is the same as the traditional machine translation system. In the Mor-MS method, the morphological segmentation of a word can be regarded as a flat tree (morphological segmentation tree), where the root node corresponds to the whole word and the leaves correspond to morphemes of this word. Figure 1 gives an example. First, the joint probabilistic distribution (Creutz and Lagus, 2007) of all morphemes in the morphological segmentation tree are calculated. And then by using the Viterbi decoding algorithm, the maximum probability segmentation combination is selected.

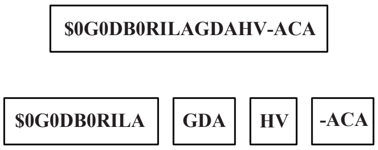


Figure 1: Morphological segmentation tree

3 Chained SMT system

3.1 Overview

In order to improve the performance of Chinese-Mongolian machine translation, the paper proposes an approach which incorporates the morphology information within a chained SMT system. More concretely, this system first translates Chinese into Mongolian morphemes instead of Mongolian words by the Chinese-Morphemes SMT. And then it uses the Morphemes-Mongolian SMT to translate Mongolian morphemes into Mongolian words. Namely, morphemes are regarded as pivot language in this system.

The chained SMT system consists of a morphological segmentation system and two phrase-base machine translation systems, which are given as follows:

- Morphological segmentation: segmenting Mongolian words (from the Chinese-Mongolian parallel corpus) into Mongolian morphemes and obtaining two parallel corpus: Chinese-Morphemes parallel corpus and Morphemes-Mongolian parallel corpus.
- SMT$_1$: training the Chinese-Morphemes SMT on the Chinese-Morphemes parallel corpus.
- SMT$_2$: training the Morphemes-Mongolian SMT on the Morphemes-Mongolian parallel corpus.

Figure 2 illustrates the overview of chained SMT system.

3.2 Phrase-based SMT

The authors assume the reader to be familiar with current approaches to machine translation, so that we briefly introduce the phrase-based statistical machine translation model (Koehn et al., 2003) here, which is the foundation of chained SMT system.

In statistical machine translation, given a source language f, the aim is to seek a target language e, such that $P(e|f)$ is maximized. The phrase-based translation model can be expressed by the following formula:

$$e^* = \arg\max_e P(e|f) = \arg\max_e \{P(f|e)P(e)\}$$

where e^* indicates the best result, $P(e)$ is the language model and $P(f|e)$ is the translation model. According to the standard log-linear model proposed by (Och and Ney, 2002), the best result e^* that maximizes $P(e|f)$ can be expressed as follows:

$$e^* = \arg\max_e \{\sum_{m=1}^{M} \lambda_m h_m(e,f)\}$$

where M is the number of feature functions, λ_m is the corresponding feature weight, each $h_m(e,f)$ is a feature function.

In our chained SMT system, SMT$_1$, SMT$_2$ and the SMT for morphological segmentation (namely SMT-MS in Section 2) are all phrase-based SMTs.

3.3 Features of Chained SMT system

As shown in Figure 2, Chinese is translated into Mongolian morphemes in SMT$_1$, which is the core part of the chained SMT system. Here morphemes are regarded as words. Therefore, morphemes can play important roles in SMT$_1$ as follows: the roots present the meaning of the word and the suffixes help select the correct grammatical environment. The word alignments between Chinese words and Mongolian morphemes are learned automatically by GIZA++. Figure 3 gives an instance of word alignment in SMT$_1$.

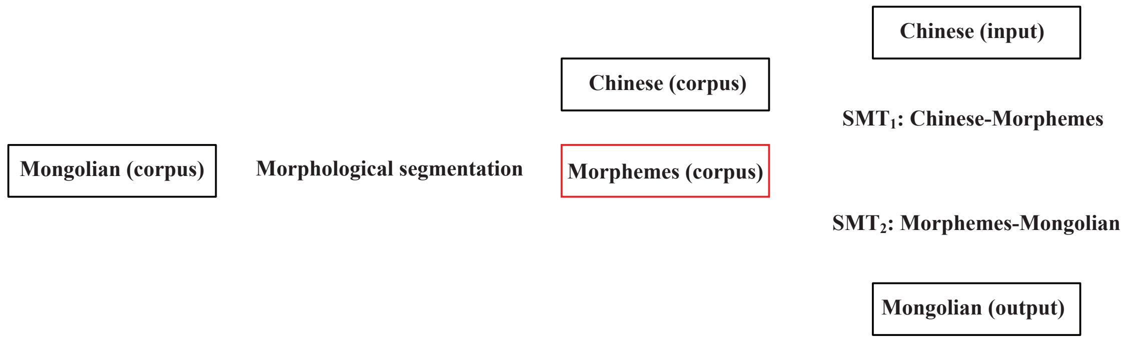


Figure 2: Morphemes as pivot language in Chained SMT system

We can see that the morphemes "BI","TAN" etc. are all regarded as words.

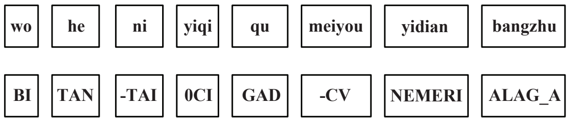


Figure 3: Word alignments between Chinese words and Mongolian morphemes in SMT_1

All the most commonly used features of standard phrase-based SMT, including phrase translation model, language model, distortion model and word penalty, are selected in SMT_1. These commonly used features determine the quality of translation together. The phrases of f and e are ensured to be good translations of each other in the phrase translation model $P(f|e)$. The fluent output is guaranteed in the language model $LM(e)$. The reordering of the input sentence is allowed in the distortion model $D(e, f)$. The translation is however more expensive with the more reordering. The translation results are guaranteed neither too long nor too short in the word penalty $W(e)$.

In SMT-MS and SMT_2, the task is to find the mapping relations between Mongolian morphemes and Mongolian words, which is considered as the word-for-word translation. Therefore, only phrase translation model and language model are considered. All the features weights are uniform distribution by default. Mongolian words may have multiple kinds of morphological segmentations. And there is the ambiguity of word boundaries in the processing of generating Mongolian words from morphemes. These ambiguities can be solved in SMT-MS and SMT_2 respectively, since the SMT-based method can endure mapping errors and solve mapping ambiguities by the multiple features which can consider the context of Mongolian words.

4 Experiments

4.1 Experimental setup

In the experiments, first we preprocess the corpus, such as converting Mongolian into Latin Mongolian and filtering the apparent noisy segmentation of the gold standard morphological segmentation corpus. And then we evaluate the effectiveness of the SMTs which find the mapping relations between the morphemes and their corresponding word forms. Namely, SMT-MS and SMT_2. As mentioned above, SMT_1 is the core part of the chained SMT system, which decides the final quality of translation results. So the evaluation of SMT_1 can be reflected by the evaluation of translation results of whole chained SMT system. Finally, we evaluate and analyze the performance of the chained SMT system by using the automatic evaluation tools.

The translation model consists of a standard phrase-table with lexicalized reordering. Bidirectional word alignments obtained with GIZA++ are intersected using the grow-diag-final heuristic (Koehn et al., 2003). Translations of phrases of up to 7 words long are collected and scored with translation probabilities and lexical weighting. The language model of morphemes is a 5-gram model with Kneser-Ney smoothing. The language model of Mongolian word is 3-gram model with Kneser-Ney smoothing too. All the language models are built with the SRI language modeling toolkit (Stolcke, 2002). The log-linear model feature weights are learned by using minimum error rate training (MERT) (Och, 2003) with BLEU score (Papineni et al., 2002) as the objective function.

4.2 Corpus preprocessing

The Chinese-Mongolian parallel corpus and monolingual sentences are obtain from the 5th China Workshop on Machine Translation. In the experiments, we first convert Mongolian corpus into Latin Mongolian. In morphological segmentation, the gold standard morphological segmentation corpus contains 38000 Mongolian sentences, which are produced semi-automatically by using the morphological analyzer Darhan (Nashunwukoutu, 1997) of Inner Mongolia University. Moreover, in order to obtain the higher quality corpus, most of the wrong segmentation in the results of morphological analyzer are modified manually by the linguistic experts. However, there are still some wrong segmentation in the gold standard corpus. Therefore, we adopt a strategy to filter the apparent noisy segmentation. In this strategy, the sum of the lengths of all the morphemes is required to be equivalent to the length of the original word. After filtering, there are still 37967 sentences remained. In addition, the word alignment is vulnerable to punctuation in SMT-MS. So all punctuation of the gold standard morphological segmentation corpus are removed to eliminate some mistakes of the word alignment.

Meanwhile, since the Chinese language does not have explicit word boundaries, we also need to do the segmentation of Chinese words. The word segmentation tool ICTCLAS (Zhang, 2008) is used in the experiments.

4.3 Evaluation of SMT-MS and SMT$_2$

The tasks of SMT-MS and SMT$_2$ are to find the mapping relations between the morphemes and their corresponding word forms. Morphological segmentation is done by SMT-MS. Contrarily, SMT$_2$ is used to generate the words from morphemes. To evaluate the effectiveness of SMT-MS and SMT$_2$, we divide the filtered gold standard corpus into two sets for training (90%) and testing (10%) respectively. The correct morpheme boundaries are counted for SMT-MS evaluation, while the correct word boundaries are counted for SMT$_2$ evaluation. We use the two measures *precision* and *recall* on discovered word boundaries to evaluate the effectiveness of SMT-MS and SMT$_2$, where *precision* is the proportion of correctly discovered boundaries among all discovered boundaries by the algorithm, and *recall* is the proportion of correctly discovered boundaries among all correct boundaries. A high *precision* indicates that a morpheme boundary is probably correct when it is suggested. However the proportion of missed boundaries can not be obtained from it. A high *recall* indicates that most of the desired boundaries were indeed discovered. However it can not point out how many incorrect boundaries were suggested either. In order to get a comprehensive idea, we also make use of the evaluation method: *F-measure* as a compromise.

$$\textit{F-measure} = \frac{1}{\frac{1}{2}(\frac{1}{\textit{precision}} + \frac{1}{\textit{recall}})}$$

These measures assume values between zero and 100%, where high values reflect good performance. Therefore, we evaluate the SMT-based methods by incrementally evaluating the features used in our phrase-based SMT model.

Table 1 gives the evaluation results, where PTM denotes Phrase Translation Model, LW denotes Lexical Weight, LM denotes Language Model, IPTM denotes Inverted PTM, ILW denotes Inverted LW. Table 1(a) and Table 1(b) are corresponding to the evaluations of SMT-MS and SMT$_2$ respectively, where P, R and F denote the three measures, namely *precision*, *recall* and *F-measure*.

The results show that when we add more features incrementally, the *precision*, *recall* and *F-measure* are improved consistently. These indicate that the features are helpful for finding the mapping relations between morphemes and Mongolian words.

Table 1: Evaluation of SMT-MS and SMT_2

(a) Evaluation of SMT-MS

Feature	$P(\%)$	$R(\%)$	$F(\%)$
(1): PTM+LW	73.35	72.45	72.90
(2): (1)+LM	94.91	94.91	94.91
(3): (2)+IPTM+ILW	94.95	94.95	94.95

(b) Evaluation of SMT_2

Feature	$P(\%)$	$R(\%)$	$F(\%)$
(1): PTM+LW	75.86	60.04	67.03
(2): (1)+LM	95.13	89.92	92.45
(3): (2)+IPTM+ILW	95.13	90.02	92.51

Table 2: Evaluation of systems

(a) without MERT

	NIST	BLEU (%)
Baseline	5.3586	20.71
Chain_1	5.6471	23.91
Chain_2	5.6565	24.57

(b) with MERT

	NIST	BLEU (%)
Baseline	5.6911	24.13
Chain_1	5.7439	24.70
Chain_2	5.8401	25.80

4.4 Evaluation of chained SMT system

We use NIST score (Doddington, 2002) and BLEU score (Papineni et al., 2002) to evaluate chained SMT system. The training set contains 67288 Chinese-Mongolian parallel sentences. The test set contains 400 sentences, where each sentence has four reference sentences which are translated by native experts.

In the training phase, we convert Mongolian into Latin Mongolian. And while in the test phase, we convert the Latin Mongolian back into the traditional Mongolian words. We compare the chained SMT system with the standard phrase-based SMT. Table 2 gives the evaluation of experiment result of each system, where Baseline is the standard phrase-based SMT, Chain_1 is a chained SMT consisting of SMT-MS, SMT_1 and SMT_2, Chain_2 is also a chained SMT consisting of Mor-MS, SMT_1 and SMT_2. In Table 2(b), we use MERT to train the feature weights of the baseline system and the feature weights of SMT_1 in Chain_1 and Chain_2.

The experiment results show that both Chain_1 and Chain_2 are much better than the baseline system. The BLEU score is improved by 18.6 percent, from 20.71 (Baseline) to 24.57 (Chain_2). In addition, Chain_2 is better than Chain_1. We believe that it is essentially related to the different morphemes corpus of Chain_1 and Chain_2. The morphemes corpus of Chain_1 takes lemmatization into account, while the morphemes corpus of Chain_2 changes all morphemes to inflected forms which are identical to the original word forms. As the example in Section 2, the word "BAYIG_A" is segmented into "BAI+G_A" in Chain_1 and "BAYI+G_A" in Chain_2. Meanwhile, "BAI" is an independent Mongolian word in the corpus. So Chain_1 can not discriminate the word "BAI" from the morpheme "BAI".

As well known, the translation quality of SMT relies on the performance of morphological segmentation. We give the following example to intuitively show the quality of translation of the chained SMT system.

Example 1 *Table 3 gives four examples of translating Chinese into Mongolian. In each example, four reference sentences translated by native experts are also given. These examples indicate that the chained SMT system can help choose the correct inflections, and partly alleviate the data sparseness problem.*

In Table 3(a), the Mongolian word "HAGAS" (corresponding to the Chinese word "yiban") has multiple inflectional forms as follows:

Mongolian	*Chinese*
HAGAS-VN	*yi bande*
HAGAS-IYAR	*yiban de*
HAGAS-TV	*zaiban*
HAGAS-I	*ba ban*

From the above example, we can see that the baseline system translates the Chinese word "ban" to the incorrect inflection "HAGAS-TV, while Chain_2 translates it to the correct inflection "HAGAS" which is the morpheme of all the other inflections.

Table 3: Examples of translating Chinese into Mongolian

(a) Lexicalization of morphemes

Chinese	xianzai shi jiu dian ban .
Baseline	0D0 B0L YISUN CAG HAGAS-TV.
Chain$_1$	0D0 B0L YISUN CAG HAGAS-TV.
Chain$_2$	0D0 B0L YISUN CAG HAGAS B0LJV BAYIN_A.
References	0D0 YISUN CAG HAGAS B0LJV BAYIN_A.
	0D0 YISUN CAG HAGAS.
	0D0 YISUN CAG HAGAS B0LBA.
	0D0 YISUN CAG HAGAS B0LJV BAYIN_A.

(b) Tense

Chinese	qunian zheshihou ni zai ganshenme ?
Baseline	NIDVNVN ENE HIRI CI YAGV HIJU BAYIHV BVI?
Chain$_1$	NIDVNVN ENE UYES CI YAGV HIJU BAYIHV BVI?
Chain$_2$	NIDVNVN ENE UY_E-DU CI YAGV HIJU BAYIG_A BVI?
References	NIDVNVN-V ENE UYE-DU TA YAGV HIJU BAYIG_A BVI?
	NIDVNVN ENE UY_E-DU TA YAGV HIJU BAYIBA?
	NIDVNVN JIL-VN ENE UYES TA YAGV HIJU BAYIBA?
	0D0 NIDVNVN-V ENE UYE-DU TA YAGV HIJU BAYIG_A BVI?

(c) Syntax

Chinese	wo xiwang jinnian dong tian hui xiaxue .
Baseline	BI ENE JIL EBUL-UN EDUR-UN CASV 0R0JV B0L0N_A.
Chain$_1$	BI ENE EBUL-UN EDUR CASV 0R0HV-YI HUSEJU BAYIN_A.
Chain$_2$	BI ENE EBUL-UN EDUR CASV 0R0HV-YI HUSEJU BAYIN_A.
References	BI ENE EBUL CAS 0R0HV-YI HUSEJU BAYIN_A.
	ENE EBUL CASV 0R0HV-YI HUSEJU BAYIN_A.
	BI ENE EBUL CASV 0R0HV-YI HUSEN_E.
	BI ENE EBUL CAS 0R0HV-YI HUSEJU BAYIN_A.

(d) Out-Of-Vocabulary words

Chinese	wo guoqu chang yidazao chuqu sanbu .
Baseline	... URGULJI yidazao GADAN_A GARCV SELEGUCEN ALHVLABA.
Chain$_1$	... URGULJI BODORIHU-BER GADAGVR ALHVLAN_A.
Chain$_2$	... URGULJI ORLOGE ERTE GARCV ALHVLAN_A.
References	... URGULJI OROLGE ERTE GARCV AVHVDAG.
	... URGULJI ORLOGE ERTE GADAGVR ALHVLADAG.
	... YERU NI ORLOGE ERTE B0S0GAD GADAGVR ALHVLADAG.
	... URGULJI OROLGE ERTE GARCV AVHVDAG.

In Table 3(b), the word “BAYIN” in the result of the baseline system indicates the past tense environment. However, the correct environment is the past continuous tense which is indicated by the word “BAYIN_A” appearing in the results of chain$_1$ and chain$_2$.

In Table 3(c), the baseline system translates “dongtian” into “EDUR-UN” as an attribute, while the correct translation should be “EDUR” as the subject of the object clause.

The statistical data-sets from word alignment corpus show that the vocabularies of the baseline system includes 376203 Chinese-Mongolian word pairs, while Chain$_1$ and Chain$_2$ contain 326847 and 291957 Chinese-Morphemes pairs respectively. This indicates that the chained SMT system can partly alleviates the data sparseness problem. As shown in Table 3(d), the baseline system can not translate the word "yidazao", while Chain$_1$ and Chain$_2$ can.

5 Concluding remarks

The paper proposes the chained SMT system using morphemes as pivot language for translating an isolated language with non-morphology into an agglutinative language with productive derivational and inflectional morphology. The experiment results show that the performance of the chained SMT system is encouraging. And the SMT-based method is quite effective for finding mapping relations between morphemes and words. When adding more features, the precision, recall and F-measure are all improved more obviously.

In the future, we will consider the confusion network or lattice of N-best translation results instead of one best translation result in the chained SMT system. Meanwhile, the distortion of morpheme order in Mongolian is still obscure and needs more investigation. And comparing our work with other language pairs, such as English-to-French translation, English-to-Spanish translation, and so on, is also concerned.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful reviews. The work is supported by the National Key Technology R&D Program of China under No. 2009BAH41B06 and the Dean Foundation (2009) of Hefei Institutes of Physical Science, Chinese Academy of Sciences.

References

[Callison-Burch et al.2009] Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 workshop on statistical machine translation. In *StatMT*, pages 1–28.

[Creutz and Lagus2007] Creutz, Mathias and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *TSLP*, 4(1):1–34.

[de Gispert et al.2009] de Gispert, Adrià, Sami Virpioja, Mikko Kurimo, and William Byrne. 2009. Minimum bayes risk combination of translation hypotheses from alternative morphological decompositions. In *HLT*, pages 73–76.

[Doddington2002] Doddington, George. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT*, pages 128–132.

[Jiang and Zhou2008] Jiang, Long and Ming Zhou. 2008. Monolingual machine translation for paraphrase generation. In *COLING*, pages 377–384.

[Koehn et al.2003] Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*, pages 48–54.

[Koehn et al.2007] Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180.

[Minkov et al.2007] Minkov, Einat, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *ACL*, pages 128–135.

[Nashunwukoutu1997] Nashunwukoutu. 1997. An automatic segmentation system for the root, stem, suffix of the mongolian. *Journal of Inner Mongolia University*, 29(2):53–57.

[Och and Ney2002] Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*, pages 295–302.

[Och2003] Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

[Papineni et al.2002] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

[Quirk et al.2004] Quirk, Chris, Chris Brockett, and William B. Dolan. 2004. Generating chinese couplets using a statistical MT approach. In *EMNLP*, pages 142–149.

[Ramanathan et al.2009] Ramanathan, Ananthakrishnan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT. In *ACL-IJCNLP*, pages 800–808.

[Riezler et al.2007] Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *ACL*, pages 464–471.

[Stolcke2002] Stolcke, Andreas. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 901–904.

[Watanabe et al.2006] Watanabe, Taro, Hajime Tsukada, and Hideki Isozaki. 2006. Ntt system description for the wmt2006 shared task. In *WMT*, pages 122–125.

[Zhang2008] Zhang, Huaping. 2008. ICTCLAS. http://ictclas.org/.

Author Index