STATUS OF THESIS

Title of thesis **ANNOTATED DISJUNCT FOR MACHINE TRANSLATION**

I, TEGUH BHARATA ADJI

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP

2. The IRC of UTP may make copies of the thesis for academic purposes only.

3. This thesis is classified as

□ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

Endorsed by

Signature of Author

Permanent address: Gendeng GK IV/642
RT 69 RW 17 Baciro,
Yogyakarta, Indonesia

Date : 10 May 2010

Signature of Supervisor

Dr. Baharum Baharudin

Date : 10 May 2010

UNIVERSITI TEKNOLOGI PETRONAS

DISSERTATION TITLE:

ANNOTATED DISJUNCT FOR MACHINE TRANSLATION

by

TEGUH BHARATA ADJI

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfilment of the requirements for the degree stated.

Signature: ___________________________________

Main Supervisor: Dr. Baharum Baharudin

Signature: ___________________________________

Head of Department: Dr. Mohd Fadzil Hassan

Date: 10 May 2010

ANNOTATED DISJUNCT FOR MACHINE TRANSLATION

by

TEGUH BHARATA ADJI

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER AND INFORMATION SCIENCE DEPARTMENT

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK

MAY 2010

DECLARATION OF THESIS

Title of thesis: ANNOTATED DISJUNCT FOR MACHINE TRANSLATION

I, TEGUH BHARATA ADJI

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author

Signature of Supervisor

Permanent address: Gendeng GK IV/642
RT 69 RW 17 Baciro,
Yogyakarta, Indonesia

Dr. Baharum Baharudin

Date : 10 May 2010

Date : 10 May 2010

ACKNOWLEDGEMENTS

First of all, this work cannot be accomplished without Allah permission. Allah creates all knowledge in this universe. If the ocean were ink (wherewith to write out) the words of Allah, sooner would the ocean be exhausted than would the words of my Lord, even if we added another ocean. Our knowledge is just as a small drop of water from the ocean. Therefore, all the contributions that I can provide in this work are for the shake of Allah only.

This work can finish in four years with all valuable helps of both my supervisors, Dr. Baharum Baharudin and Ms. Norshuhani Zamin. I would not forget about their patience in guiding me to write the dissertation. I would like to thank Dr. Mohd Fadzil Hassan, Dr. Mohd Nordin Zakaria, and Prof. Alan Oxley for their helpful comments, comprehension, and reviewing my thesis.

Many thanks also go to Professor Daniel Sleator from Carnegie Mellon University as one of the Link Grammar author, Dr. Tang Enya Kong from Universiti Sains Malaysia as one of Synchronous Structured String Tree Correspondence author, and Dr. Mirna Adriani from University of Indonesia as one of experts of Computational Linguistic in Indonesian Language, for their efficient discussion via email during their very busy time.

Special thanks to my family (Antik, Lutfi, and Qornain), for their love, patience and faith throughout all days in three years accompanying me during my doctoral study in UTP.

To all that I mention above and to all that I cannot list individually including all of my friends in Gadjah Mada University and Universiti Teknologi Petronas, surely that your supports cannot be paid with everything, and may those only be rewarded by Allah.

ABSTRACT

Most information found in the Internet is available in English version. However, most people in the world are non-English speaker. Hence, it will be of great advantage to have reliable Machine Translation tool for those people. There are many approaches for developing Machine Translation (MT) systems, some of them are direct, rule-based/transfer, interlingua, and statistical approaches. This thesis focuses on developing an MT for less resourced languages i.e. languages that do not have available grammar formalism, parser, and corpus, such as some languages in South East Asia. The nonexistence of bilingual corpora motivates us to use direct or transfer approaches. Moreover, the unavailability of grammar formalism and parser in the target languages motivates us to develop a hybrid between direct and transfer approaches. This hybrid approach is referred as a hybrid transfer approach. This approach uses the Annotated Disjunct (ADJ) method. This method, based on Link Grammar (LG) formalism, can theoretically handle one-to-one, many-to-one, and many-to-many word(s) translations. This method consists of transfer rules module which maps source words in a source sentence (*SS*) into target words in correct position in a target sentence (*TS*). The developed transfer rules are demonstrated on English → Indonesian translation tasks. An experimental evaluation is conducted to measure the performance of the developed system over available English-Indonesian MT systems. The developed ADJ-based MT system translated simple, compound, and complex English sentences in present, present continuous, present perfect, past, past perfect, and future tenses with better precision than other systems, with the accuracy of 71.17% in Subjective Sentence Error Rate metric.

Index terms: Annotated Disjunct, Hybrid Transfer Approach, Link Grammar, Machine Translation, and Natural Language Processing.

ABSTRACT

Kebanyakan maklumat yang didapati di Internet adalah di dalam bahasa Inggeris. Namun demikian, kebanyakan pengguna Internet di dunia terdiri dari mereka yang tidak menggunakan Bahasa Inggeris. Jadi, adalah lebih baik sekiranya alat mesin penterjemah disediakan bagi mereka. Terdapat pelbagai pendekatan yang telah digunakan dalam membuat mesin penterjemah, antaranya ialah pendekatan “direct”, “rule-based/transfer”, “interlingua”, dan statistik. Fokus dalam tesis ini ialah pembinaan suatu mesin penterjemah untuk bahasa yang tidak mempunyai formula tata bahasa, “parser”, dan corpus, seperti beberapa bahasa di Asia Tenggara. Ketiadaan corpus bilingual ini telah memberikan motivasi untuk menggunakan pendekatan “direct” atau “transfer”. Tambahan lagi, ketiadaan parser dan corpus juga telah memotivasikan membina sistem hibrid antara pendekatan “direct” dan “transfer”. Pendekatan ini dinamakan sebagai “hybrid transfer approach”. Pendekatan ini menggunakan teknik Annotated Disjunct (ADJ). Teknik ini, yang berasaskan kepada formula tata bahasa Link Grammar (LG), secara teori boleh menangani penterjemahan kata satu-ke-satu, banyak-ke-satu, dan banyak-ke-banyak. Teknik ini mempunyai modul aturan alih bahasa yang berfungsi untuk memeta perkataan sumber dalam ayat sumber ke perkataan sasaran pada posisi yang betul dalam ayat sasaran. Aturan alih bahasa tersebut telah digunakan dalam tugasan penterjemahan Bahasa Inggeris → Indonesia. Penilaian eksperimen telah dilakukan bagi mengukur keupayaan sistem tersebut berbanding dengan sistem penterjemah Inggeris-Indonesia yang lain. Sistem penterjemah berasaskan ADJ yang telah dibangunkan ini berjaya menterjemahkan ayat Bahasa Inggeris yang berupa ayat selapis, majmuk, dan kompleks dalam beberapa kala: kini, kini berterusan, kini sempurna, lampau, lampau sempurna, dan kala depan dengan ketepatan 71.17% dalam metrik Kadar Ralat Ayat Subjektif berbanding dengan sistem yang lain.

Indeks istilah: Annotated Disjunct, Hybrid Transfer Approach, Link Grammar, Mesin Penterjemah, dan Pemproses Bahasa Asal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATIONS

ADJ	: Annotated Disjunct
CFG	: Contex-Free Grammar
CG	: Constituency Grammar
CL	: Computational Linguistics
DCG	: Definite Clause Grammar
DG	: Dependency Grammar
EBMT	: Example-Based MT
IE	: Information Extraction
IR	: Information Retrieval
LG	: Link Grammar
MT	: Machine Translation
NER	: Name Entity Recognition
NLP	: Natural Language Processing
POS	: part-of-speech
RBMT	: Rule-Based MT
SL	: source language
SMT	: Statistical MT
SS	: source sentence
TL	: target language
TS	: target sentence

CHAPTER 1

INTRODUCTION

1.1 Motivation

Indonesia is a country in which English is not the first language. As such, the level of English competency among Indonesians is considered low. Considering that vast amount of available digital information nowadays is in English, such as the information in the Internet as global information repository, there is a need to translate this information into the Indonesian language. This goal can be made possible by the development of English-Indonesian MT system. MT is defined as the use of computers to automate some or all of the process of translating from one language to another [58]. Besides three classical approaches for developing MT systems namely direct approach, rule-based/transfer approach, and interlingua approach, there are two other well-known approaches: example-based approach and statistical approach. The use of direct approach for English-Indonesian MT system was done by a research group from Gadjah Mada University, Indonesia [84]. The MT system could solve many translation cases in several tenses such as present, present continuous, present perfect, past, past perfect, and future tenses but the precision was yet to examine. Another MT activity for Indonesian language is the Multilingual Machine Translation System (MMTS) project as part of a multi-national research project between China, Indonesia, Malaysia, Thailand, and led by Japan. This MMTS includes Bahasa Indonesia Analyzer System (BIAS), an analysis component for Indonesian language part [130]. BIAS uses Interlingua approach which takes Indonesian text as input and produces abstract meaning representation, called an Interlingua. Unfortunately, the system accuracy was not provided. The example-based and statistical approaches are categorized as data-driven approaches. Data-driven approaches learn translation information automatically from bilingual corpora (i.e. text that is provided in parallel in two languages). In consequence, these approaches minimize human involvement and are able to achieve rapid development of MT systems within a matter of months, thus overcoming the bottlenecks when using the rule-based approach [94]. Unfortunately, bilingual corpora involving some less-resourced languages (such as languages in South East Asia including Indonesia) are very limited or even none. Contrarily, there are efforts on the development of English-Indonesian MT system using data-driven approaches. The first is the Google Translate application that provides translation from multiple languages to Indonesian as well as from Indonesian to those languages. This application is a statistical approach based on phrase translation [87]. The precision was calculated during this research in BLEU metric and the result was 0.59 for 3-gram precision. The second is English-Indonesian SMT system, which was developed by Agency for the Assessment and Application of Technology (BPPT) and National News Agency (ANTARA) and is based on Pharaoh using 500K sentences pair (current BLEU score 0.72) [97]. Since Indonesian language has the same root and hence shares many aspects with the Malay language, MT studies on Malay languages are also referred. A work in the field of MT was conducted by a research group in the University of Science Malaysia (USM) which uses the EBMT approach to solve English-Malay translation cases [10]. This third data-driven approach-based MT system precision is also yet to question.

In this work, an MT system is specifically developed for scenarios where bilingual corpora are very limited, and where the source language is a major language (English), and the target language is a less-resourced language (Indonesian). The definition of a major/less-resourced language pair in this paper is based on Probst [94]:

- little or no bilingual corpora is available,
- there is no syntactic parser for the less-resourced language.

Bilingual corpora are data that is given in one language with the translation of each sentence or phrase in another language. A syntactic parser is a mean that gives the structural composition or POS of a sentence (e.g. noun, adjective, verb, etc.). The nonexistence of bilingual corpora motivates us to use direct or rule-based approaches, rather than to use data-driven approaches. While the nonexistence of the parser for the target language motivates us to use a hybrid approach between direct and rule-based approaches.

The advantage of this hybrid approach is the use of structural and feature information. This information has been noted by the NLP research community in recent years as an important component for translation quality [94]. Examples of structural information can be understood from problems such as follows.

- How are noun phrases or a sentence constructed in a language?
- How do words and word group orderings change when they are translated into another language?

In solving such problems, the composition of the noun phrase or the sentence is analyzed, and rules are given for how the composition is transferred into another language. For example, the English sentence ‘THIS IS THE CAR’ is considered to consist of two constituents, a noun ‘THIS’ and a verb phrase ‘IS THE CAR’.

Structural transfer information would subsequently address such questions.

- Do the noun and verb phrases appear in the same order in another language?
- Does the verb phrase appear in the same composition, a verb ‘IS’ followed by a determiner ‘THE’ and then a noun ‘CAR’?
- Is there another added word in the target verb phrase, or is there a word in the source verb phrase chopped during the translation?

Although the transfer information or rules become quite complicated if it is applied on sentences of more than 30 words with complex structural composition, structural information is very useful for translation. It allows the decomposition of sentence into meaningful elements (such as noun and verb phrases), that can be composed again in the translation result as a whole sentence.

Example of feature information is as follows. The noun phrase “THE CAR” is singular, as there is only one “CAR”. Feature information then addresses such problems as how singular is expressed in another language and the guarantee that the MT system produces the equivalent of one “CAR” rather than of many “CARS”.

Due to the absence of the parser for the less-resourced language, we make use of:

1) the *SL* syntactic parser that results in the *SL* structural sentences/phrases information which is then used for generating transfer rules that maps *SL* sentences/phrases into *TL* correct sentences or phrases,

2) the available direct method modules from our previous work which consist of unification constraints [84] with some modification.

In our major/less-resourced language pair, a readily available English parser which can deal with complete English sentence structures was used. However, since the Link Parser by Grinberg et al. [47] in LG formalism [109] was used, the derivation of the transfer rules was rather unusual. The difference is that most of the transfer rules in the hybrid approach consider the sentence constituents or dependents. LG does not acknowledge explicit notion on sentence constituents or dependents. In LG, Link Parser is utilized to parse a sentence to obtain a linkage. The linkage contains a sequence of words and a set of links. Each link describes a connection or relationship between two words. The link is expressed as a left connector for a word on its right and as a right connector for a word on its left. A collection of left and right connectors for a word is called a disjunct for that word. This disjunct is considered as one of parameters which contribute to the composition of an ADJ set, besides the corresponding source word and target word. This ADJ set is then used for the development of transfer rules. In a brief explanation, these transfer rules contain LG components (words and their disjuncts) that capture the structural information of a *SS* and map the sentence into a correct *TS*.

Nevertheless, due to the nature of Indonesian language as the *TL* that has no parser available, structure-to-structure mapping cannot be applied. Instead, we must extract transfer rules based merely from the parses of English side, and utilize available English to Indonesian grammatical constraint module taken from previous research using direct approach explained by Novento [84], to be added to the rules.

The initial transfer rules of the ADJ-based MT system were sentence-based since they take into account all disjuncts of all words in a *SS*. However, this needed tedious work in the development of the transfer rules for all cases. In other words, transfer rules generalization for similar cases in the translation process was never obtained. Bond and Shirai [22] also stated that generally rule-based phrase translation gives better sentence translation results. This finding motivates us to incorporate phrase translation method into the ADJ-based MT system. It is done by generalizing the transfer rules with the consideration of phrase-based translations. Moreover, Chiang [26] presented a hierarchical phrase-based MT system that gives higher translation precision than a state-of-the-art phrase-based system proposed by Och and Ney [88]. The last result also encourages us to further incorporate hierarchical phrase translation method into the ADJ-based MT system.

The evaluation and comparison of the developed system is done using human evaluation and automatic MT evaluation since both have advantages and disadvantages. The first evaluation is done since human evaluation on MT measures many aspects of translation including adequacy, fidelity, and fluency. However, it is quite expensive and may take weeks or months. The second is done since MT developers need to monitor the effect of small changes to the MT systems as fast as possible and as cheap as it can. For the second evaluation, an automatic MT evaluation tool based on BLEU metric introduced by Papineni et al. [89] was developed. The evaluation and comparison are done for the sentence-based ADJ system, phrase-based ADJ system, hierarchical phrase-based ADJ system, and other available English-Indonesian MT systems developed by several companies.

1.2 Objectives

The thesis pursues four objectives to be achieved.

1. To explore a bilingual MT method and grammar formalism fits for the task of translating from a major language to a less-resourced language, which yet to have available grammar formalism and parser.

2. To evaluate an algorithm based on the proposed MT method for mapping source sentences into target sentences.

3. To evaluate transfer rules algorithms for target word reordering based on the annotated dictionary.

4. To evaluate an English-Indonesian MT based on the proposed method and to compare with other available MT systems.

1.3 Contribution

The main contributions of this thesis are as follows.

1. A new method of incorporating direct approach into a rule-based/transfer approach so-called ADJ-based method for bilingual MT system.

2. An algorithm for annotating the source words and their word disjuncts with the target words in LG formalism, implemented as ADJ Algorithm.

3. An algorithm which maps from the source sentences into the target sentences in the bilingual MT system, implemented as transfer rules algorithm.

4. An English-Indonesian MT system based on ADJ method.

The minor contributions of this thesis are given in following lines.

1. A transfer rules module manually extracted from the developed English-Indonesian bilingual text, which can easily be adapted for other closely related bilingual MT systems, such as for English-Malay MT systems.

2. An English-Indonesian annotated dictionary which is utilized by the English-Indonesian transfer rules module.

3. An evaluation and comparison method using SSER and BLEU metrics for English-Indonesian MT systems.

4. An automatic MT evaluation tool using BLEU metric.

5. A collection of 450 English-Indonesian sentence pairs as a tool for the development of the transfer rules module and as an instrument to evaluate and compare English-Indonesian MT systems.

1.4 Scope of Study

Research effort presented in this thesis focuses on exploring an MT method on the condition that there is no available bilingual corpus and that the *TL* does not have available grammar formalism and parser. Based on the proposed method, an MT system is then developed to prove that the discovered method works well.

In developing bilingual MTs, native speakers or linguists of the *SL* and *TL* are mostly involved in the bilingual corpora construction [85], grammar analysis, and evaluation process [89]. To reduce the cost of hiring linguists in addition to make the scope achievable for producing this thesis, coupled with the availability of four Indonesian language native speakers with enough experiences in taking courses involving both English and Indonesian grammar analysis, an English-Indonesian MT system is thus developed for a case study.

However, open-domain MT system is difficult to build [117]. Hence a particular domain is suggested in developing the English-Indonesian MT system. In this thesis, a domain of story books for elementary students is chosen since the system is targeted to be used by Indonesian people, who are in the basic level of English proficiency and still understand limited tenses such as the present, present continuous, present perfect, past, past perfect, and future tenses. The MT system is still considered as an initial version which still has a limited dictionary (3000 pairs of common English-

Indonesian words). This makes this initial version appropriate to users in elementary schools.

Bilingual English-Indonesian corpora are an appropriate mean for the development of a SMT system. These corpora can also be utilized for constructing transfer rules module in a hybrid transfer MT system. Nevertheless, publicly available bilingual corpora of both languages are not available at the time this thesis is written. Thus, we merely developed sparse bilingual English-Indonesian sentence pairs (i.e. 450 sentence pairs), which comprise 300 sentence pairs for English to Indonesian grammar analysis and 150 sentence pairs for translation evaluation process.

The developed MT system is not design for translation tasks of complex English sentences in all possible tenses. In fact, only sentences in present, present continuous, present perfect, past, past perfect, and future tenses can be handled by the system. The system is also not targeted for translating sentences, which consist of sayings, idioms, proverbs, and ambiguous words.

1.5 Organization of the Dissertation

The remainder of this thesis is arranged as follows.

Chapter 2 describes three grammar formalisms (dependency, constituency, and link grammars), less-resourced language research activities, research in NLP of Indonesian language, followed by four machine translation approaches (direct, transfer, interlingua, and statistical approaches) and English-Indonesian MT system developments.

In Chapter 3, the proposed ADJ method for the MT is explained. The explanation covers the proposed MT schema, the proposed MT system with the case study on English-Indonesian MT system, transfer rules of the developed English-Indonesian MT system, and the mechanism of the hierarchical phrase-based transfer rules.

In Chapter 4, the experimental setup for conducting this research is explained to allow other academicians or researchers to understand the data collection, tools such as dictionaries used for the MT system, the developed MT system set up, and two metrics to evaluate the developed system and to compare with other available systems.

Chapter 5 evaluates three kind transfer rules of the developed MT system, namely sentence-based, phrase-based, and hierarchical phrase-based transfer rules. A summary of all the results is also given.

Chapter 6 summarizes the proposed hybrid transfer MT method along with its transfer rules and its implementation on an English-Indonesian MT system. Several contributions and limitations of the research as well as future works to the development of the MT system are put across.

CHAPTER 2

LITERATURE REVIEW

The beginning of this chapter (Section 2.1) explains the three well-known grammar formalisms. These three formalisms are frequently used as platforms for CL or NLP related research activities such as for developing POS, parsers, and MT systems. One of the formalisms is chosen as the base of our approach to develop the ADJ method. Section 2.2 presents some related works on NLP research activities for less-resourced languages other than Indonesian language to give a comparative study which in turn bring up ideas to NLP researchers on how they should invest for Indonesian language technology provision, in particular MT. NLP research of Indonesian language other than MT such as corpus analysis and morphological analysis is discussed in Section 2.3 to list the available Indonesian language technology resources, which are useful for developing other applications or systems such as MT system. The three classical approach to MT (direct approach, transfer approach, and interlingua approach), statistical approach, and hybrid approach are then discussed. The underlying needs in terms of resources for these approaches are identified in Section 2.4. Research effort for Indonesian MT is given in Section 2.5.

2.1 Grammar Formalisms

Grammar formalism is an effort of introducing formal mechanisms for capturing grammatical knowledge of a natural language. Grammar is a branch of linguistics that deals with syntax and morphology. The word syntax can be rooted from the Greek “syntaxis”, which means arrangement. Thus, syntax can be understood as the way words are arranged together [55]. In the next sub sections, three grammar formalisms: dependency grammar, constituency grammar, and link grammar will be discussed.

2.1.1 Dependency Grammar (DG)

DG is an intuitive and the least famous grammar concept. In DG, one word form depends on the other. In other words, individual word both acts as terminal node and as non-terminal node. The words are terminal because they directly access the lexicon. Dependency only recognizes words in its purest form. The words are also considered as non-terminal because they “subcategorize” other words, so-called dependents [104]. DG has been less known among linguists than CG more recently, especially since the start of modern grammar theory. DG is also considered as an old concept explained as follows.

> “Dependency analysis’ is an ancient grammatical tradition which can be traced back in Europe at least as far as the Modistic grammarians of the Middle Ages, and which makes use of notions such as ‘government’ and ‘modification’. In America the Bloomfieldian tradition (which in this respect includes the Chomskyan tradition), assumed constituency analysis to the virtual exclusion of dependency analysis, but this tradition was preserved in Europe, particularly in Eastern Europe, to the extent of grammar teaching in schools. However, there has been very little theoretical development of dependency analysis, in contrast with the enormous amount of formal, theoretical, and descriptive work on constituent structure.” [51]

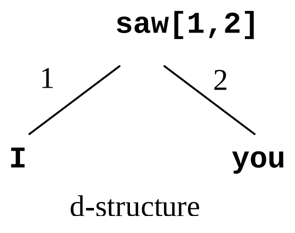


Figure 2.1: An illustration of DG

Figure 2.1 is a representation of a dependency structure (d-structure) of a sentence “I saw you”. In this representation, the head (the word “saw”) is placed above its dependents (the words “I” and “you”). The numbers in square brackets ([1, 2]) show the number of dependents or arguments in a logical representation.

Dependency is an asymmetrical connection between a head and a dependent. It forms a vertical organization principle where heads and dependents are related immediately since there are no terminals [66]. The non-existence of terminals led many dependency grammarians to claim that DG is more economic than CG [75], [106]. If heads and dependents are put together then there exist dependency structures, which have the following constraints [66].

1) There should be one independent element.

   Every word must depend on some other words, with the exeption of one element – the root.

2) All dependency structures must be connected.

   All the words should be connected by the same one structure.

3) Every dependent must possess a unique head.

   Each dependent must depend exactly on one head, except for the root.

4) Heads must be adjacent to dependents.

There are three types of syntactic relations in DG.

1) *Connection*

   This relation, which corresponds to dependency, is the most basic relation between words [121]. A connection is visualized using a stemma, a straight line between the head and its dependent (see Figure 2.3).

2) *Junction*

This type of relation is used to relate elements on the same level [121], i.e. non-dependently elements which poses major problems in dependency [104]. An example that needs junction exists in the sentence “Lutfi and Qornain saw you” as shown in Figure 2.2. Qornain does not depend on Lutfi, and vice versa. Therefore, both words need to appear at the same level as indicated by ‘j’ (stands for junction) line, which shows a junction between Lutfi and Qornain.

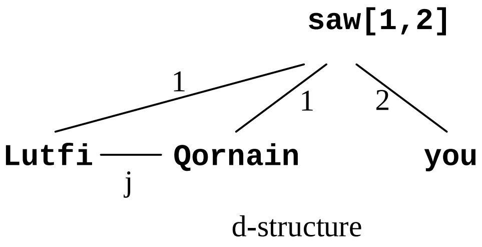
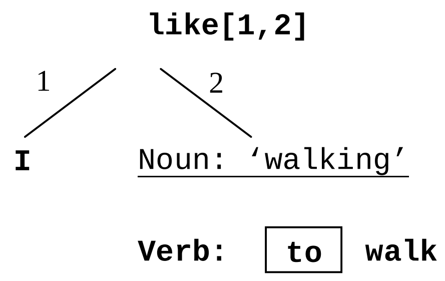


Figure 2.2: A junction in DG

3) *Translation*

This relation type allows the explanation of words with other words of other word classes in syntactosemantic positions and functions [121]. In the sentence “I like to walk” in Figure 2.3, the infinitive form “to walk” can be explained with or translated into the gerund “walking”. The bar symbolizes the translation wherein the quoted element is used for an explanatory purpose. The boxed element is so-called ‘translative’ which triggers the translation. In this case, “to” triggers the translation to a noun.



d-structure

Figure 2.3: A translation in DG

Some MTs and parsers were developed based on DG formalism such as a method for MT so-called Synchronous - Structured String Tree Correspondence (S-SSTC) [10], a Korean-English MT system which starts from parsed bilingual (Korean-English) text to induce mapping rules [68], a description of 500,000 word Prague Dependency Treebank for Czech [48] which has been used to train probabilistic dependency parsers [28], a parser for discontinues constituents in DG [29], and an online functional dependency parser of English developed by Helsinki University [54]. Schneider [104] already tested the last parser and found that its coverage is broad but slightly below Link Grammar explained in Sub Section 2.1.3. He also added that dependency analyses are much more functional than those of Link Grammar. In other words, functional terms subject, object, attribute, modifier, and complement are used very consistently.

2.1.2 Constituency Grammar (CG)

In constituency, a sentence consists of certain elements which in turn consist of other elements or words. The usual definition is that a constituent consists of any word plus all its dependents, their dependents, and so on recursively. In other words, groups of words may behave as a single unit or phrase, called a constituent. For example, a group of words called a noun phrase can acts as a unit that include single words like “she” or “John” and phrases like “the tree” and “Indonesian books”.

It must be noted here that DG still recognizes constituents, but they are a defined rather than a basic concept [30]. Another distinguishing factor between the CG and DG is that CG is a horizontal organization principle which groups together constituents into phrases (larger structures) until the entire sentence is accounted for [66]. Figure 2.4 is an illustration of a constituency structure (c-structure) of the sentence “I saw you”.

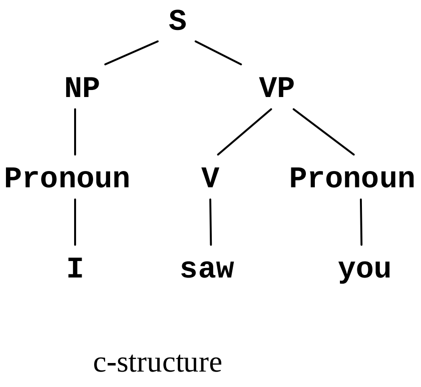


Figure 2.4: An illustration of CG

The CG was not formalized until its appearance in Chomsky [27] and independently in Backus [19]. The most commonly used mathematical model for CG is the CFG. CFG is widely used for syntactic description of constituent structures and other structures as well e.g. the syntax of programming language. CFG are also called Phrase-Structure Grammar (PSG), and the formalism is equivalent to Backus-Naur Form (BNF), and widely implemented in Prolog syntax rules so-called DCG. CFG consists of a set of rules or productions for expressing the grouping and ordering of language symbols and lexicon. Each grammar must possess one designated start symbol, which is often called S. Since CFG are often used to define sentences, S is usually interpreted as the sentence node. Examples of the rules / productions in CFG are given in Figure 2.5.

S → NP VP

NP → Pronoun

NP → Det N

N → SN

N → PN

VP → V

VP → V NP

Figure 2.5: Rules in CFG

The rules express that S is formed by a noun phrase (NP) followed by a verb phrase (VP). An NP can be composed of either a Pronoun or a determiner (Det) followed by a noun (N). An N can be a singular N (SN) or plural N (PN). A verb phrase can be made up by a verb (V) or V followed by NP. Usually the rules are combined with facts about lexicon as shown in Figure 2.6. The symbols used in a CFG are divided into two classes. The symbols that correspond to words in the language (“I”, “saw”, “you”, etc.) are called terminals e.g. Pronoun, Det, SN, PN, and V. The facts about the lexicon consist of these terminal symbols. The symbols that express clusters or generalizations of terminal symbols are called non-terminals e.g. S, NP, VP, and N. In each rule, the item to the right of the arrow (→) is an ordered list of one or more terminals and non-terminals, while to the left of the arrow is a single non-terminal symbol expressing some cluster or generalization. Take note that in the facts about the lexicon, the non-terminal associated with each word is its lexical category, or POS.

Pronoun → “he”

Pronoun → “I”

Pronoun → “it”

Pronoun → “we”

Pronoun → “you”

Det → “a”

Det → “the”

SN → “pen”

SN → “tree”

PN → “books”

V → “like”

V → “saw”

V → “see”

Figure 2.6: Facts in CFG

A CFG can be thought of as two devices: a device for generating sentences and a device for assigning a structure to a given sentence. The sequence of rule expansions generated by a CFG is called a derivation of the sentences. For example, the derivation of the sentence “I saw you” is given as follows.

S → NP VP → Pronoun VP → “I” VP → “I” V NP → “I” “saw” NP

→ “I” “saw” Pronoun → “I” “saw” “you”

The derivation is common to be represented by a parse tree such as illustrated in Figure 2.4.

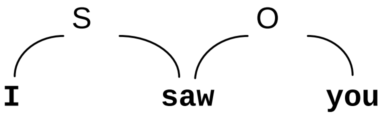
Several parsing approaches for CFG are available from the deep parsing such as Cocke-Kasami-Younger (CKY) algorithm [59], [129]; the Earley algorithm [36]; the

Chart Parsing algorithm [58], [61], and then continue to the partial or shallow parsing such as finite-state parsing models [1], [37]. Much recent work on shallow parsing applies supervised machine learning techniques to learn patterns e.g. reports by Ramshaw and Marcus [96], Argamon et al. [15], and Munoz et al. [77]. Since CFG is well-known as a modern grammar formalism, hundreds reports have been made on the development of MT system based on CFG, such as briefly described in the following lines. Nakamura et al. [78] developed a bidirectional Japanese-English MT system which utilizes two different transfer rules, which are Japanese-to-English and English-to-Japanese. The rules were expressed in tree-to-tree transformation that also consideres tree constituent levels of both languages. Kaji et al. [56] presented an MT system that learns transfer rules from of constituent trees in an EBMT framework. The training data is parsed bilingual text and an algorithm aligns the constituent trees and extracts transfer rules. A few years later, an MT system called PalmTree was built by Watanabe and Takeda [120]. This machine also used transfer rules but employed pruning techniques in the beginning and introduced example-based processing in the end of the pattern matching. Yamada and Knight [128] incorporated a decoder to find a best English parsed-tree given a Chinese sentence in a syntactic phrase-based statistical MT. The task of generating the tree structure became available with the use of a parser and training corpus, which consists of English parsed-trees (in CFG) and foreign sentences. Other examples include a DCG-based bidirectional German-English MT system [82] and a DCG-based English-Arabic Noun Phrases MT system [107]. Bond et al. [23] presented Head Driven PSG-based Japanese-English MT prototype that uses developed parsers, bidirectional grammar, transfer rules and target sentence generators. Chiang [26] reported the state-of-the-art syntax-based SMT, which was able to automatically learn transfer rules from bilingual text without syntactic annotation and then formalized the extracted rules in the form of synchronous CFG. In the meantime, Venugopal et al. [118] introduced two stages to lessen the computation of intersection between an n-gram Language Model (LM) and a Probabilistic Synchronous Contex-Free Grammar (PSCFG) for SMT. The first stage is the generation of first-best approximations by using CKY-style decoder and the second stage is the use of n-gram LM to recover the search errors made in the first stage. Zollmann et al. [135] developed an open-source Syntax Augmented MT (SAMT) based on PSCFG for SMT. The system was tested on an unseen Spanish-

English corpus after trained on 2000 sentence. A BLEU score of 32.15% was achieved and was comparable to a state-of-the art phrase-based SMT system with POS based-word reordering CMU UKA ISL system [90], which achieved 31.85% in the same test.

2.1.3 Link Grammar

Link Grammar is a formal grammatical system. This formalism was already described in detail by Sleator and Temperley [109]. In some reports, Link Grammar was categorized as DG [104], [55]; although it is still debatable. Figure 2.7 represents an LG structure (l-structure) of "I saw you", which has two links. One link connects a subject noun to a finite verb (*S* link) and the other link connects a transitive verb to its object (*O* link).



l-structure

Figure 2.7: An illustration of Link Grammar structure

Another l-structure is given in Figure 2.10 for an English sentence "John pick the heavy box up", which is taken from Al-Adhaileh et al. [10]. LG formalism consists of a set of words, where each word has a linking requirement. This linking requirement is expressed as a formula involving the operators &, or, parentheses, and connector names. The + or – suffix on a connector name indicates the direction of how the matching connector must lie. Let consider some words: "John", "Mary", "picks", "the", "a", "heavy", "green", "box", "cat", "snake", and "up" with their linking requirements. The linking requirement of each word in the LG is illustrated by the labeled object(s) above the word (see Figure 2.8). The labeled object(s) connected to each word represents the connector of the word. A connector is satisfied by matching it to a proper connector with the appropriate shape facing in the opposite direction. Thus, the word "box" requires an *A* connector to its left (or simply as *A*-), a *D* connector to its left (*D*-), and either an *O* connector to its left (*O*-) or an *S* connector to its right (*S*+).

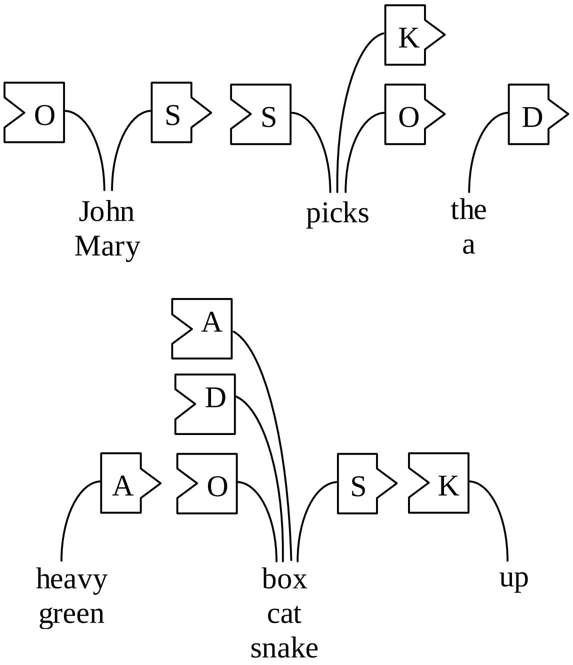


Figure 2.8: Linking requirement diagram for each word in Link Grammar

The linking requirements are expressed in a list of words and their formulas, as written in the dictionary in Table 2.1.

Table 2.1: Linking requirements dictionary of the Link Grammar expressed in each word and its formula

Word(s)	Formula
John Mary	O- or S+
picks	S- & O+ & {K+}
the a	D+
heavy green	A+
box cat snake	{@A-} & D- & (O- or S+)
Up	K-

The & operator of two formulas necessitates both formulas to be satisfied. Whilst the **or** operator of two formulas requires exactly one of its formulas to be satisfied. The order of the arguments of **&** operator is important. The more left the connector in the expression, the nearer the word to which it connects will be selected. Hence, for the word "box", its adjectives must be closer than the determiner. The notation "{exp}" describes the exp expression is optional. "@A-" means one or more *A* connectors may be connected to its pair. *A* connector connects adjectives to nouns, *D* connects determiners to nouns, *O* connects verbs to nouns, and *K* connects certain verbs to particles. Figure 2.9 shows one example of a sentence in LG, "John picks the heavy box up", which satisfies the linking requirements (see Figure 2.8).

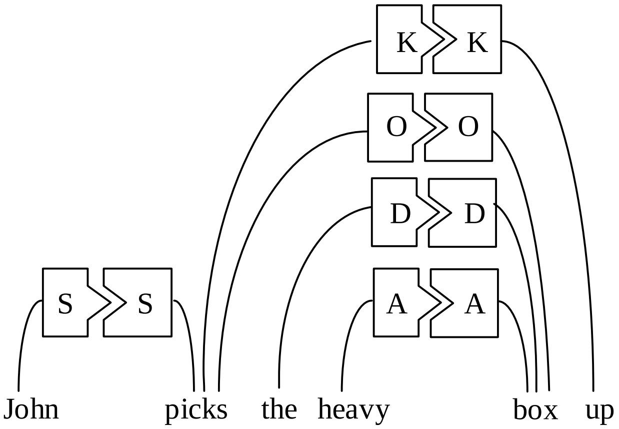


Figure 2.9: The sentence "John picks the heavy box up" in LG

A set of links which proves that a sequence of words is in the language of a LG is called a linkage. Figure 2.10 is the simpler diagram to illustrate the linkage of "John picks the heavy box up".

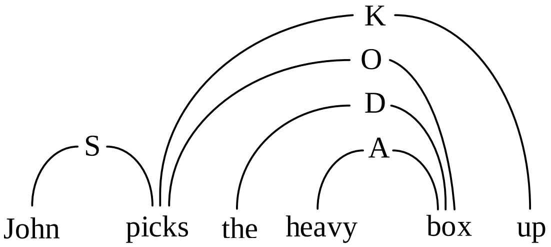


Figure 2.10: The linkage of the sentence "John picks the heavy box up"

It is more convenient for mathematical analysis to rewrite a formula of a word in a disjunctive form. In this disjunctive form, instead of having the formula of each word, the word is considered as a list of disjuncts as formulated in Equation (2.1) [47].

$$d = ((L_1, L_2, \ldots, L_i, \ldots, L_m)(R_n, R_{n-1}, \ldots, R_j, \ldots, R_1)) \qquad (2.1)$$

where L_i is left connector and R_j is right connector.

Hence, the formula of the word "John" in Table 2.1:

O- or *S*+

is considered to have two disjuncts as follows:

((*O*)( )),

(( )(*S*)).

While the formula of the word "picks":

S- & *O*+ & {*K*+}

has the following two disjuncts:

((*S*)(*O*, *K*)),

((*S*)(*O*)).

The formula of the word "the":

D+

can generate a disjunct of:

(( )(*D*)).

Whereas the formula of the word "heavy":

A+

obtains the following disjunct:

(( )(*A*)).

The formula of the word "box":

{@*A*-} & *D*- & (*O*- or *S*+)

will have four disjuncts as follows:

((*A*, *D*, *O*)( )),

((*A*, *D*)(*S*)),

((*D*, *O*)( )),

((*D*)(*S*)).

The formula of the word “up”:

K-

can derive the following disjunct:

((*K*)( )).

Several researches were done based on LG formalism. Venable [117] reported the use of LG to develop an MT system. The work used bilingual corpora to build a bilingual statistical parsing system that can infer a structural relationship between two languages. This model included syntax, but did not involve word-segmentation, morphology and phonology. One parser available in LG formalism so-called Link Parser was also used for different research area namely NER such as explained by Sari et al. [102] and IE such as explained by Zamin [131].

2.2 Less Resourced-Language Research Activities

The emergence of Internet as a universal information repository, in which all kind of information is stored, has triggered the abundance of information retrieved. However, the rising amount of information coupled with the need of automated analysis to those collected information, requires the advancement of intelligent information processing tools. Owing to the use of human language as the representation of information, a computer formulation of human language is quite a challenging task to undertake. Language technology researchers have given noteworthy fruitions on formulating human language, either majority languages or less-resourced languages, by means of CL and NLP, ranging from search engine to knowledge management application, from information technology to medical domain. Those researchers focus mostly on formulating major languages, which are widely used in the Internet or other digital documents. Languages which are categorized as majority languages are reflected from a comprehensive study reporting that 71% of the pages in the Internet (453 million out of 634 million Web pages indexed by the Excite search engine) were written in English, followed by Japanese (6.8%), German (5.1%), French (1.8%), Chinese (1.5%), Spanish (1.1%), Italian (0.9%), and Swedish (0.7%) [126].

Nevertheless, thousands of less-resourced languages, which are not widely used in the Internet or other digital documents, are considerably seldom to be used in CL or NLP research areas. Most of less-resourced languages do not have available digital resources such as POS tagger, grammar formalism, parser, and corpus. Hence, research on developing digital resources for less-resourced languages is encouraged by many research groups and conferences in recent year. Some less-resourced languages have been well researched. Nonetheless, most of them are considerably rarely investigated linguistically. Furthermore, they are politically lack of recognition and are under increasing pressure from the major languages (especially English), as explained by ISCA (International Speech Communication Association) in www.lrec-conf.org/lrec2008/IMG/ws/lrec2008-saltmil-cfp.pdf.

Forcada [41] mentioned that less-resourced language is closely connected to minority language. He also explained that minority language has the following characteristics:

- small number of speakers,
- used far from normality (used more at home than in school or administration, socially discriminated, politically repressed, etc.),
- lacking a commonly accepted writing system, spelling, or reference dialect,
- limited presence on the Internet,
- lacking linguistic expertise,
- lacking machine-readable resources: dictionary, corpus, POS tagger, etc.

Particularly, the absence of language resources (such as word stemmer, lexicon, POS tagger, dictionary, corpus, parser, grammar formalism, etc) in a less-resourced language would make difficulties on any NLP-related commercial product development. For example, word stemmer is a very important means to build an IR system for both complex agglutinative languages (such as Turkish) and languages which have relatively simple morphology (such as English). Almost all IR system needs word stemmer, since every single word in a phrase that need to be retrieved can actually be in the form of hundreds or even thousands of its variant [112]. Thus, this stemming process – a computational procedure that reduces the word variant to get its root word by applying morphological rules – will help to enhance the recall of a search [42], [65], and [69]. Hence, research on building word stemmer and other linguistic resources especially for less-resourced languages is encouraged by many research groups and conferences in recent years. The effort aims to share information on tools and best practices, so that isolated researchers will not need to start from scratch. This also minimizes duplication of research. Some group discussions already highlighted research activities on less-resourced languages. In 2006, ISCA (International Speech Communication Association) special interest group on Speech and Language Technology for Minority Languages (SALTMIL) held a workshop on "Strategies for developing machine translation for minority languages" in Italy. Meanwhile, a special session entitled “Speech and language technology for less-resourced language” is held in Interspeech 2007 conference in Belgium. Several publications also discussed less-resourced language processing as follows.

In developing word stemmer, a research work on Turkish reported that a morphological analyzer is required to achieve high quality stemming since this language employee complex agglutination which can result in long words that can contain as much semantic information as a whole English phrase, clause, or sentence [38]. Another research reported the effectiveness of word stemmer usage in Amharic (a Semitic language spoken in North Central Ethiopia by the Amhara) IR system [11], [12]. The result was obtained via a comparative study between stem-based and conventionally word-based searching of Amharic texts. Other word stemmer development report for less- resourced languages can be found in Popovic and Willett [92] for Slovene; in Ahmad et al. [7] for Malay; in Al-Kharashi and Evens [13] and Abu-Salem et al. [2] for Arabic; in Kalamboukis [57] for Greek; and in Solak and Oflazer [110] for Turkish.

In lexicon development, Berment [21] reported a collaborative work for building Lao (the language spoken by about 4 million people in Laos and by more than 10 million people in Thailand) lexical base using pivot approach. In this pivot approach, a web-based interface with a pivot is developed to provide other researchers to contribute their own language lexical base. This project, which is called PapiLex, is in the context of Papillon project and follows the fundamental rules of this project:

- lexical base in XML format,
- use of the explanatory and combinatorial lexicology (ECL) concepts (from which the core monolingual Papillon XML schema is directly derived),
- use of Unicode for the characters encoding.

This collaborative approach would prevent the dependency of huge texts and dictionaries which are limited and lacking for minority languages such as the Lao language.

In building a POS tagger, a research work was conducted by reviewing an unsupervised method to obtain POS tagger which in turn is used within the Apertium MT engine in order to produce Occitan-Catalan language pair translation. The experimental result shows that the amount of corpora required by this method is small compared with the usual corpora sizes needed by the standard method which does not embed the resulting POS tagger. Therefore, this method is appropriate for training POS tagger to be used in MT for less-resourced language pairs [101].

In developing dictionary, Max Planck Institute for Psycholinguistics created a multimedia dictionary of the Marquesan and Tuamotuan languages of French Polynesia which is called LEXUS. LEXUS allows the user to create semantic networks which are able to visualize the relationship between objects and entities in directed graphs [24]. A project called ReTraTos expected to automatically build linguistic knowledge – bilingual dictionaries and shallow transfer rules – from Brazilian Portuguese to both languages: Spanish and English. This linguistic knowledge will be useful for machine translation. The knowledge extraction is made possible through the use of word-aligned parallel corpora (Brazilian Portuguese-Spanish and Brazilian Portuguese-English parallel text) processed with shallow monolingual resources: morphological analyzer and POS taggers [25]. Other several methods for automatic bilingual dictionary builders have been proposed in Schafer and Yarowsky [103], Fung [43], Koehn and Knight [63], Langlais et al. [67], and Wu and Xia [125].

In corpus development, Ghani et al. [45] reported a technique to automatically collect Web pages in minority languages (Slovenian, Croatian, Czech, and Tagalog). This technique requires the user to supply a handful of documents or keywords. The documents are categorized into relevant or irrelevant with the target language, whilst specific terms (keywords) are categorized into inclusive or exclusive. The inclusive keywords are highly unique to the target language while the exclusive keywords are unique to irrelevant languages. This technique examines all the current documents to generate query terms (based on the frequency of inclusive and exclusive keywords) to find another document in the Internet which is similar to the relevant documents and not similar to the non relevant documents. The query terms are updated every time a new relevant document is obtained, to be used for the next relevant document searching process.

In developing a parser, Venable [117] found that developing a rule-based parser or tediously annotating huge data manually to train a statistical parser are no more interesting since both approaches requires extra works of linguists. He then came up with the idea of using an aligned bilingual (source and target languages) corpora to understand the relationship between the structure of *SL* with available parser (e.g. English) and target language. The English structure, which is generated by the English parser, is then transferred over to the target language across the bilingual corpora to automatically annotate target language sentences. These annotated sentences are used as training data for the target language new parser. This work is done without the need of linguists to develop grammar rules or to annotate data.

In the meantime, developing grammar for less-resourced languages is rather unappealing since it requires a large amount of work by computational language experts. Such work is very much correlated with the availability of grammatical and lexical resources for the target language. The expert then needs to study and formalize the lexicology and morphology of that language. Contrarily, Maxwell [73] put an effort on incremental grammar development, an approach suitable for minority languages. This paper explained the possibility of employing a linguist who merely knows little about a particular computational tool. This method works with incrementally building a grammar and dictionary based on a very small (but growing) text corpus with only a few thousand words, and no grammar or dictionary.

Consequently, grammar checking and revisions are very crucial for this work. The main theme of this research is morphological grammar development, focuses in morphemes dictionary, morphosyntactic and phonological rules. This approach was funded by Xerox Research Centre Europe and the information about the work can be obtained at http://www.xrce.xerox.-com/competencies/content-analysis/fsmbook/.

At the time this thesis is written, very limited reports discussed MT development for less-resourced languages. This is probably because most MT approaches require language technology resources such as bilingual corpora, bilingual dictionaries, or parsers of the languages. Senellart et al. [105] presented a hybrid MT system called Systran. Similar to the direct system, Systran relies on a bilingual dictionary, which has lexical, syntactic, semantic knowledge, and does word reordering in a post-processing step. However, similar to a transfer system, many of the steps are informed by syntactic and shallow semantic processing of the *SL*. Somers [111] reported a way to figure out an approach of MT methods for a major/less-resourced language pair, which is English-Welsh translation. The methods involved RBMT, SMT and a famous hybrid approach called EBMT, which can be seen as a hybrid of RBMT and SMT. EBMT approach involves the matching of the input against a database of real translation examples, and identifies the closest matches. The proposed matches are up to an automatic process, which identifies corresponding translation phrases and then recombines all phrases to give the target text. Probst [94] attempted to infer transfer rules automatically from small bilingual texts by using a variety of information, such as a parser that is available for one language of the bilingual language, and morphological information that is available for both languages. The transfer rules are learned in three phases, first producing an initial hypothesis, then capturing the syntactic structure, and finally adding appropriate unification constraints. The approach was demonstrated by the effectiveness of the learned rules on Hebrew→English and a Hindi→English translation tasks. Vandeghinste [116] described a prototype of hybrid MT called METIS-II in which SMT, EBMT, and RBMT were used and only minimal resources for both source and target languages were utilized. The resources include a shallow source language analysis, a translation dictionary, an *SL*-to-*TL* mapping system, and a target language corpus for generation. The languages involved in this MT prototype were Dutch, Greek, German, Spanish, and English. Another effort was done by researchers in USM where English-Malay MT was developed by using EBMT method. This method depends on a corpus of already existing translations, which is reused as the basis for a new translation. It involves matching of the input against a database of real translation examples by means of an automatic process that identifies corresponding translation phrases and then recombines all phrases to give the target text. The database was in the form of Bilingual Knowledge Bank containing bilingual sentences, which was annotated with their dependency structures by using SSTC [114].

2.3 Research in NLP of Indonesian Language

The Indonesian language so-called Bahasa Indonesia is the unified language for over 230 million citizen of the Republic of Indonesia, the fourth most populous country in the world and a close neighbour of Australia. Bahasa Indonesia is the official language for the country. It has the same root as the Malay language. Hence it is closely related to the Malay language spoken in Malaysia, Singapore, and Brunei. In the past before their official status, both Indonesian and Malay languages became the lingua franca for people throughout Nusantara. Nusantara is composed from two words "nusa" and "antara" which mean an archipelago located between two oceans Asia and Australia.

Indonesian language uses Roman script with 26 letters as in the English alphabet. The basic sentence order is Subject-Verb-Object. Verbs are not inflected for person or number. The language also does not distinguish tenses. Tense is denoted by the time adverb e.g. "kemarin" (yesterday) or some other tense indicators e.g. "sudah" (already). These tense indicators can be placed at the front or end of the sentence [95]. Indonesian language does not have gendered words. Plural nouns are usually expressed by word repetition. It is also a member of the agglutinative language family that has a complex range of affixes attached to base words. Thus, an Indonesian dictionary usually contains root words and base words with affixes with their different translations. Indonesian language employs affixes more heavily than English. Besides prefixes and suffixes, Indonesian language has infixes (insertions) and confixes (circumfixes). Confixes are combinations of prefixes and suffixes [4].

Indonesian language is found to be present in 1,000,000,000 documents on the public World Wide Web [53]. It is thus considered as a major language. On the contrary, the fact that the Indonesian language has yet to have a POS tagger, parser, and grammar formalism makes the language be categorized as a less-resource language. Therefore, developing a systematic understanding of Indonesian language is definitely a necessity. This systematic understanding of language, also known as language grammar formalism, will enable the development of various systems that are beneficial to other information technology areas, such as Question-Answering Algorithm, MT, Voice Dialogue Algorithm, IR, and IE. Unfortunately, there are inadequate research activities in Indonesian language with regards to computational linguistics. On one hand, Indonesian linguists seem to be keen on working "manually" instead of using computers in conducting their linguistics research activities [76]. On the other hand, most computer scientists working on this area do not tend to take into account complete framework to formulate the language. Thus, with the aim to link up computer science with the discipline of linguistic, several NLP activities in Indonesian language have been conducted in the following four areas, i.e. corpus analysis, morphological analysis, information retrieval, and machine translation, as shown in Figure 2.11.

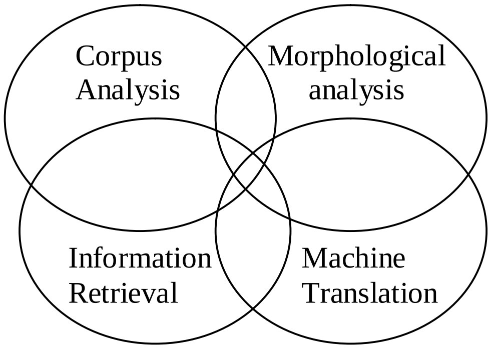


Figure 2.11: Research fields on CL for the Indonesian language

The first area of NLP research for the Indonesian language is corpus analysis, a study to understand the evolution of language usage by its people. In the case of

Indonesian language, research activities on corpus analysis were very limited. There was one work at Monash University which conducted word frequency analysis of Indonesian newspapers [49]. A group of researchers from the University of Indonesia conducted an intensive Indonesian corpus analysis using newspapers as the text source [79]. They collected 52 editions of Kompas, a national newspaper. From this collection, they constructed a corpus consisting of 2,200,818 words that were formed by 74,559 unique words. 1,826,740 words of them that were formed by 27,738 unique words are actually words that matched with Kamus Besar Bahasa Indonesia (KBBI) entries while the unmatched words are either names or foreign words. KBBI is the standard dictionary for the Indonesian language which contains more than 70,000 words. A research collaboration by the Indonesian Agency for the Assessment and Application of Technology (BPPT) and National News Agency (ANTARA) developed parallel corpora for supporting Bahasa Indonesia Analyzer System (BIAS)-II, an analysis system for Indonesian language suitable for multilingual MT system [97]. There are also online monolingual corpuses such as Tempointerakif.com (56,471 articles) and Kompas corpus (71,109 articles), which can be found at http://ilps.science.uva.nl/Resources/BI. A joint research between Telkom RDC and ATR-Japan has constructed speech corpus from 42 speakers (20 males and 22 females) with each speaker uttering 510 basic travel expression corpus (BTEC) sentences resulting in a total of 21,420 utterances (23.4 hours of speech). The research also achieved the optimum performance of automatic speech recognition on the BTEC to 92.47% word accuracy [100].

The second NLP done for Indonesian language is morphological analysis, a study to understand how a root word can change into its derived word [4]. Bali and Mohamad [20] created a program to determine whether a text is Malay or Indonesian using the criteria at the character level, criteria at morphology level, and criteria at the lexicon level. Another implementation of this research activity is spelling checker. This tool helps users of word processors in producing an error-free document. To develop a spelling checker, it is necessary to understand the morphological structure of words especially how derived-words are constructed from their root-words and the addition of affixes. An example of a spelling checker and spelling-error corrector utilities was developed at University of Indonesia as part of the Lotus Smartsuite3 package [79]. There is an implementation of morphological analysis so-called word stemmer, which stems a derived Indonesian word in order to obtain the root-word. Unlike English, where the role of suffix dominates the generation of derived-words, Indonesian language depends on both prefix and suffix to derive new words. In addition, similar to English, multiple suffixes can also be present on a given derived-word. Hence, to stem a derived Indonesian word, presence of both prefix and suffix in that derived-word must be taken into account.

Several researches have been conducted on stemming Indonesian words. Most of the researches were to define manually a set of stemming rules to find the possible roots of a word [53].

There are two kinds of words stemmers:

1) dictionary-based stemmers, i.e. the stemming process is assisted by a dictionary to check whether the candidate stem is a valid Indonesian word,
2) non dictionary-based stemmer.

The choice between the two kinds of word stemmers depends on the purpose and needs of the application that utilizes it. In a stemmer built for an IR system, a root word found by the stemmer may be invalid (i.e. not found in the dictionary). However, this is not a big issue since the IR system concerns more on finding the presence of morphological relations among words. For example, the words “dies” and “died” are stemed to “di” by the Porter stemming algorithm. Although “di” is not a valid English word, it is enough for the IR system to stem “dies” and “died” to the same word [53]. However, studies have shown that the use of a dictionary to assist a stemmer can improve the accuracy. This fact was also verified in the work by Ahmad et al. [7] which showed that high error rates and incomprehensible stems are obtained when no dictionary is used in stemming Malay words. This claim is also believed to be true for the Indonesian language since the Indonesian and Malay languages are closely related.

Examples of the first category are an Indonesian word stemmer devised by a student of University of Indonesia [108] and another developed by Kent Ridge Digital

Lab, Singapore. There is also a web-based word stemming so-called ‘Kamus Elektronik Bahasa Indonesia’ (KEBI) – an Indonesian dictionary which incorporates a stemmer when looking up a word – built by the BPPT and can be accessed online at http://nlp.aia.bppt.go.id/kebi/. This dictionary contains 500,000 word entries and more than 2 million derivational and inflected words which enable users to add new words and definition [97]. It has also been observed that the set of affixes in Indonesian language is not fixed. That is to say, its usage evolves and grows, particularly because of the growth of slangs or bahasa gaul which is popular amongst younger generations. This sublanguage is frequently used on the Internet in chat rooms, newsgroups, websites and emails. Thus, Indradjaja and Bressan [53] developed an approach that is adaptive to the changes by automatically induce the stemming rules from a corpus with the dictionary obtained from http://www.seasite.niu.edu/Indonesian/. Other Indonesian stemmer developments were conducted by using CS algorithm and DICT-UI dictionary with 29,337 Indonesian words [4], [16]. This stemmer implements confix-stripping approach and was claimed to be the best-performing stemmer for the Indonesian language and can improve the effectiveness of Indonesian text IR [4].

An example of the second category was developed by Vinsensius [119] which works roughly like the Porter stemmer for English. This stemmer is utilized by an IR system for the Indonesian language.

The third activity of NLP for Indonesian language is IR. An example of IR system is the Indonesian English cross language IR which takes a query written in Indonesian language and retrieves document written in English [5], [3]. This system is useful to people who are not fluent in English to be able to find English text documents relevant to their information needs.

The last Indonesian NLP activity is MT and will be discussed separately in Section 2.5. MT is defined as the use of computers to automate some or all of the process of translating from one language to another [55]. We pay attention to this research area since generally Indonesia is a country in which English is not the first language. Hence, there is a need to translate this information into Indonesian language. However, only few works have been done to develop MTs on Indonesian language. Most of them are still in ongoing research focusing more on the development of MT resources such as bilingual corpora, which will be used to develop SMT.

2.4 Machine Translation Approaches

Since grammar formalism was introduced in ancient times, from 500 B.C. by Panini for Sanskrit grammar, to 7th – 8th centuries A.D. during Umayyad and Abbasid times for classical Arabic grammar, the ability to translate from one language to another has been made very possible. Currently, many of automatic systems which are doing this translation have reached very good performance, although for open-domain translation, the output quality cannot yet be compared with that of bilingual human. In the last decade, many machine translation systems have reached the stage which was comparable to human translation. These high achievements are mostly carried out on English or other European language. These languages are considered to have well-defined grammar formalism, parser and corpus. Previously, defining rules for sentence-to-sentence mapping between two languages are mostly done by researchers. However, since they found the difficulty to write rules that capture all the complexities of actual utterances, statistical approaches dominate machine translation research in recent years.

Current computational models of machine translation systems deal with a number of non-literary translation tasks, including: (1) tasks where a rough translation is sufficient, (2) tasks which still require a human post-editor, and (3) tasks for small sublanguage domains in which fully automatic high quality translation is still achievable. Information acquisition on the web is the kind of task for which a rough translation is adequate. Suppose we were at the market in Jakarta and smell good aroma as we walked in front of a “soto” restaurant. If we want to know how to cook “soto” – a famous food in Indonesia – then find the “soto” recipe in the web and use online English-Indonesian MT system and immediately we get very rough English sentence as follow.

> “Fry onion and garlic until chocolate, fill in beef stock until boilt and fill in enough carrot, bean sprouts, cabage, celery, and small cuts of meat.”

Although the translation seems to be funy to native English speakers, it is sufficient for them to get an idea of something to try in the kitchen.

An MT system can also be used to speed-up the human translation process, by producing a draft translation that is fixed up by a human translator in a post-editing phase. That is to say, systems are doing computer-aided human translation (CAHT or CAT) rather than fully automatic machine translation. Such MT task is effective especially for huge volume jobs and those that require quick cycle, such as software manual translations to reach local markets.

Weather forecasting is an example of a sublanguage domain that can be modeled by using raw MT output even without post-editing. Weather forecasts consist of phrases like "Sunny with a chance of cloudy on Saturday", or "Outlook for Sunday: Rainy". This domain merely uses limited vocabularies and phrases. Ambiguous words are rare and can be easily disambiguated based on local context by using word classes and semantic features such as WEEKDAY, PLACE, or TIME POINT. Other examples of the sublanguage domain include appointment scheduling, air travel queries, hotel or restaurant recommendations, and equipment maintenance manuals.

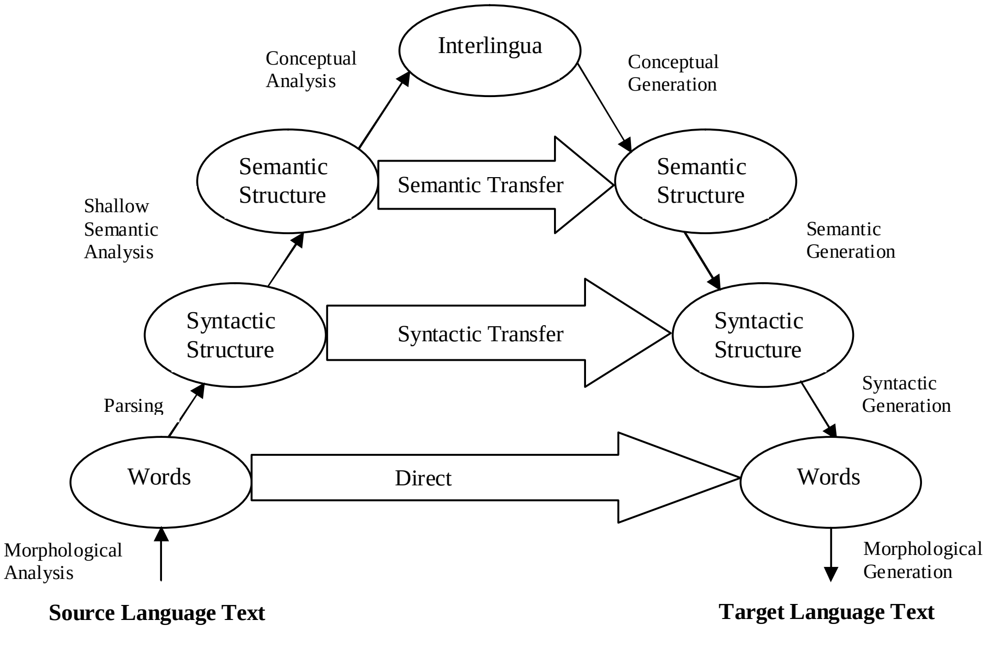


Figure 2.12: The Vauquois triangle

The next sub sections introduce three classical MT approaches (direct, transfer, and interlingua) and the modern statistical MT approach. Some real systems tend to involve combinations of elements from these three approches e.g. combining direct and transfer approaches. The Vauquois triangle shown in Figure 2.12 is a common way to visualize these three approaches. The increasing depth of analysis and generation process can be seen as we move from the direct approach through the transfer approach, to the interlingua approach. Oppositely, if we move up the triangle, it shows the decreasing amount of transfer knowledge needed from high amounts of transfer at the direct level (almost all knowledge is transfer knowledge for each word) through transfer, to small amount of transfer knowledge at interlingua. Each of the four MT approaches is explained in the following sub sections.

2.4.1 Direct Approach

In direct translation, word-by-word mapping analysis is done through the comparison between source and target language texts. The mapping process is started from the first word, proceeds to the next word until the end of the source text and always be evaluated when a problem arise. Structures of the source and target languages are not studied. However, this approach focuses on individual words too much. To deal with real examples thus phrasal and structural knowledge still need to be added. Each source word is directly mapped into target word, involving shallow morphological analysis. Direct translation uses a large bilingual dictionary. After the words are translated, simple reordering rules can be applied, for example for moving adjectives after nouns when translating from English to Indonesian. While the pure direct approach is no longer used, this transformational intuition is still used in other more modern MT approach, for example the direct module by Novento [84] which is combined with the rule-based/transfer approach MT system used for this research.

2.4.2 Transfer Approach

In transfer model, both source and target language sentence morphological (e.g. the ‘-s’ of ‘cakes’ indicates plural) and syntactical (e.g. ‘article+noun’ is grammatical but ‘article+verb’ is not) structures are extracted, with the goal of producing transfer rules for grammatical translation. The strength of the syntac-based model is that the translation results tend to be syntactically well-formed. The structural and feature information in this approach can also be reusable for other syntactic purpose. The disadvantage of the transfer model is on the phrasal coverage and the decoding efficiency. The biggest overhead is the machine readable dictionary, which contains grammatical properties of words. Another drawback is the need for different set of transfer rules for each pair of the languages. It will require $n(n-1)$ transfer rules if each transfer module is not reversible or $n(n-1)/2$ transfer rules if each transfer module is reversible, where n is the number of languages involved [115]. These transfer rules development usually requires many years time period and a lot of human resources. This work especially becomes the job of computational linguists to study language(s) concerned and to write computational grammars that correctly analyze and generate grammatical structures. Nevertheless, there are certain advantages which motivate the development of MT using this approach:

1) many MT systems are bilingual,

2) portion of transfer rules module can be shared between closely related languages, such as transfer rules module sharing between English-Indonesian module and English-Malay module.

Furthermore, it was claimed by Somers [111] that the most successful commercial MT systems were all transfer, though many reflect a long period of development and investment.

2.4.3 Interlingua Approach

One problem with the transfer approach is that it requires a distinct set of transfer rules for each pair of languages. This is clearly less advantageous for translation systems employed in many-to-many language translation environments. Thus, the interlingua approach is introduced to treat translation as a process of extracting the meaning of the *SL* and then expressing that meaning in the *TL*. It is done by analyzing the *SL* text into some abstract meaning representation, called an interlingua. From this interlingua representation then the target language is generated. This MT approach requires more analysis work than the transfer approach, which only requires syntactic rules. It is relies on the syntactic and semantic rules used by a standard interpreter and generator for each language. As a result, the amount of knowledge needed is proportional to the number of languages the system handles. In fact, only $2n$ knowledge converters will have to be written in interlingua approach [31]. Therefore, this approach is suitable for translation systems employed in many-to-many language environments and is only used in sublanguage domains such as in the air travel, hotel reservation, and restaurant recommendation domain. Examples of interlingua systems are the Multilingual Machine Translation System (MMTS) project as part of a multi-national research project between China, Indonesia, Malaysia, Thailand, led by Japan [79]. Another implementation of this approach is the international project on Universal Networking Language (UNL) which involves 12 languages like Japanese, English, Hindi, etc [31]. The project studies several language divergences and the implication to machine translation between these languages using the Universal Networking Language (UNL). UNL has been introduced by the United Nations University in Tokyo to facilitate the transfer and exchange of information over the Internet.

2.4.4 Statistical-based Approach

Statistical-based MT – which is a modern method – focuses more on analyzing the sequence of words during the translation process. For example, how are an input sentence consisting sequential words translated into a *TS* in another language with different sequence of words. Afterward, what is the probability of the *SS* translated into the *TS*? That is to say, this method derives mostly the nonstructural information from bilingual corpora. Thus, this becomes one of a suitable model for building an MT of two structurally different languages. Such method can achieve fast system development without the need of linguists with tedious works of checking and revising the grammatical translation of the MT system. The weakness of this approach is the lack of syntactic knowledge [23]. Somers [111] reported that experiments with English-French produced about 60% usable translations, while the rest consisted of syntactical errors such as wrong genders. Therefore many SMT researchers become aware that incorporating a small amount of linguistic knowledge will increase the MT performance enormously. The pre-requisite is large amounts of bilingual text. MT systems using this approach were reported to be the most successful systems in the NIST 2006 MT evaluation [81] such as the statistical string-to-tree model [44]. More recently, Koehn et al. [64] and DeNeefe et al. [32] explained that many statistical MT systems have improved their quality with the use of phrase-based translation, such as statistical phrase-based model [88], the syntax-based translation system [127] which used phrase translation, and the joint-probability model for phrase translation [71]. Most recently, Chiang [26] presented a hierarchical phrase-based MT system that performed significantly better than the statistical phrase-based system [88].

2.4.5 Hybrid Approach

Many modern and successful MT systems use a combination or hybrid between two approaches such as direct with transfer and transfer with statistical approach. The approach so-called hybrid approach is performed in order to eliminate the disadvantages of one approach by taking the advantages of another approach. For example, Jurafsky and Martin [55] highlight that although the transfer approach offers the ability to deal with more complex *SL* phenomena than the direct approach; it still requires many constraints which combine rich lexical knowledge of both languages with syntactic and semantic features. These constraints are usually already implemented as modules of direct-based MT system. For this reason, commercial MT systems tend to be combinations of the direct and transfer approaches, using not only rich bilingual dictionaries but also using taggers and parsers.

Hutchins and Somers [52] and Senellart et al. [105] describe an example of a hybrid machine called the Systran system, which has three components. The first component is a shallow analysis phase with the following tasks:

- morphological analysis and part of speech tagging,
- chunking of NPs, PPs, and larger phrases,
- shallow dependency parsing (subjects, passives, head-modifiers).

The second component is a transfer phase which includes:

- translation of idioms,
- word sense disambiguation,
- assigning prepositions based on governing verbs.

Finally, the third component is the synthesis phase, where the system:

- applies a rich bilingual dictionary to do lexical translation,
- deals with word reorderings,
- performs morphological generation.

Similar to the direct system, the translation process of Systran relies much on the bilingual dictionary, which has lexical, syntactic, semantic knowledge, and does word reordering in a post-processing step. However, similar to a transfer system, many of the translation steps are informed by syntactic and shallow semantic processing of the *SL*. These efforts inspire and provide the main reason of why the developed MT system uses a hybrid approach between direct and transfer approaches. Another reason is that we do not have available corpus which takes linguists-years resources to be used for developing another hybrid approach so-called EBMT (explained in the following lines) or the pure SMT.

Somers [111] described an approach of MT methods that is suitable for less-resourced language such as Welsh. The methods used involved RBMT, SMT and a famous hybrid approach called EBMT, which can be seen as a hybrid of RBMT and SMT. Similar to SMT, EBMT depends on a corpus of already existing translations, which is reusesed as the basis for a new translation. Such process is similar to the translator’s aid known as a Translation Memory (TM). Both EBMT and TM involve the matching of the input against a database of real translation examples, and identifying the closest matches. They differ in that in TM it is up to the translator to decide what to do with the proposed matches, whereas in EBMT the automatic process continues by identifying corresponding translation phrases and then recombining all phrases to give the target text. Hence, the process is broken down into three phases: “matching” (which EBMT and TM have in common), “alignment”, and “recombination”. Each of these phases tends to be similar to RBMT in implementation, though statistical probabilities also take a part. Like SMT, one attraction of this approach to MT is the extent to which the computer can learn on how to do translations. Another example of EBMT system is the English-Malay MT system reported by Al-Adhaileh and Tang [8]. In this system, Bilingual Knowledge Bank containing bilingual sentences are annotated with their dependency structures [99] using SSTC.

2.5 English-Indonesian MT System Developments

Some works have been done to develop MT on Indonesian language. A notable MT activity for Indonesian language is the Multilingual Machine Translation System (MMTS) project. This project was conducted by the Agency for Assessment and Application of Technology (BPPT) as part of a multi-national research project between China, Indonesia, Malaysia, Thailand, and led by Japan, as explained by Nazief [79]. This MMTS includes Bahasa Indonesia Analyzer System (BIAS), an analysis component for Indonesian language part [130]. BIAS uses Interlingua approach which takes Indonesian text as input and produces abstract meaning representation, called an Interlingua. From this Interlingua representation, the target language is generated. This MT system is only relying on the syntactic and semantic rules used by a standard interpreter and generator for each language involved in that multilingual system. As a result, the amount of knowledge needed is proportional to the number of languages the system handles. Therefore, this kind of system is only used in sublanguage domains [55] and is appropriate for many-to-many languages translation tasks, as suggested by its name ‘interlingua’.

In recent years, there are a few available English-Indonesian MT software, such as Rekso Translator [98], Translator XP [93], and KatakuTM [60]. No details are available on the algorithm applied in their translation engines. Another available English-Indonesian MT system is the Google Translate application [46]. This application provides translation from multiple languages to Indonesian as well as the tranlation from Indonesian language to the other languages (http://en.wikipedia.org/wiki/Google_Translate#cite_note-2). In the Google Translate application, a statistical approach based on phrase translation [87] is implemented.

Another SMT system is developed by BPPT and ANTARA – referred as BPPT-ANTARA MT system – based on Pharaoh using 500K sentences pair (current BLEU score 0.72) [97].

Indonesian language has the same root and hence shares many aspects with the Malay language. Both languages have close similarities in terms of phonetic, morphology, semantic and syntactic. Thus, MT studies on Malay languages are also referred. An extensive work in the field of MT was conducted by a research group in the University of Science Malaysia (USM) which uses the EBMT approach (referred as USM MT system). In the research work, a technique to construct the SSTC for the Malay sentences by means of a synchronous parsing technique was introduced. This technique automatically generates the SSTC for the English sentence through the use of existing English parser [10]. The technique used synchronous parsing technique to parse the Malay sentence based on the English sentence parse tree together with the alignment result obtained from the alignment algorithms. The advantage of this technique is that it can solve non-projective cases. The limitations include extra work required to annotate all dependent/constituent levels in the English-Malay corpora and the need to formalize both English and Malay grammars.

The approaches used by the last three MT systems previously discussed, namely Google Translate and BPPT-ANTARA systems which use SMT approaches, and USM MT system which uses EBMT approach, are categorized as data-driven approach. As already explained in Section 1.1, data-driven approaches highly depend on the size of its training bilingual corpora [72]. Unfortunately, bilingual corpora involving some less-resourced languages (such as languages in South East Asia including Indonesia) are very limited or even none. Although Indonesian corpus developments have been started by some research groups [79], [97]; the availability is still unknown. The other way to get the bilingual corpora is to create it, but this needs high cost of human resources and time consuming [85]. Hence, it will be wiser if a quick start is taken on an MT project using other available language technology resources e.g. free parser of the *SL* (English) side such as Link Parser [47] and any available bilingual English-Indonesian dictionary.

The problem of the nonexistence of bilingual corpora motivates us to use a direct, or a transfer, or a hybrid between direct and transfer approaches. The use of direct approach for English-Indonesian MT system was done by a research group from Gadjah Mada University, Indonesia [84]. However, pure direct approach is no longer used recently, but its constraint modules and rich bilingual dictionary can be useful and adapted into the RBMT system, which is the main work reported in this thesis. The combination between the direct and transfer approach – referred as hybrid transfer approach – is chosen over the pure transfer approach since the *TL* (Indonesian language) is yet to have available grammar formalism and parser. The hybrid transfer MT system so-called ADJ-based system uses LG formalism. This formalism is chosen because it is inline to linguistics intuition better than DG and CG [104]. Different with that of DG and CG, LG system has no explicit notion of constituents or dependents in a sentence. This will reduce the transfer rules complexity, but, at the same time also cause the limitation of the system with its inability to handle non-projective cases (the problems arise in the dependent or constituent levels). Another limitation is that the translation is only in one direction, which is from English to the *TL* (Indonesian language). ADJ-based system also uses Link Parser – a free English parser built in LG formalism – that has an algorithm complexity of $O(n^3)$, the same as that of CG parser [104].

CHAPTER 3

METHODOLOGY

In this chapter, the idea of the use of word disjuncts in a sentence for an MT system is explained. The discussions start with an MT schema based on LG formalism which is given in Section 3.1. The section describes each component of the developed MT model. The ADJ Component, the most important component in the system, gives readers an idea on how the word disjuncts are annotated. The annotation yields ADJ set, which is utilized in the Transfer Rules Component of the MT model. Section 3.2 highlights the development of ADJ-based MT system with a case study on English-Indonesian MT system. How the ADJ-based MT model solves translations for blocks of texts is discussed here. The transfer rules for the English-Indonesian ADJ-based MT system are separately given in detail in Section 3.3. Finally, hierarchical phrase-based transfer rule which are used in the latest version of the developed system is explained in Section 3.4.

3.1 MT Schema Based on Link Grammar Formalism

One of the main contributions in this research is an MT model (see Figure 3.1), which is based on the LG formalism. It consists of four components:

1) Pruning Algorithm Component,
2) Parsing Algorithm Component,
3) ADJ Component,
4) Transfer Rules Component.

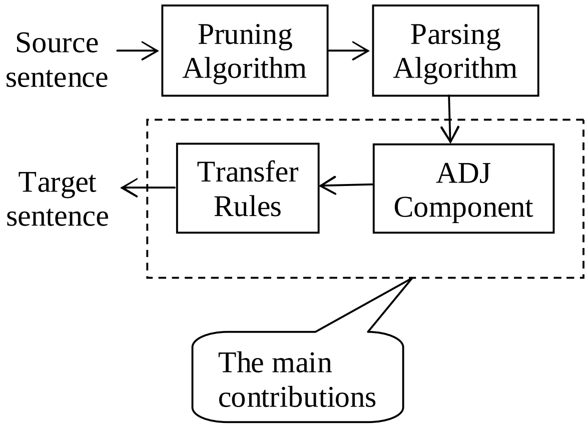


Figure 3.1: MT model diagram based on Link Grammar

This MT model does not employ structure-to-structure mapping mostly used in syntax or transfer model. Instead, a hybrid of transfer and direct approaches is used for sentence-to-sentence mapping. This hybrid transfer approach applies the pruning and parsing algorithms to a *SS* that satisfies LG formalism. The pruning and parsing algorithms can be found in Sleator and Temperley [109] and Grinberg et al. [47]. The ADJ and Transfer Rules Components are the main contributions or novelties of this work. In the ADJ Component, disjuncts annotation is done to process the output of the pruning and parsing algorithms modules i.e. the pruned and the parsed *SS*. This component yields ADJ set, which is an important property to a word in a sentence. This property is useful for sentence-to-sentence mapping of an MT model of two structurally different languages.

The input sentence in Figure 3.1 must not only satisfy the LG formalism. It also must be in the dictionary of ADJ. This dictionary was built during the disjunct annotations process. It contains all pairs of source and target words. Each of the pair words is attached with the appropriate ADJ. The transfer rules utilize this ADJ to give the translation of each source word and to arrange all the target words in a correct *TS* structure. The following sub sections explained each component of the developed MT model.

3.1.1 Pruning Algorithm and Parsing Algorithm Components

The Pruning Algorithm Component was implemented in ANSI C as reported by Sleator and Temperley [109]. This code is embedded into our system to prune an English sentence. The Parsing Algorithm module consists of ANSI C codes developed by Grinberg et al. [47]. The algorithm is also modified to obtain the first linkage of the pruned sentence. A linkage contains a sequence of words, which form a sentence and all links that connect each word satisfies the linking requirement (see Sub Section 2.3.3 for detail).

3.1.2 Annotated Disjunct Component

The original parsing algorithm in LG generates a list of linkages at their own cost [109]. The ADJ Algorithm module in this work is designed to consider only the first linkage in the generated list as it holds the lowest cost. As a result, only a single set of word disjuncts is obtained. These disjuncts are the components used for producing the ADJ. In this research, all the disjuncts of a word are rewritten as $d_i(k)$. $k \geq 1$ and $k \leq$ the total number of disjuncts for the word. $d_i(k)$ can be explained as the k^{th} disjunct of the i^{th} word of an *SS* in LG.

For clear explanation, let us rewrite all the word disjuncts in the first linkage of “John picks the heavy box up” (see Figure 2.9) in terms of $d_i(k)$. Note that the words disjuncts are already derived as explained in Sub Section 2.1.3. Thus, all the disjuncts of the word “John” can be rewritten into four disjuncts in the forms of $d_i(k)$ as:

$d_1(1) = ((O)(\ ))$,

$d_1(2) = ((\ )(S))$.

The $d_i(k)$ forms of the disjuncts of the word “picks” are:

$d_2(1) = ((S)(K, O))$,

$d_2(2) = ((S)(O))$.

While the $d_i(k)$ form of the disjunct of the word “the” is:

$d_3(1) = ((\ )(D))$.

The disjunct of the word “heavy” has the $d_i(k)$ form of:

$d_4(1) = ((\ )(A))$.

Whereas the disjuncts of the word “box” are rewritten in $d_\mathrm{i}(k)$ forms as follows:

$d_5(1) = ((A, D, O)(\ )),$

$d_5(2) = ((A, D)(S)),$

$d_5(3) = ((D, O)(\ )),$

$d_5(4) = ((D)(S)).$

The last word (“up”) is then rewritten as:

$d_6(1) = ((K)(\ )).$

The disjunct of the i^{th} word in a sentence which satisfies a linkage is subsequently annotated along with the target word e.g. the Malay word. For example, based on Figure 2.9 and the translation as in Figure 3.2, the annotations for all the words in “John picks the heavy box up” are:

- the disjunct of the word “John” – which is equal to $d_1(2) = ((\ )(S))$ – is annotated with δ_1 if the target word is “John”,
- the disjunct of the word “picks”, $d_2(1) = ((S)(K, O))$, is annotated with δ_2 if the target word is “kutip”,
- the disjunct of the word “the”, $d_3(1) = ((\ )(D))$, is annotated with δ_3 if the target word is “itu”,
- the disjunct of the word “heavy”, $d_4(1) = ((\ )(A))$, is annotated with δ_4 if the target word is “berat”,
- the disjunct of the word “box”, $d_5(1) = ((A, D, O)(\ ))$, is annotated with δ_5 if the target word is “kotak”,
- the disjunct of the word “up”, $d_6(1) = ((K)(\ ))$, is annotated with δ_6 if the target word is “ ” or blank character.

A set with three elements namely the source words, target words, and the appropriate disjunct annotations is called as ADJ set. Hence, the ADJ set obtained from the first linkage of “John picks the heavy box up” and its translation has the ADJ set of {($John, John, \delta_1$), ($picks, kutip, \delta_2$), (the, itu, δ_3), ($heavy, berat, \delta_4$), ($box, kotak, \delta_5$), (up, “ ”,δ_6)}. This set of ADJ will map the *SS* into the *TS* as illustrated in Figure 3.2.

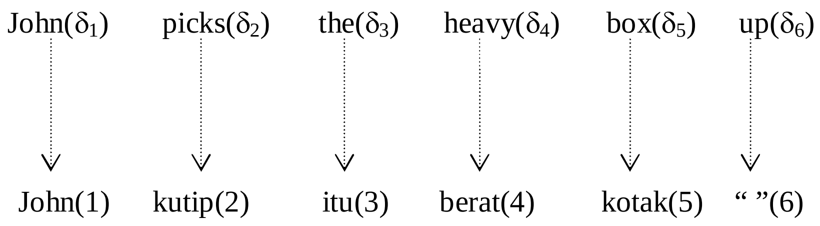


Figure 3.2: English-Malay word-by-word mapping using ADJ Component

3.1.3 Transfer Rules Component

It can be seen from Figure 3.2 that the words in the *TS* are not in a correct order. Hence, Transfer Rules Component is needed for repositioning words in a *SS* to obtain a grammatical *TS*. In the MT system development process, more than one transfer rule are built to handle one-to-one, one-to-many, and many-to-many translation cases. The mathematical formula for the transfer rules is given in Equation (3.1).

$$SS(W_1(d_1), W_2(d_2), \ldots, (W_i(d_i), W_{i+j}(d_{i+j})), \ldots, W_n(d_n))$$

$$\rightarrow TS(W_1'(new1), W_2'(new2), \ldots,(W_i'(newi), W_{i+k}'(newi+k)), \ldots, W_n'(newn)) \qquad (3.1)$$

where $SS(W_1(d_1), W_2(d_2), \ldots, (W_i(d_i), W_{i+j}(d_{i+j})), \ldots, W_n(d_n))$ is a source sentence in a language L comprising a list of ordered words W_1, W_2, …, W_n. *SS* consists of one phrase (W_i, W_{i+j}) which can be located in any place in *SS*. *j* is an integer from 1 to (n-i). $TS(W_1'(new1), W_2'(new2), \ldots, (W_i'(newi), W_{i+k}'(newi+k)), \ldots, W_n'(newn))$ is a target sentence in language *L’* comprising a list of words W_1', W_2', …, W_n', and one phrase (W_i', W_{i+k}'). *k* is an integer from 1 to (*newn*-*newi*). $W_i(d_i)$ is the source word having d_i disjunct, which is located in the n^{th} position in *SS*. $W_i'(newi)$ means the target word of the mapping from the source word W_i. W_i' takes the $newi^{th}$ position in *TS*.

The number of words in *SS* and *TS* in Equation (3.1) may not be equal. It occurs if $i+k = 0$, so that $W_{i+k}' = W_0'$. W_0 and W_0' is assigned as null value of *SS* and *TS*. As a result, a phrase of words (W_i, W_{i+j}) is translated into one word W_i'. This phenomenon is handled by Equation (3.2).

$$\begin{aligned}
&W_1(d_1) \rightarrow W_1'(new1) \text{ } if \text{ } d_1 = \delta_1,\\
&W_2(d_2) \rightarrow W_2'(new2) \text{ } if \text{ } d_2 = \delta_2,\\
&\quad\vdots\\
&(W_i(d_i), W_{i+j}(d_{i+j})) \rightarrow W_j'(newi) \text{ } if \text{ } d_i = \delta_i \text{ } \& \text{ } d_{i+j} = \delta_{i+j},\\
&\quad\vdots\\
&W_n(d_n) \rightarrow W_n'(newn) \text{ } if \text{ } d_n = \delta_n.
\end{aligned} \tag{3.2}$$

Illustration in Figure 3.3 shows how the annotated disjunct can be used in the translation from the English sentence John picks the heavy box up into Malay sentence “John kutip kotak berat itu”.

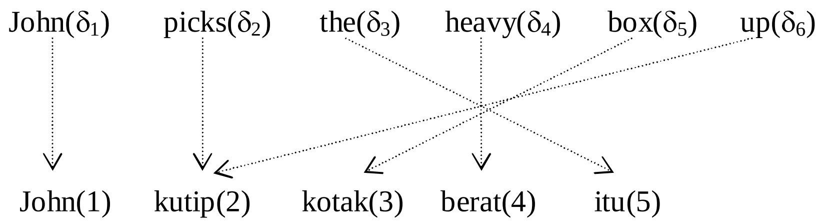


Figure 3.3: English to Malay translation using Transfer Rules Component

Equation (3.2) is used for the translation illustrated in Figure 3.3.

$John(d_1) \rightarrow John(1) \text{ } if \text{ } d_1 = \delta_1,$

$(picks(d_2), up(d_6)) \rightarrow kutip(2) \text{ } if \text{ } d_2 = \delta_2 \text{ } \& \text{ } d_6 = \delta_6,$

$the(d_3) \rightarrow itu(5)\ if\ d_3 = \delta_3$,

$heavy(d_4) \rightarrow berat(4)\ if\ d_4 = \delta_4$,

$box(d_5) \rightarrow kotak(3)\ if\ d_5 = \delta_5$.

Hence, there is a many-to-one translation from the English phrase “picks up” into the Malay word “kutip”. The order of the translated words can also be corrected, for example the English word box which is the 5th word of the *SS* is translated into Malay word “kotak” which is repositioned into the 3rd position of the *TS*.

A one-to-many translation case is the translation from the English sentence “John picks the heavy box up” into Japanese sentence “John wa sono omoi hako o toru”. In Japanese, “wa” is a particle that follows a subject, and the word “o” is another particle that follows an object. Thus, there exist one-to-many translations from the English word “John” into the Japanese words “John wa” and from the English word “box” into the Japanese words “hako o”. Figure 3.4 shows how the ADJ can handle the translation.

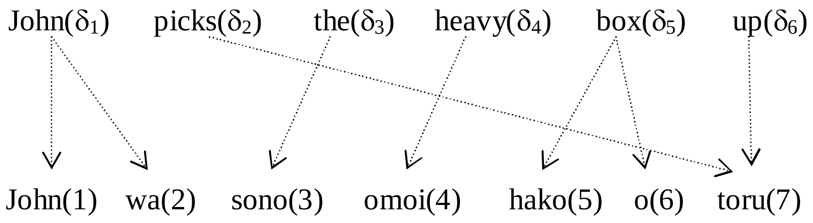


Figure 3.4: English to Japanese translation using Transfer Rules Component

It is apparently seen from this figure that many-to-many translation also occurs in the translation from the English phrase “picks up” into the Japanese word “toru”. The computation using Equation (3.2) for Figure 3.4 is:

$John(d_1) \rightarrow (John(1), wa(2))\ if\ d_1 = \delta_1$,

$(picks(d_2), up(d_6)) \rightarrow toru(7)\ if\ \mathrm{d}_2 = \delta_2\ \&\ d_6 = \delta_6$,

$the(d_3) \rightarrow sono(3)\ if\ d_3 = \delta_3$,

$heavy(d_4) \rightarrow omoi(4)\ if\ d_4 = \delta_4$,

$box(d_5) \rightarrow (hako(5), o(6))\ if\ d_5 = \delta_5$.

It is also obvious from the illustration in Figure 3.3 and Figure 3.4 that the transfer rules do not have explicit notions of the heads, constituents, and dependents in a sentence. Hence, this simplicity will ease the transfer rules development, as shown in Equation (3.2). The drawback is that this model cannot handle non-standard cases. Meanwhile, most MT models in DG and CG formalisms consider the constituent and dependent structures. Models uses DG and CG formalism can resolve non-standard cases i.e. non-projective cases. The disadvantage is that the dependent and constituent annotations need extra work.

3.2 ADJ-based MT System: Case Study on English-Indonesian MT System

Based on the MT model in Figure 3.1 in which Annotated Disjunct and Transfer Rules Components are proposed as the main contributions or novelties, an English-Indonesian MT system is developed to prove that the ADJ-based method proposed is valuable to the MT research community. This section clarifies the fundamentals of the ADJ-based method used in the English-Indonesian MT. The discussions start with the architecture overview, followed by a description of how disjuncts are annotated for single sentence, and end with the ADJ-based method for blocks of texts. Since the Transfer Rules Component for the English-Indonesian MT system needs many pages to highlight, it will be given separately in Section 3.3.

3.2.1 Architecture Overview

The architecture of the developed English-Indonesian MT system is shown in Figure 3.5. It consists of four modules: Pruning Algorithm module, Parsing Algorithm module, ADJ Algorithm module, and Transfer Rules Algorithm module. Since the MT system is a sentence-based system then the expected form of input for the system is also a sentence (i.e. an English sentence). If the input is a text or paragraph, then the system utilizes one of its functions to decompose the text into sentences. A sentence in a text is identified by either the full stop or its question/imperative mark. Each sentence is fetched sequentially into all modules. The pruning module deletes all disjuncts in the sentence containing a connector that does not match any other connector of any word. This method aims to reduce the running time of the next module (i.e. the parsing algorithm module).

The parsing algorithm module is adapted from [47]. The original parsing, which is called Link Parser, can result in more than a single linkage solution. In the ADJ-based MT system, the Link Parser was customized to obtain the first linkage of the pruned sentence, which holds the lowest cost among the generated linkage list. The cost is defined as the number of null links within a linkage. A null link is an unlabeled link that connects adjacent words. The parsing module is an algorithm that accepts a sentence with all the pruned word disjuncts as an input and parses this input to build up a linkage.

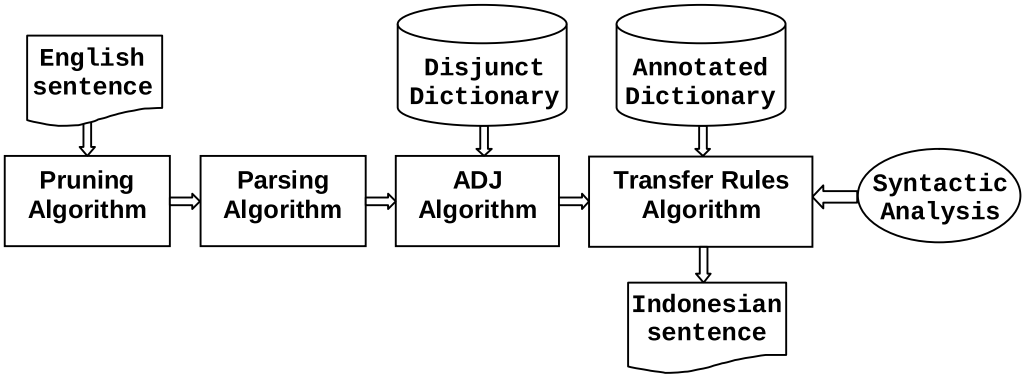


Figure 3.5: ADJ-based English-Indonesian MT system architecture

How the Link Parser does the parsing is somewhat the opposite with the task of the pruning module. This robust parser finds the match between all connectors of the words in a sentence. This parsing module has a dictionary of approximately 60,000 words and is able to recognize a wide range of English syntactic phenomena, namely many types of nouns, complex and irregular verbs, questions, imperatives, past or present participles, commas, a variety of adjective types, prepositions, adverbs, relative clauses, possessives, conjunctions, and others [33]. The parser was tested on a Switchboard corpus of conversational English. This corpus consists of about three million words of text. This corresponds to more than 150 hours of transcribed speech collected from telephone conversations in 70 different topics. However, most of the sentences in the corpus are not grammatical. Nonetheless, the robust Link Parser can handle a large portion of the corpus. If a complete linkage cannot be found, the parser tries to form a “partial linkage” by ignoring one word or more in the sentence. The parser has an internal timer. If the parsing process time exceeds the limit before a complete or partial linkage has been found, the parser will output whatever fragmented linkage it has generated [47]. A version of the parser is available at http://www.link.cs.cmu.edu/link/.

The ADJ module, which consists of the ADJ Algorithm, takes the first linkage as an input to construct an ADJ. This ADJ set is obtained by utilizing the knowledge of English grammatical relationship between words which is stored in the disjunct dictionary. ADJ is a set of source word (English), target word (Indonesian), and the associated disjunct. Each associated disjunct can relate the English word and its translation in Indonesian. It must be noted that an English word can be translated into more than one Indonesian word. The list of possible English-Indonesian pairs as can be generated from one English word is stored in the annotated dictionary.

These transfer rules accept the ADJ set as an input and use the translation knowledge in the annotated dictionary to decide the best match among the English-Indonesian pairs. The ADJ set consists of a sequence of source words and a sequence of target words. Both sequences describe their word(s)-to-word(s) mapping. The source words come from a grammatical *SS*. Thus, its sequence is grammatical. The sequence of the target words is obtained by the direct mapping of the source words. However, the target words sequence follows the source words sequence and this will, most of the time, consists of incorrect *TS* grammar. Hence, we developed transfer rules with the task to reposition the target words to get a grammatical *TS*.

The following sub sections explain in detail how the ADJ-based method can be used for mapping a single sentence and blocks of texts.

3.2.2 ADJ-based Method for Single Sentence

The simple way to understand the ADJ-based MT system is to start with an example of a linkage. Figure 3.6 shows a linkage for a sentence “She saw the saw”. The linkage contains four links, *Wd* link (or the left wall/border link), *Ss* link which connects a singular subject noun to a finite verb, *Os* link connecting a transitive verb to its singular object, *Ds* link connecting a determiner to a singular noun, and *Xp* link which connects a punctuation symbol either to another punctuation or to words (see Appendix A for the list of links and their description). Each link is made up of two connectors. For example, *Ss* link that connects the singular noun “She” and the verb “saw” is made up of *Ss* left connector for “She” and *Ss* right connector for “saw” (see Figure 3.6). This linkage concept explains the connectivity of each word in a sentence. Thus, to explain the word modularity in terms of sentence’s independence, a disjunct terminology is introduced by Sleator and Temperley [109]. The disjunct concept is convenient to express words in terms of its connectors, on the left and right, in all possible sentences. The following is just one example of a disjunct which is obtained from one particular sentence. It can be seen that the word “she” of “She saw the saw” in Figure 3.6 has a disjunct of *Wd* left connector and *Ss* right connector.

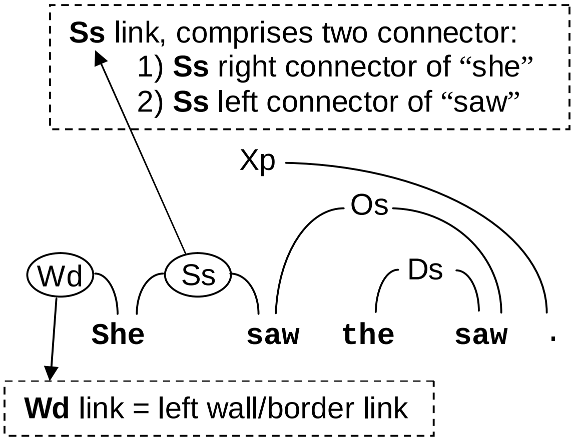

Ss link, comprises two connector:
1) **Ss** right connector of “she”
2) **Ss** left connector of “saw”
Xp
Os
Wd
Ss
Ds
She
saw
the
saw
.
Wd link = left wall/border link

Figure 3.6: A linkage for “She saw the saw.”

All the words and their disjuncts are:

“She” : *Wd* left connector and *Ss* right connector,
“saw” : *Ss* left connector and *Os* right connector,
“the” : empty left connector and *Ds* right connector,
“saw” : *Os* and *Ds* left connectors and empty right connector,
“.” : *Xp* left connector and empty right connector.

The word “saw” has two disjuncts and the other two words (“she” and “the”) has only one disjunct (notice that “She” starting with capital letter and “she” without capital letter are considered as the same word, which result in the same disjuncts). Hence, the words and their disjuncts can be expressed in disjunct formulations (refer to Sub Section 3.1.2) as:

disjunct of “she” = d_1 = ((*Wd*)(*Ss*)),
disjunct of “saw” = d_2 = ((*Ss*)(*Os*)),
disjunct of “the” = d_3 = (( )(*Ds*)),
disjunct of “saw” = d_4 = ((*Os*, *Ds*)( )),
disjunct of “.” = d_5 = ((*Xp*)( )).

It must be noted here that d_i could be explained as the disjunct of the i^th word of *SS* in LG. This associated disjuncts (d_i), along with the corresponding source words (*W*) and target words (*W’*), are used to construct the ADJ set. Thus, the ADJ set is represented in the following structure:

$\{(W_\mathrm{i}, W_\mathrm{i}', d_\mathrm{i}) \mid 1 \leq i \leq n, n = \text{the total number of source words in } SS\}$,

where:

- W_i is the source word located in the i^th position of *SS*,
- W_i’ is the translation of W_i located in the i^th position of *TS*.

For example, the sentence “She saw the saw.” is literally mapped into the ungrammatical Indonesian sentence “Dia melihat itu gergaji.”, giving the following ADJ: {(*She*,*Dia*,((*Wd*)(*Ss*))), (*saw*,*melihat*, ((*Ss*)(*Os*))), (*the*,*itu*,(( )(*Ds*))), (*saw*,*gergaji*,((*Os*,*Ds*)( ))), (**.**,**.**,((*Xp*)( )))}. How do we get the ADJ set? Given a source sentence $SS(W_1, W_2, \ldots, W_n)$ where $W_1, W_2, \ldots, W_n$ are consecutive source words and a target sentence $TS(W_1', W_2', \ldots, W_n')$ where $W_1', W_2', \ldots, W_n'$ are consecutive target words, the ADJ set is obtained using Algorithm 3.1.

Algorithm 3.1 ADJ Algorithm for Single Sentence

```
1.   Obtain the first linkage of a given $SS$.
2.   Obtain all word disjuncts of the first linkage.
3.   For each $W_i$ ($i = 1$ to $n$), do steps 3.1, 3.2, and 3.3.
3.1      Obtain $d_i$.
3.2      Insert $W_i'$.
3.3      Group the $W_i$, $W_i'$, $d_i$ into ($W_i$, $W_i'$, $d_i$) and add
         into the ADJ set.
```

Figure 3.7 illustrates the result of the algorithm when used for mapping the English words in Figure 3.6.

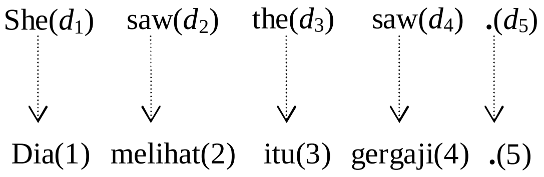


Figure 3.7: English-Indonesian word-by-word mapping using associated disjunct

It is clear from Figure 3.7 that the associated disjunct d_i can distinguish both disjuncts of the homonym “saw” as d_2 and d_4 within a single sentence. What will happen when another sentence such as “What saw is that?”, having the linkage shown in Figure 3.8, is added. Is ADJ Algorithm for Single Sentence still valid for this case? The answer is given in the discussion about ADJ-based Method for Blocks of Texts.

3.2.3 ADJ-based Method for Blocks of Texts

The ADJ-based MT method is based on the Link Parser, which is sentence-based. Given two English sentences to be translated into target sentences, then the ADJ-based system will process the first sentence before the second sentence. Assuming that there is a single bilingual text, which consists of two sentences; the first sentence is the sentence in Figure 3.6 and the second is "What saw is that?". The linkage for the second sentence is given in Figure 3.8.

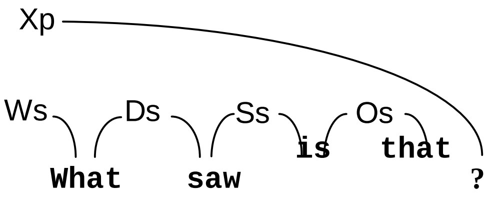


Figure 3.8: A linkage for "What saw is that?"

The linkage is formulated in terms of words and their associated disjuncts which are as follows:

disjunct of "What" = d_1 = ((Ws)(Ds)),
disjunct of "saw" = d_2 = ((Ds)(Ss)),
disjunct of "is" = d_3 = ((Ss)(Os)),
disjunct of "that" = d_4 = ((Os)( )),
disjunct of "?" = d_5 = ((Xp)( )).

The sentence "What saw is that?" is translated into the Indonesian language of "Gergaji apa itu?". In this sentence, "What" is translated into Indonesian word "apa", "is" is mapped into " " (blank), and "that" is translated into "itu". Meanwhile, "saw" is mapped into "gergaji". Now, the word "saw" in the second sentence is associated with d_2. However, d_2 = ((Ss)(Os)) for the first sentence and d_2 = ((Ds)(Ss)) for the second sentence. It can be seen that d_2 from the first and second sentences are not the same. Thus, the annotated disjuncts can be used to differentiate between an associated disjunct with other associated disjuncts of different pairs of source and target words in the whole documents. That is to say, associated disjunct is a disjunct obtained from a single linkage and is in correspondence with a pair of source and target words in the linkage. Whilst annotated disjunct is an associated disjunct that is already annotated by considering all possible linkages in the whole documents. As such, for this particular situation which involves blocks of texts, annotated disjunct is applied.

In this work, the annotated disjunct is represented as $\delta_{\mathrm{j}}(k)$. If k parameter is neglected then δ_{j} itself means all annotated disjuncts for j^{th} source word in the entire evaluated bilingual texts. k is a parameter to represent different annotated disjuncts for a unique source word. In other words, $\delta_{\mathrm{j}}(k)$ is the k^{th} annotated disjunct of the j^{th} source word in a bilingual texts. $k \geq 1$ and $k \leq$ the total number of annotated disjuncts for a unique source word, while $j \geq 1$ and $j \leq$ the total number of unique source words in the annotated dictionary as specified in Figure 3.5.

ADJ set is defined as a set of source word (English), target word (Indonesian), and the annotated disjuncts and is represented in the following structure:

> $\{(W_{\mathrm{i}}, W_{\mathrm{i}}', \delta_{\mathrm{j}}(k)) \mid 1 \leq i \leq n$, n = the total number of source words in a SS, $1 \leq j \leq l$, l = the total number of unique source words in the whole documents, $1 \leq k \leq s$, s = the total number of annotated disjuncts for a unique source word$\}$.

This new definition of ADJ set is now valid for any number of sentences in which their disjuncts need to be annotated.

Let us consider if there is only a single sentence “She saw the saw.” with its linkage as shown in Figure 3.6. All pair of words and their annotated disjuncts of this linkage are given by:

$\delta_1(1) = ((Wd)(Ss))$ if “she” → “dia”,

$\delta_2(1) = ((Ss)(Os))$ if “saw” → “melihat”,

$\delta_2(2) = ((Os, Ds)(\ ))$ if “saw” → “gergaji”,

$\delta_3(1) = ((\ )(Ds))$ if “the” → “itu”,

$\delta_4(1) = ((Xp)(\ ))$ if “**.**” → “**.**”.

Note that the word “saw” is a homonym i.e. has two meanings, which are “melihat” (to see) and “gergaji” (a cutting tool with a zigzag edge) and are distinguished by two annotated disjuncts, which are $\delta_2(1)$ and $\delta_2(2)$ respectively.

Now, if there are two sentences then the disjunct annotation process must be evaluated in both sentence linkages. The pair of words and their annotated disjuncts are now updated into the followings:

$\delta_1(1) = ((Wd)(Ss))$ if “she” → “dia”,

$\delta_2(1) = ((Ss)(Os))$ if “saw” → “melihat”,

$\delta_2(2) = ((Os, Ds)(\ ))$ if “saw” → “gergaji”,

$\delta_2(3) = ((Ds)(Ss))$ if “saw” → “gergaji”,

$\delta_3(1) = ((\ )(Ds))$ if “the” → “itu”,

$\delta_4(1) = ((Xp)(\ ))$ if “**.**” → “**.**”,

$\delta_5(1) = ((Ws)(Ds))$ if “what” → “apa”,

$\delta_6(1) = ((Ss)(Os))$ if “is” → “ ”,

$\delta_7(1) = ((Os)(\ ))$ if “that” → “itu”,

$\delta_8(1) = ((Xp)(\ ))$ if “?” → “?”.

It is obvious that the word “saw” has three annotated disjuncts now: $\delta_2(1)$, $\delta_2(2)$, and $\delta_2(3)$. Two of these annotated disjuncts ($\delta_2(2)$ and $\delta_2(3)$) do not distinguish the meaning since the target words are the same, which is “gergaji”. However, these annotated disjuncts can distinguish the produced associated disjuncts, which are $((Ds, Os)(\ ))$ and $((Ds)(Ss))$.

How do we annotate word disjuncts on a multiple-sentence set? This involves a task of evaluating all possible source word meanings, which are the target words, in the entire bilingual texts. Let say that there are *m* sentences in the entire documents, then the algorithm to obtain the redefined ADJ set can be defined by Algorithm 3.2.

Algorithm 3.2 ADJ Algorithm for *m* Sentences

```
1.      Initialize $j$ and $k$ with 1.
2.      For each $SS_r$ ($r$ = 1 to $m$), do steps 2.1, 2.2, and 2.3.
2.1       Obtain the first linkage of $SS_r$.
2.2       Obtain all word disjuncts of the first linkage.
2.3       For each $W_i$ ($i$ = 1 to $n$) of $SS_r$, do steps 2.3.1, 2.3.2, and 2.3.3.
2.3.1       Insert $W_i$'. Annotate $d_i$ with $\delta_j(k)$.
2.3.2       If different $W_i$ is found, then increment $j$. Annotate $d_i$ with $\delta_j(k)$.
            Do step 2.3.2.1.
2.3.2.1       If different $d_i$ is found, then increment $k$. Annotate $d_i$ with $\delta_j(k)$.
2.3.3       Group the $W_i$, $W_i$', $\delta_j(k)$ into ($W_i$, $W_i$', $\delta_j(k)$) and add into the ADJ
            set.
```

Figure 3.9 shows the result of the ADJ Algorithm for *m* sentences (with *m* = 1 sentence) when the words in a single English sentence "She saw the saw" are mapped to Indonesian words.

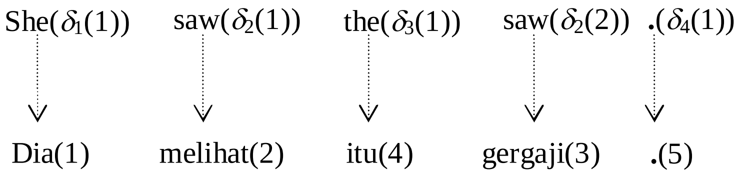


Figure 3.9: English-Indonesian word-by-word mapping using ADJ Algorithm

Meanwhile, Figure 3.10 is the illustration of the mapping from two English sentences to Indonesian sentences using the same algorithm (with *m* = 2 sentences).

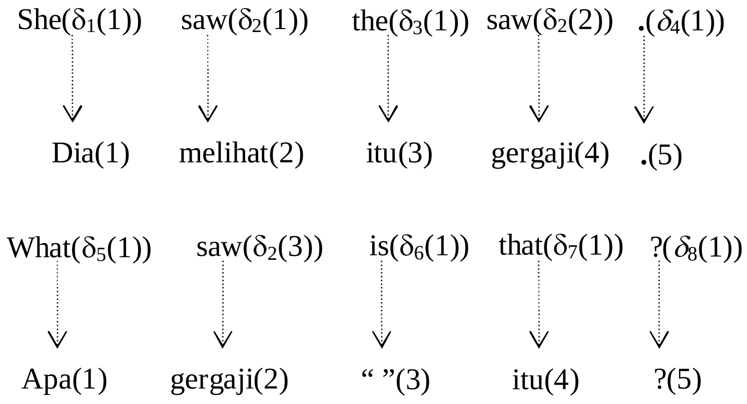


Figure 3.10: Illustration of the mapping of two sentences

Both translations in Figure 3.9 and Figure 3.10 are not correct in terms of fluency since the sequence of the target words is incorrect. The correction of these misplaced words is done by the transfer rules, which is discussed in Section 3.3.

3.3 Transfer Rules of the English-Indonesian ADJ-based MT System

During the ADJ-based MT system development, an evaluation on the transfer rules has always been made. Firstly, sentence-based transfer rules were built following that the Link Parser was also based on sentence. Secondly, as things become complicated for the transfer rules to translate a *SS* with more than 30 words, a phrase-based transfer rules were then developed to solve this problem. Finally, a hierarchical phrase-based transfer rules were made to smoothly and systematically translate phrases in many categories. Sentence-based and phrase-based transfer rules are explained in Sub Sections 3.3.1 and 3.3.2. Whilst the hierarchical phrase-based transfer rules which gives the best precision is discussed in more detail in Sub Section 3.3.3 separately since the latest version of the ADJ-based system used this transfer rules.

3.3.1 Sentence-based Transfer Rules

Based on the ADJ set, a transfer rules algorithm module is developed to arrange all target words in a correct *TS* structure by referring to the syntactic analysis of Indonesian language structure (see Figure 3.5). For example, in English, determiners always precede nouns. This structure is in contrast to the structure of Indonesian language where nouns always precede determiners. ADJ-based method can solve the mapping problem from English to Indonesian sentences. For instance, when a linkage in Figure 3.6 is given as an input for the ADJ module, then the generated Indonesian words is “Dia melihat itu gergaji” as illustrated in Figure 3.9. This sequence consists of grammatical error since the word “itu” (determiner) precedes “gergaji” (noun). But, if the sentence “She saw the saw.” is applied to the Transfer Rules Algorithm, the result produced is “Dia melihat gergaji itu.”. Now, the word “gergaji” (noun) precedes “itu” (determiner). This is due to the result of the word “the” in the 3rd position of the

SS is mapped into “itu” in the 4th position in the *TS* (see Figure 3.11). The transfer rules algorithm for implementing the illustration in Figure 3.11 is given in Algorithm 3.3.

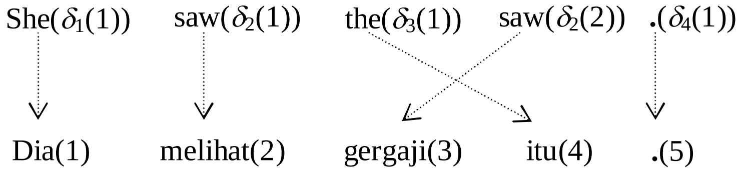


Figure 3.11: English to Indonesian sentence translation using transfer rules algorithm

Algorithm 3.3 Sentence-Based Transfer Rules Algorithm

```
1.    For each $W_i$ ($i$ = 1 to $n$), do steps 1.1.
1.1       If $d_i$ equals a certain $\delta_i$, then $temporary\_word[i] \leftarrow W_i'$.
2.    For each $W_i$ ($i$ = 1 to $n$), do steps 2.1.
2.1       $target\_word_i \leftarrow temporary\_word[x_i]$.
```

The variables used in this algorithm can be explained as follows:

- δ_i is the annotated disjunct of W_i which is obtained from the previous ADJ Algorithm,
- d_i is a temporary variable for checking a certain δ_i of W_i,
- *temporary_word*[*i*] is a temporary buffer to store W_i’,
- x_i is for identifying the position of W_i before shifting to the correct position, where x_i has a value from 1 to *n*,
- *target_word*$_i$ is for locating the target word in the *i*th position of *TS*.

In the Sentence-Based Transfer Rules Algorithm, line 1.1 checks whether all disjuncts of an input sentence equal certain annotated disjuncts and then store all target words in a *temporary_word* variable. If the input sentence is “She saw the saw”, and if each word disjunct match with certain annotated disjunct, then all target words in “Dia melihat itu gergaji” are stored in a *temporary_word* variable. For example, line 1.1

will store the Indonesian word “itu” (the) in the 3rd position of *temporary_word* (see Figure 3.9). Assigning x_i = 3 when i = 4 in line 2.1 will then shift the word “itu” from the 3rd position to the 4th position in the *TS* (see Figure 3.11).

Another result produced by Sentence-Based Transfer Rules Algorithm can be seen in Figure 3.12 which illustrates translation from two English sentences “She saw the saw. What saw is that?” into grammatically correct Indonesian sentences.

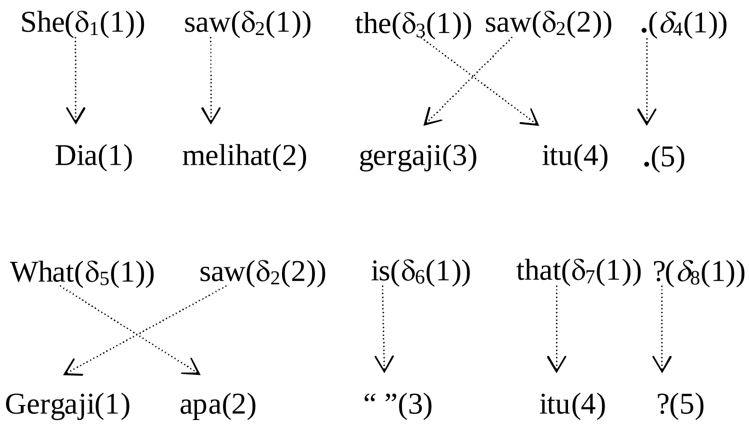


Figure 3.12: Illustration of the translation of two sentences

3.3.2 Phrase-based Transfer Rules

The disjunct annotation in the ADJ set represents the uniqueness of word pair alignment. This disjunct provides another information since it also describes the word element (POS) in the *SS*. For example, “red” is an adjective to the noun “saw” since “red” has a disjunct of (( )(*A*)), which expresses that “red” is connected with an A connector to a word on its right. This explains that “red” is an adjective to the noun “saw”. Another example is the word “the” which has a disjunct (( )(*D*)). From Figure 3.6, it can be seen that this disjunct forms a *D* link that connects “the” with the word “saw”. This means that “the” is a determiner to the noun “saw”. These phenomena are useful in the developed Transfer Rules Algorithm. This transfer rules are sentence-based since they take into account all disjuncts of all words in a *SS*. This needed tedious work in the development of the transfer rules for all cases. In other words, transfer rules generalization for similar cases in the translation process was never obtained.

Koehn et al. [64] explained that many SMT systems have improved their quality with the use of phrase-based translation, such as the alignment template model in Och et al. [86] which can be reframed as a phrase translation system, the syntax-based translation system in Yamada and Knight [127] which used phrase translation, and the joint-probability model for phrase translation which was introduced in Marcu and Wong [71]. In addition, DeNeefe et al. [32] compared and contrasted the advantages and disadvantages of the syntax-based MT model with the phrase-based SMT model. The comparison was based on two models of the most successful systems in the NIST 2006 MT evaluation [81]. The first model is the statistical phrase-based model [88] and the second is the statistical string-to-tree model [44]. They mentioned that the phrase-based model can consistently gain all the phrase translations based on the computed word alignment, concatenate and reorder those phrases using several cost models. While the strength of the syntac-based model is that the translation results tend to be syntactically well-formed and reusable for other syntactic purpose. The weakness of the phrase-based model is the lack of syntactic knowledge while the disadvantage of the syntac-based model is on the phrasal coverage and the decoding efficiency. They viewed the possibility to gain insights from the strengths of the phrase-based extraction model to increase both the phrasal coverage and translation accuracy of the syntax-based model. Bond and Shirai [22] also stated that generally transfer phrase translation contributes to the better results of the sentence translation.

Those findings motivate us to incorporate phrase translation method into our ADJ-based method. This is done by dividing the *SS* into phrases before the mapping process. Based on the ADJ set of each phrase, each source phrase is mapped into a target phrase. All target phrases are then merged to obtain the *TS*. The following briefly explains two examples on how our phrase-based transfer rules can handle standard case and non-standard case translations respectively. The definitions of both cases are taken from Al-Adhaileh and Tang [9]. A standard case is a projective correspondence e.g. one-to-one word foreseeable mapping. While a non-standard case is not projective correspondence e.g. scrambling, cross serial dependencies, etc.

A. Handling Standard Cases

Example of a standard case translation is the mapping of an English noun phrase to an Indonesian noun phrase, since this kind of English phrase is one of phrases which is frequently translated in one-to-one mapping. English noun phrase can be in forms such as:

- determiner + noun e.g. “the car”,
- determiner + superlative adjective + noun e.g. “the best car”,
- determiner + adjective + noun e.g. “the red car”,
- determiner + adjective + adjective + noun e.g. “the big red car”,
- possessive adjective + noun e.g. “his car”,
- noun-modifier + noun e.g. “car seat”.

How the ADJ method can solve noun phrase translation is explained by examining the most difficult case i.e. “the big red car”, which involves multiple adjectives. Since Link Parser is sentence-based, the phrase has a linkage only in a sentence e.g. “She saw the big red car” (see Figure 3.13). In this sentence, the ADJ set for the phrase is {(*the*,*itu*,(( )(*D*))), (*big*,*besar*,(( )(*A*))), (*red*,*merah*,(( )(*A*))), (*car*,*mobil*,((*O*,*D*,*A*)( )))}. Afterward, our system needs to decompose the input sentence into phrases. Our definition of a phrase is an extension to what was given by Zens et al. [134] and Chiang [26]:

> “… a phrase is simply a sequence of words. So the basic idea of phrase-based translation (PBT) is to segment the given source sentence into phrases, then to translate each phrase and finally compose the target sentence from these phrase translations …” [134]

> “… phrases, that is, substrings of potentially unlimited size (but not necessarily phrases in any syntactic theory).” [26]

What is identified as a phrase here is referred to either a single word (in case the word is not part of any phrase) or a collection of words with specific connectors. For example, a noun phrase that is made up of single/multiple adjectives (with *A* right connector) followed by a noun (with *A* left connector) is defined as an adjective-noun phrase. Meanwhile, a determiner-noun phrase is defined as it consists of a determiner and a noun / noun phrase. Thus, the phrase “the big red car” will be decomposed into two phrases as can be seen in Fig 3.13. The first decomposition results in an adjective-noun phrase “big red car”. These adjectives (“big” and “red”) precede the noun “car”. It is not valid for the corresponding Indonesian phrase since adjectives (“besar” and “merah”) always follow the noun “mobil” (car). Hence, the mapping is done by swapping the target adjectives and noun. This swapping process is done when a phrase transfer rules identify source words with *A* right connector followed by another source word which contains *A* left connector. This swapping technique is resolved using stack implementation. Afterward, this grammatical target words “mobil merah besar” is grouped into a target adjective-noun phrase.

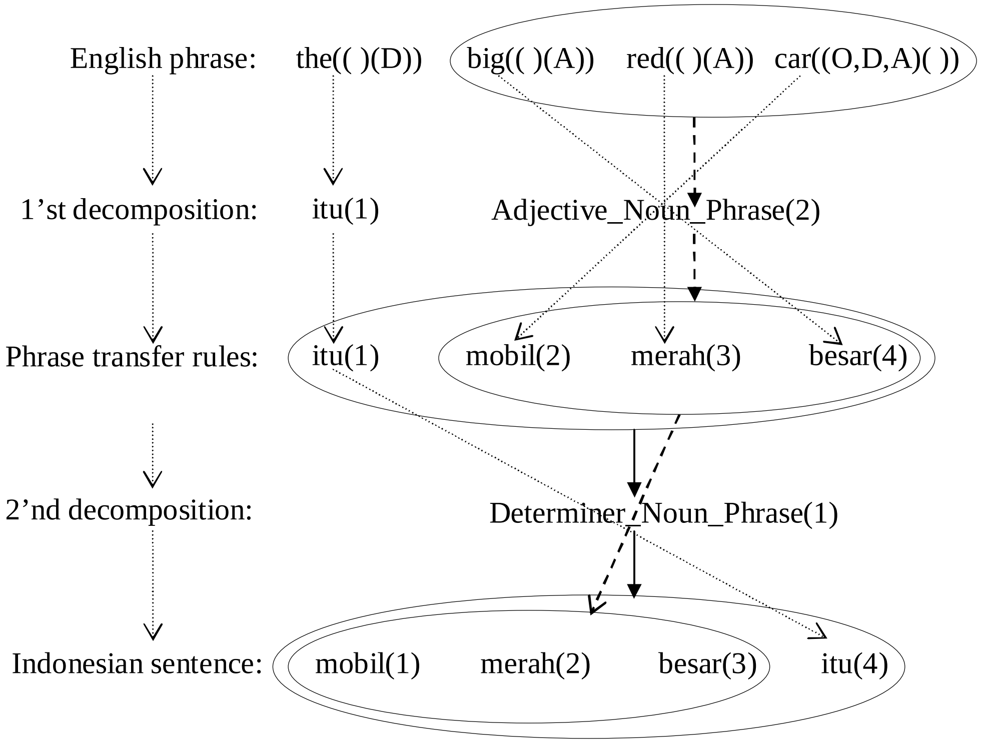


Figure 3.13: English to Indonesian phrase-based transfer rules of a standar case

The second decomposition yields an English determiner-noun phrase, which consists of a determiner “the” and a noun phrase “big red car”. This determiner “the” precedes the noun phrase. Meanwhile, the corresponding Indonesian phrase must have its determiner “itu” (the) following the noun “mobil” (car). This problem is resolved by swapping the target determiner and noun which is done when the phrase transfer rules identify words with *D* right connector followed by another word or noun phrase which contains *D* left connector. Finally, a grammatical Indonesian phrase “mobil merah besar itu” is obtained.

B. Handling Non-Standard Cases

Non-standard phenomena exist in the translation of English phrases to Indonesian phrases, for example the phrase “picks up” in the sentence “John picks the box up” is translated into the Indonesian word “mengambil” (see Figure 3.14).

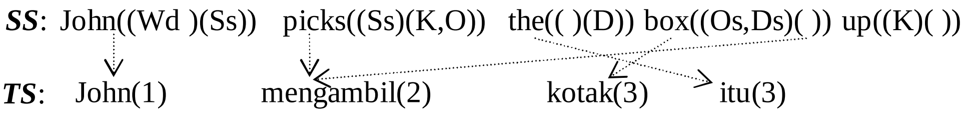

SS: John((Wd)(Ss)) picks((Ss)(K,O)) the(( )(D)) box((Os,Ds)( )) up((K)( ))
TS: John(1) mengambil(2) kotak(3) itu(3)

Figure 3.14: Phrase-based translation of a non-standard case

In this translation, many-to-one word mapping exists where two words “picks” and “up” in the *SS* correspond to one word “mengambil” in the *TS*. The ADJ set for the source phrase are {(*picks*,*mengambil*,((*Ss*)(*K*,*O*))), (*up*,“ ”,((*K*)( )))}. Thus, the alignment is resolved by mapping “up” into “ ” (blank) if “up” has a disjunct ((*K*)( )). However, the technique fails if the same words “picks” and “up” appear in different context, for example in the sentence “John picks the flower up on the hill” where “up” is translated differently although the disjunct is the same. Thus, it comes to conclusion that the ADJ-based method cannot solve non-standard cases as predicted. At present, these cases are solved by incorporating the direct method as explained by Novento [84].

How does the system translate a sentence with multiple phrases? This is possible since the main transfer rules algorithm has two layers. The first layer is an algorithm for mapping each source phrase into its target phrase. This layer has two functions: *Identify_source_phrases*( ) and *transfer_rules_algorithm*( ) function. The second layer is an algorithm for merging all target phrases into a correct *TS*. Figure 3.15 depicts two layers of the transfer rules module.

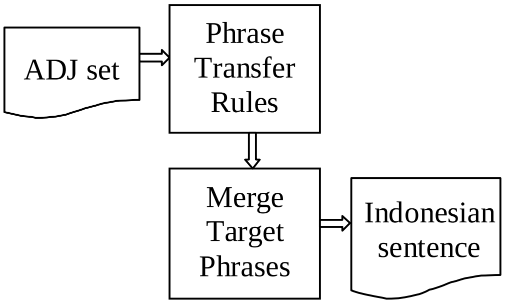


Figure 3.15: Diagram of phrase-based transfer rules

The algorithm for applying the diagram is given by Algorithm 3.4.

Algorithm 3.4 Phrase-Based Transfer Rules Algorithm

```
1.   Identify_source_phrases(ADJ_SET).
2.   For each source phrase, do step 2.1.
2.1      Run transfer_rules_algorithm(en_phrase_words,
         in_phrase_words, en_disjuncts) to obtain target_phrases.
3.   Compose_sentence(target_phrases).
```

The variables and functions used in this algorithm are explained as follows:

- *ADJ_SET* is a set of source words, target words, and the source word disjuncts for an English sentence,
- *transfer_rules_algorithm*( ) is a function to map a sequence of source words into a correct sequence of target words (see Sub Section 3.3.1),
- *en_phrase_words* is the English phrase words, *in_phrase_words* is the Indonesian phrase words, and *en_disjuncts* is the English phrase word disjuncts,

- *compose_sentence*( ) is for composing correct *TS* from all target phrases,
- *target_phrases* is all Indonesian phrases as results of line 2.1.

In this Phrase-Based Transfer Rules Algorithm, line 1 identifies all phrases in the *SS*. If an input sentence is “That might be the car”, the word “that” and two phrases (“might be” and “the car”) are identified (see Fig 3.16). Note in this stage that if a word does not belong to any phrase than it is considered as a phrase with a single word. The tranfer_rules_algorithm will translate from the most left, from the word “That” which is translated into Indonesian “Itu”, to the phrases “might be” and “the car” which are translated into Indonesian phrases “mungkin” and “mobilnya”. The next step is to group these target phrases. As there are no longer available phrase in the *SS*, the *Compose_sentence*( ) function merges the target word “Itu” and all the target phrases (“mungkin” and “mobil itu”) into a complete Indonesian sentence “Itu mungkin mobilnya”.

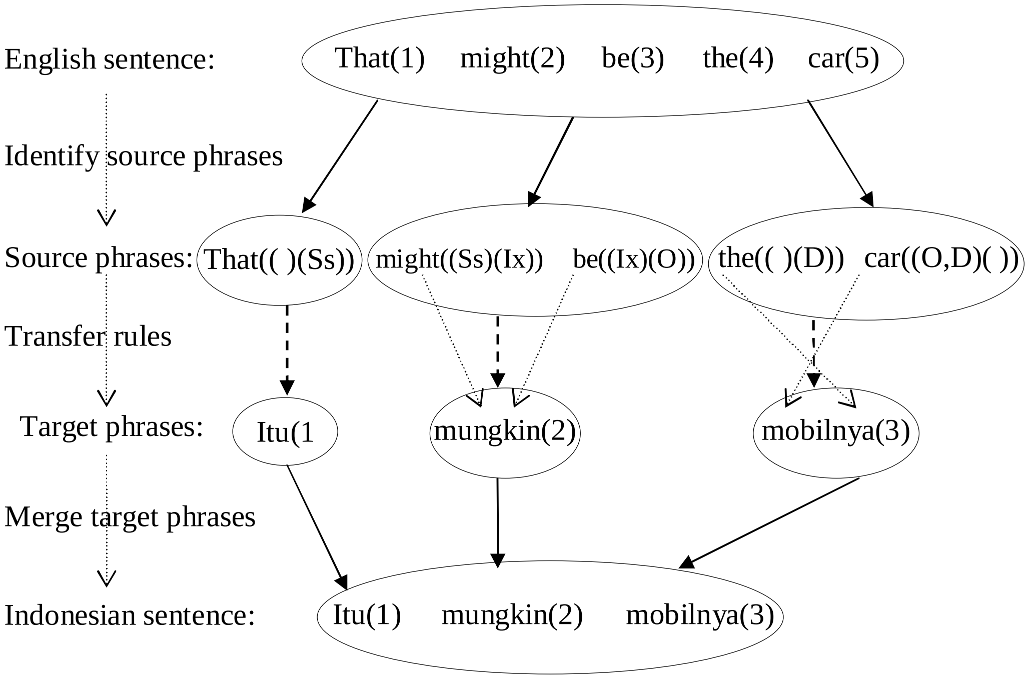


Figure 3.16: Translation of multiple phrases using phrase-based transfer rules

3.3.3 Hierarchical Phrase-based Transfer Rules

Previous sub section explains about how to decompose a *SS* into phrases, translate these phrases into their target phrases, and finally compose the target phrases into a correct *TS*. However, problems arise during the decomposion of hierarchical phrases. The problems are solved by the implementation of hierarchical phrase-based transfer rules discussed in this sub section.

Chiang [26] presented a hierarchical phrase-based MT system that performed significantly better than the Alignment Template System (a state-of-the-art phrase-based system proposed by Och and Ney [88]) in a comparison using BLEU as a metric of translation precision. Additionally, Lopez [70] stated the following statement for the implementation of hierarchical phrase-based MT system:

> "Given an input sentence, efficiently find all hierarchical phrase-based translation rules for that sentence in the training corpus." [70]

Both achievements encourage us to incorporate hierarchical phrase translation method into the ADJ-based method. How do the hierarchical phrase-based transfer rules work? After looking through our example data of 300 training sentences, these sentences can be assumed to have three layers of phrases maximally. This findings let us to define a hierarchical phrase as a phrase that consists of sub phrases where each sub phrase may or may not consist of sub sub phrases, which has more explanation but is not opposed to the definition by Chiang [26]:

> "… hierarchical phrases—phrases that contain subphrases." [26]

We calls the first top phrase layer (or a phrase that consists of two phrase layers) as the third group phrase, the second top phrase layer (or a phrase that consists another phrase layer) as the second group phrase, and the last layer from the top (or a phrase that no longer consists any phrase layer) as the first group phrase. Each of these groups is solved by generating what we call first group transfer rules, second group transfer rules, and third group transfer rules respectively. The diagram of the transfer rules are thus divided into three as seen in Figure 3.17.

This diagram explained that the transfer rules receive an ADJ set as its input. Based on the source words and annotated disjunct, which are the elements of the ADJ set, the transfer rules then classify which of the sequence of source words can be categorized as the first group phrase.

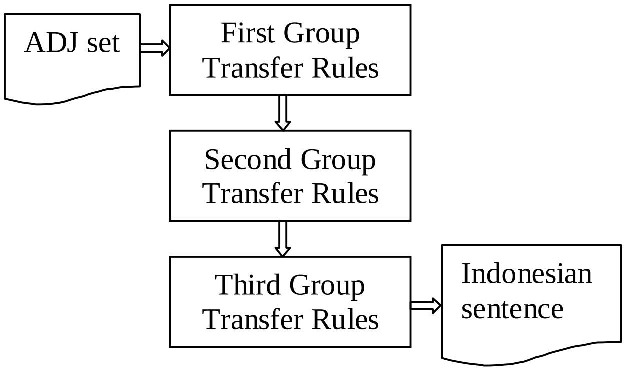

ADJ set
First Group Transfer Rules
Second Group Transfer Rules
Third Group Transfer Rules
Indonesian sentence

Figure 3.17: Diagram of hierarchical phrase-based transfer rules

When a first group phrase is found, first group transfer rules process and translate the phrase into a correct first group target phrase. The transfer rules then identify other sequence which can be categorized as the second group phrase, and translates the phrase into second group target phrase while repositioning the first group target phrase inside the second group target phrase. Finally, the transfer rules categorize other sequences of the third group phrase, translate the phrase into the third group target phrase, and reposition the second group target phrase inside the third group target phrase. The diagram in Figure 3.17 can be expressed as Algorithm 3.5.

Algorithm 3.5 Hierarchical Phrase-Based Transfer Rules Algorithm

```
1.    Obtain source words and annotated disjuncts from ADJ set.
2.    For each $W_i$ ($i = n$ to 1), do steps 2.1, 2.2, and 2.3.
2.1       If first group source phrase is found, then apply first group
          transfer rules. Decrement $i$, then do step 2.2.
2.2       If second group source phrase is found, then apply second group
          transfer rules. Decrement $i$, then do step 2.3.
2.3       If third group source phrase is found, then apply third group
          transfer rules.
```

The flow chart diagram for applying the diagram in Figure 3.17 is illustrated in Figure 3.18.

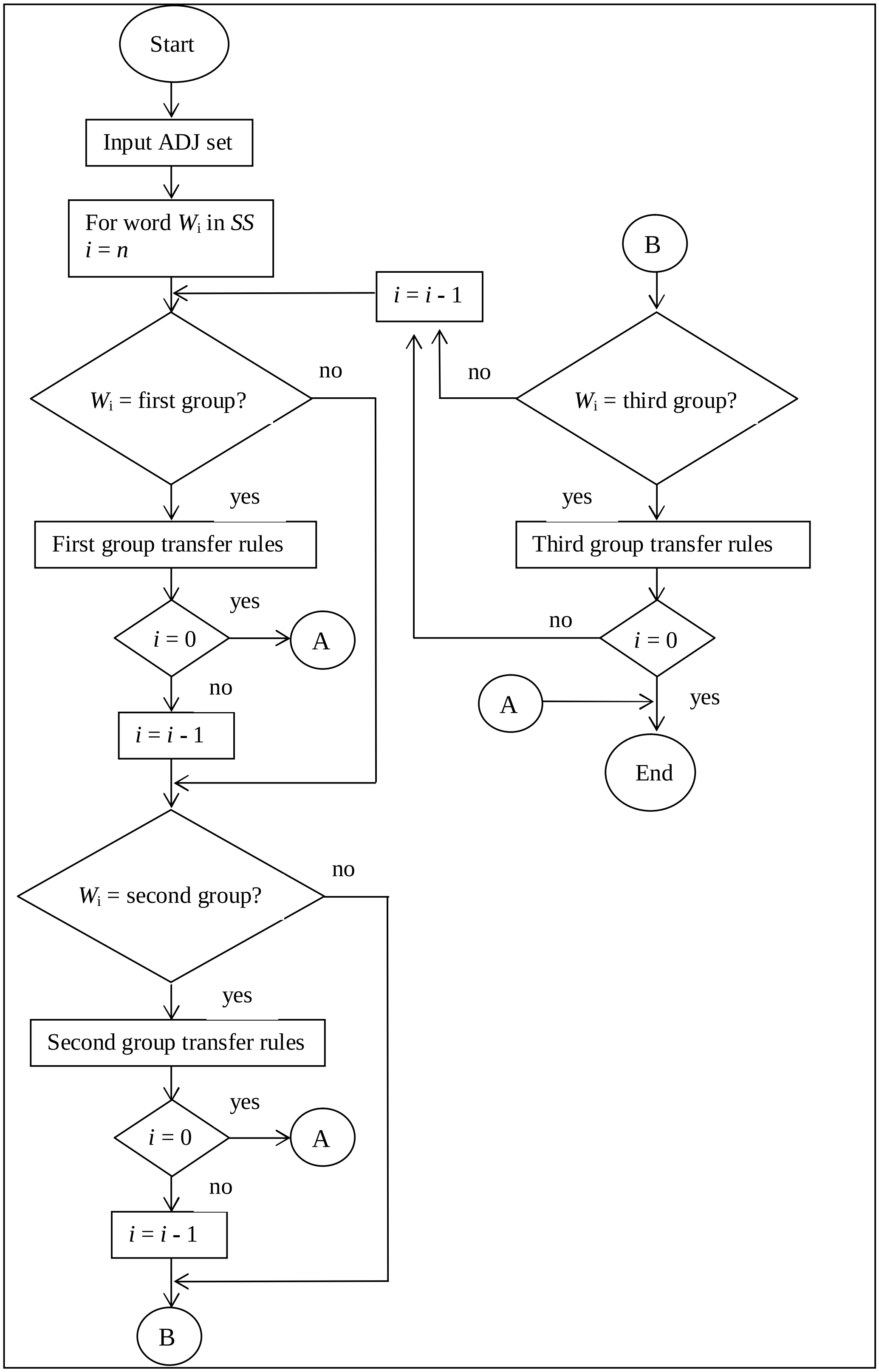


Figure 3.18: Flowchart diagram of hierarchical phrase-based transfer rules

The variables used in Algorithm 3.5 are already explained in Sub Sections 3.3.1 and 3.3.2. In this Hierarchical Phrase-Based Transfer Rules Algorithm, line 1 obtains source words (i.e. English words) and their annotated disjuncts. Line 2.1 identifies all first group English phrases, which consist of words (e.g. prenominal adjectives, superlative adjectives, and noun-modifiers) that modify nouns. If the input sentence and its linkage are given as in Figure 3.19, the seventh word "?" (W_7) is evaluated first and it is found that "?" is not in the first group phrase. Since the condition in line 2.1 is not fulfilled then it goes to line 2.2, which checks that "?" is not in the second group phrase either. Line 2.3 also checks that "?" do not belong to the third group phrase and the step goes back to line 2 again to decrease *i* to allow the evaluation of the sixth word "go" (W_6). Line 2.1 until 2.3 also found that "go" is not in any group of phrase. The same results obtained in the evaluation of the fifth word "car" (W_5) where this word also does not belong to any group of phrase. The evaluation goes to the fourth word "red" (W_4) to find that the phrase "red car" is identified as the first group English phrase since "red" has an empty left connector and *A* right connector. These two connectors describe that "red" is the adjective of "car". This indicates that "red car" is an adjective-noun phrase which is then translated into the Indonesian words "mobil merah" by the first group transfer rules. How the first group transfer rules work is the same as the work of the phrase-based transfer rules as already explained in Sub Section 3.3.2 (see Figure 3.13). The number stored in variable *i* is decreased by 1, meaning that the third word "the" (W_3) is now being evaluated in line 2.2, which identifies that this word is the starting point for the second group English phrase (i.e. phrases that consist of demonstrative pronouns or determiner "the", possessive adjectives, and possessive nouns) because it has an empty left connector and a *D* right connector. The second group transfer rules, which is explained in detail in Sub Section 3.4.2, then translate "the red car" (the second group English phrase) into a grammatical Indonesian phrase "mobil merah itu". The value of *i* is decremented by 1 to allow the evaluation of the second word "will" (W_2). The phrase starts with "will" is not classified into any group of phrase. The evaluation goes to the first word "Where" and it is found that the phrase starts with a phrase "Where will" satisfies the condition in line 2.3. This means that the phrase started with "Where will" and ended with the punctuation "?" is of the third group phrase in which the third transfer rules, which are explained in detail in Sub Section 3.4.3, is applied to translate the entire sentence into “Kemana mobil merah itu akan pergi?” (see Figure 3.20).

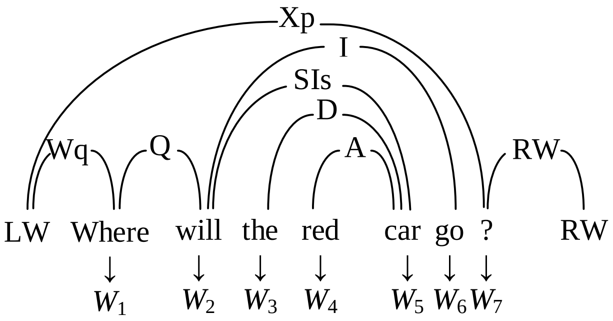

Xp
I
SIs
D
Wq
Q
A
RW
LW Where will the red car go ? RW
W_1 W_2 W_3 W_4 W_5 W_6 W_7

Figure 3.19: An input English sentence that consists of a hierarchical phrase

Note in the linkage (see Figure 3.19) that *LW* is the left wall/border, *RW* is the right wall/border, *Wq* link connects the subjects of main clauses to the wall in most questions (except yes-no questions), *Q* link connects the question word to the auxiliary in where-when-how questions, *SIs* connects singular subject nouns to finite verbs in cases of subject-verb inversion, *D* link connects determiners to nouns, *A* connects prenominal adjectives to nouns, *I* connects certain words (such as modal and “to”) with infinitive verb forms, and *Xp* link connects punctuation symbols either to another punctuation or to words. The translation mapping can be seen in Figure 3.20.

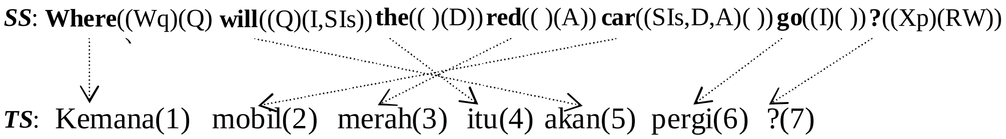

SS: **Where**((Wq)(Q) **will**((Q)(I,SIs)) **the**(( )(D)) **red**(( )(A)) **car**((SIs,D,A)( )) **go**((I)( )) **?**((Xp)(RW))
TS: Kemana(1) mobil(2) merah(3) itu(4) akan(5) pergi(6) ?(7)

Figure 3.20: An English-Indonesian mapping of a sentence that contains the first, second, and third group phrases

For more clear explanation, the hierarchical phrase-based transfer rules will be explained in separate section, which includes the discussion of the first, second, and third group transfer rules with one example for each of these transfer rules as follows.

3.4 Hierarchical Phrase-based Transfer Rules

32 transfer rules have been developed for the ADJ-based MT system and these rules are categorized into three groups. Those three groups and the examples of the transfer rules belonging to each of the groups are described in the following sub sections.

3.4.1 First Group Transfer Rules

This group consists of simple rules, which handle phrases consisting of words (e.g. prenominal adjectives, superlative adjectives, and noun-modifiers) that modify nouns, handle phrases of adverbs modifying adjectives, and handle phrases of determiners followed by nouns in idiomatic time expressions.

A. Rule for phrases consisting of prenominal adjectives that modify nouns

In English grammar, prenominal adjectives always precede nouns. Oppositely, in Indonesian grammar, adjectives always come after nouns. Therefore, the translation of an English phrase consisting of those kinds of words into Indonesian is done with the swapping between the Indonesian prenominal adjective and noun. Let us consider an English phrase "red car" with its link as seen in Figure 3.21.

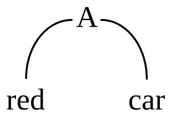


Figure 3.21: A phrase with *A* link connecting a prenominal adjective and a noun

The linkage in Figure 3.21 has an *A* link which shows the prenominal adjective "red" which precedes the noun "car". The English-to-Indonesian words mapping of the phrase is illustrated in Figure 3.22. The mapping is divided into two processes. Process A translates each word in the English phrase into the Indonesian word. The English phrase "red car" consists of two words, "red" and "car".

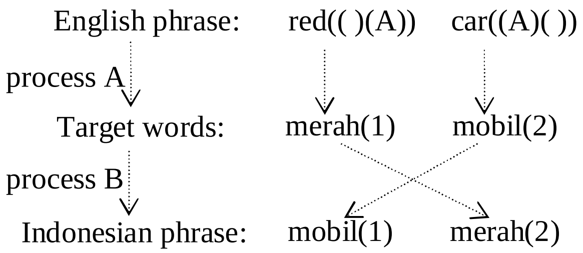


Figure 3.22: Translation mapping of prenominal adjective-noun phrase

The word "red" has an empty left connector and a *A* right connector. The word "car" has a *A* left connector and an empty right connector. Based on these source words and their disjuncts, the English prenominal adjective "red" is translated into the Indonesian adjective "merah" and the English noun "car" is translated into the Indonesian noun "mobil". Thus, process A results in the Indonesian target words "merah mobil". Subsequently, process B swaps the position of the target words "merah" and "mobil" to get a grammatical Indonesian phrase "mobil merah". This mapping is implemented in the following IF-THEN statement.

```
0. IF (ADJ.$W_i$.right_connect.Contains(A) & ADJ.$W_{i+1}$.left_connect.Contains(A))
1. THEN
2. {      temporary_word[i] ← ADJ.$W_i$';
3.        temporary_word[i+1] ← ADJ.$W_{i+1}$';
4.        $word_i$ ← temporary_word[i+1];
5.        $word_{i+1}$ ← temporary_word[i];  }
```

Line 0 identifies prenomial adjectives by checking whether the first English word "red" (*ADJ.W_i*) has a disjunct which contains the *A* right connector and identifies nouns by checking whether the second English word "car" (*ADJ.W_{i+1}*) has a disjunct which contains the *A* left connector. If both conditions are fulfilled, lines 2-3 stores the translated words ("merah" and "mobil") into temporary variables. Lines 4-5 swaps the positions of both Indonesian words into a grammatical phrase "mobil merah".

B. Rule for phrases consisting of superlative adjectives that modify nouns

Superlative adjectives always precede nouns in English grammar. This condition is opposite to the Indonesian grammar where superlative adjectives always follow nouns. Hence, the translation is accomplished by the swapping of the Indonesian superlative adjective and noun. For example, assume an English phrase “best car” and its link as given in Figure 3.23.

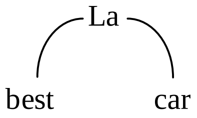


Figure 3.23: A phrase with *La* link connecting a superlative adjective and a noun

La link in Figure 3.23 explains the superlative adjective “best” that precedes the noun “car”. The translation of the phrase is illustrated in Figure 3.24.

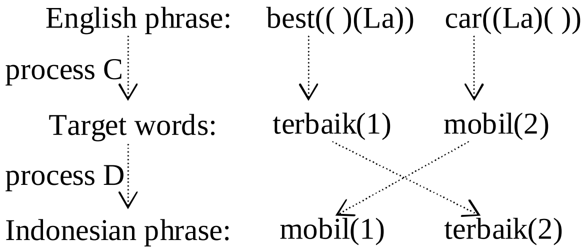


Figure 3.24: Translation of superlative adjective-noun phrase

The mapping is similar with the mapping for the prenomial adjective case. Process C translates each word in the English phrase into its target word in Indonesian. This translation process into Indonesian words “terbaik mobil” is done if: (1) the word “best” has an empty left connector and a *La* right connector, and (2) the word “car” has a *La* left connector and an empty right connector. Subsequently, process D swaps the position of the target words “terbaik” and “mobil” to get the correct Indonesian phrase “mobil terbaik”. This mapping is implemented as follows.

```
0. IF (ADJ.$W_i$.right_connect.Contains(La) & ADJ.$W_{i+1}$.left_connect.Contains(La))
1. THEN
2. {     temporary_word[i] ← ADJ.$W_i$';
3.       temporary_word[i+1] ← ADJ.$W_{i+1}$';
4.       $word_i$ ← temporary_word[i+1];
5.       $word_{i+1}$ ← temporary_word[i];   }
```

Line 0 identifies superlative adjectives by checking whether the first English word (*ADJ.*W_i) has a disjunct which contains the *La* right connector and identifies nouns by checking whether the second English word (*ADJ.*W_{i+1}) has a disjunct which contains the *La* left connector. If both conditions are satisfied, the translated words (*ADJ.*W_i' and *ADJ.*W_{i+1}') of both English words are stored into temporary variables by lines 2-3. Lines 4-5 swaps the positions of both Indonesian words into the grammatically correct Indonesian phrase “mobil terbaik”.

C. Rule for phrases consisting of noun-modifiers that modify other nouns

Noun-modifiers always precede nouns in English grammar which is opposite to Indonesian grammar. Thus, the translation is correct after the swapping between the Indonesian noun-modifier and noun. Consider an English phrase “car seat” and its link as seen in Figure 3.25.

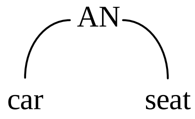


Figure 3.25: A phrase with *AN* link connecting a noun-modifier and a noun

The AN link in Figure 3.25 describes the noun-modifier “car” precedes the noun “seat”. The translation of the phrase is depicted in Figure 3.26.

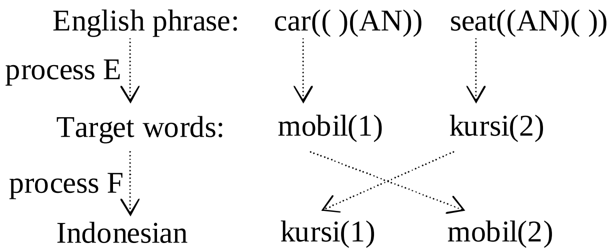


Figure 3.26: Translation of a phrase with a noun-modifier modifying a noun

The mapping is divided into two processes. Process E translates each word in the English phrase into the Indonesian word if both following conditions are satisfied: (1) the word “car” has an empty left connector and an *AN* right connector, (2) the word “seat” has an *AN* left connector and an empty right connector. Subsequently, process F swaps the position of the target words “mobil” and “kursi” to get the grammatical Indonesian phrase “kursi mobil”. The IF-THEN implementation is given as follows.

```
0. IF ($ADJ.W_i$.right_connect.Contains(AN) & $ADJ.W_{i+1}$.left_connect.Contains(AN))
1. THEN
2. {     temporary_word[i] ← $ADJ.W_i$’;
3.       temporary_word[i+1] ← $ADJ.W_{i+1}$’;
4.       $word_i$ ← temporary_word[i+1];
5.       $word_{i+1}$ ← temporary_word[i];     }
```

Line 0 identifies noun-modifiers by checking whether the first English word (*$ADJ.W_i$*) has a disjunct which contains the *AN* right connector and identifies nouns by checking whether the second English word (*$ADJ.W_{i+1}$*) has a disjunct which contains the *AN* left connector. If both conditions are true, the translated words (*$ADJ.W_i$*’ and *$ADJ.W_{i+1}$*’) of both English words are stored into temporary variables by lines 2-3. Lines 4-5 swaps the positions of both Indonesian words “mobil” and “kursi” into the grammatical Indonesian phrase “kursi mobil”.

D. Rule for phrases consisting of adverbs modifying adjectives

Adverbs always precede nouns in English grammar which is sometimes in contrary to Indonesian grammar, for example in the case of phrases containing a sequence of adverb-adjective-noun. Thus, the translation is correct after the swapping between the Indonesian adverb and adjective. Consider an English phrase “very big car” with its link as can be seen in Figure 3.27.

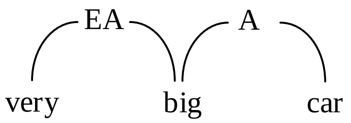


Figure 3.27: A phrase with *EA* link connecting an adverb and an adjective

EA link in Figure 3.27 explains the adverb “very” precedes the adjective “big”. The translation of the phrase is illustrated in Figure 3.28.

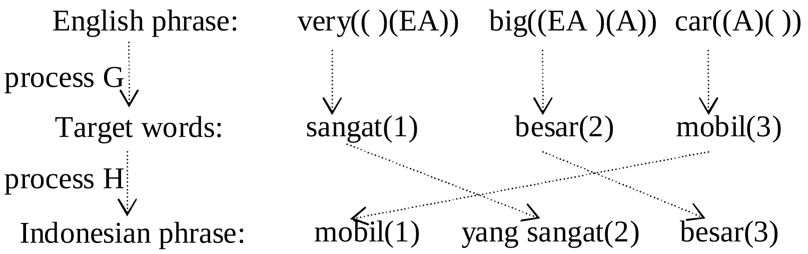


Figure 3.28: Translation of a phrase with an adverb modifying an adjective

In this case, “very” is translated by process G into two Indonesian words “yang sangat”, although normally only translated into a single word “sangat” in most other cases. Process H rearranges the position of the target words “sangat”, “besar”, “mobil”, and putting the word “yang” in front of the word “sangat” to obtain the grammatical Indonesian phrase “kursi mobil yang sangat besar”. The IF-THEN implementation is as follows.

```
0. IF ($ADJ.W_i$.right_connect.Contains(EA) & $ADJ.W_{i+1}$.left_connect.Contains(EA)
        & $ADJ.W_{i+2}$.left_connect.Contains(A))
1. THEN
2. {      temporary_word[i] ← $ADJ.W_i$’;
3.        temporary_word[i+1] ← $ADJ.W_{i+1}$’;
4.        temporary_word[i+2] ← $ADJ.W_{i+2}$’;
5.        $word_i$ ← temporary_word[i+2];
6.        $word_{i+1}$ ← “yang” + temporary_word[i];
7.        $word_{i+2}$ ← temporary_word[i+1];        }
```

Line 0 identifies adverbs by checking whether the first English word (*$ADJ.W_i$*) has a disjunct which contains the *EA* right connector, identifies adjectives by checking whether the second English word (*$ADJ.W_{i+1}$*) has a disjunct which contains the *EA* left connector, and identifies nouns by checking whether the third English word (*$ADJ.W_{i+2}$*) has a disjunct which contains the *A* left connector. If these three conditions are fulfilled, the translated words (*$ADJ.W_i$*’, *$ADJ.W_{i+1}$*’, and *$ADJ.W_{i+2}$*’) are stored into temporary variables by lines 2-4. Lines 5-7 rearranges the positions of these translated words while line 6 also puts the word “yang” in front of the word “sangat” to get the grammatical Indonesian phrase “mobil yang sangat besar”.

E. Rule for phrases consisting of determiners followed by nouns in idiomatic time expressions

Determiners always precede nouns in idiomatic time expressions in English grammar. This condition is opposite to the Indonesian grammar. Hence, the translation is accomplished by the swapping between the Indonesian determiner and noun. For example, assume an English phrase “best car” with its link as given in Figure 3.29.

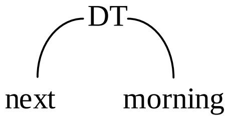


Figure 3.29: A phrase that consists of *DT* link connecting a determiner and a noun in idiomatic time expressions

DT link in Figure 3.29 explains the determiner “next” that precedes the noun “morning”. The mapping of the phrase is illustrated in Figure 3.30.

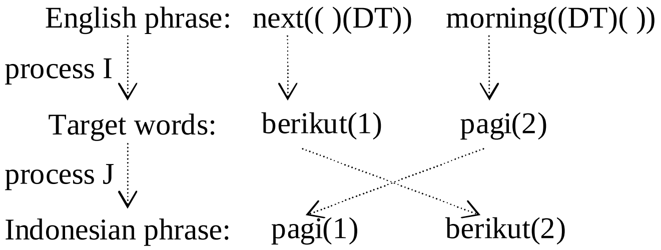


Figure 3.30: Translation of determiner-noun phrases in idiomatic time expressions

The mapping is divided into two processes. Process I translates each word in the English phrase into the Indonesian word if both the following conditions are fulfilled: (1) the word “next” has an empty left connector and a *DT* right connector, (2) the word “morning” has a *DT* left connector and an empty right connector. Subsequently, process J swaps the position of the target words “berikut” and “pagi” to get the grammatical Indonesian phrase “pagi berikut”. The IF-THEN implementation is given as follows.

```
0. IF (ADJ.$W_i$.right_connect.Contains(DT) & ADJ.$W_{i+1}$.left_connect.Contains(DT))
1. THEN
2. {     temporary_word[i] ← ADJ.$W_i$’;
3.       temporary_word[i+1] ← ADJ.$W_{i+1}$’;
4.       $word_i$ ← temporary_word[i+1];
5.       $word_{i+1}$ ← temporary_word[i];     }
```

Line 0 identifies determiners in idiomatic time expressions by checking whether the first English word ($ADJ.W_i$) has a disjunct which contains the *DT* right connector and identifies nouns by checking whether the second English word ($ADJ.W_{i+1}$) has a disjunct which contains the *DT* left connector. If both conditions are satisfied, the translated words ($ADJ.W_i$’ and $ADJ.W_{i+1}$’) of both English words are stored into temporary variables by lines 2-3. Lines 4-5 swaps the positions of both translated words into the grammatical Indonesian phrase “pagi berikut”.

3.4.2 Second Group Transfer Rules

This group consists of rules with more complexity than those of the first group, such as rules for phrases that consist of demonstrative pronouns or determiner "the", possessive adjectives, and possessive nouns.

A. *Rule for phrases consisting of demonstrative pronouns (e.g. "this" and "those" in "this car" and "those cars") or determiner "the"*

Demonstrative pronouns or determiner "the" always precede nouns or noun phrases in English grammar. These contradict with Indonesian grammar where Indonesian demonstrative pronouns or determiner "the" are located after nouns or noun phrases. Let us assume an English phrase "this red car" and its link as seen in Figure 3.31.

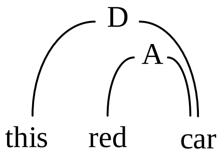


Figure 3.31: A phrase with *D* link connecting a demonstrative pronoun and a noun phrase

The actual links for demonstrative pronouns and determiner "the" vary with *Ds*, *Dsu*, *Dmc*, etc. Since the use of the complete name for these links in the implementation sometimes failed, it was decided not to utilize the link names. Hence, only the upper case letter is used for the links i.e. *D* link. However, this will obviously decrease the ability of the system to identify demonstrative pronouns or determiner "the" and to distinguish them from other word categories such as other determiners and possessive adjectives, which sometimes also can be identified with *D* link. For example, determiners such as "any" and "some" can also be identified by an empty left connector and a *D* right connector. However, they are translated in different way as in Figure 3.32 for "any". Therefore, the combination of *D* link and the source words themselves (i.e. the demonstrative pronouns) are used as the conditions in the implementation. Hence, in this thesis, all links for demonstrative pronouns are written as *D* link only for this particular explanation.

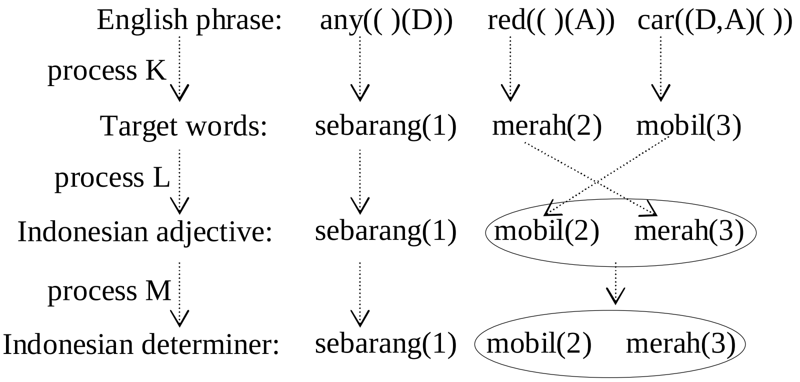


Figure 3.32: Translation of a phrase with a determiner “any”

D link in Figure 3.31 shows the demonstrative pronoun “this” which precedes the noun phrase “red car”. The translation of the phrase in Figure 3.31 is depicted in Figure 3.33.

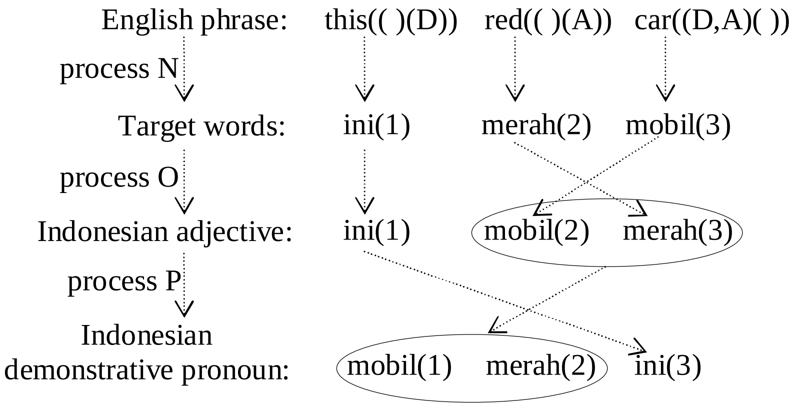


Figure 3.33: Translation of a demonstrative pronoun preceding a noun phrase

This second group transfer rule is done in three processes, if the following three conditions are true:

1) the word “this” has an empty left connector and a *D* right connector,
2) the word “red” has an empty left connector and an *A* right connector,
3) the word “car” has two left connectors (*D*, *A*) and an empty right connector.

Process N translates each word in the English phrase into the Indonesian target words “ini merah mobil”. Process O handles the allignment of the translated words “merah mobil” using the first group rules (see Sub Section 3.4.1) to get the correct phrase “mobil merah”. Subsequently, process P swaps the position of the target word “ini” and the phrase “mobil merah” to obtain the grammatical Indonesian phrase “mobil merah ini”. The mapping of the English phrase with a demonstrative pronoun is implemented in the following IF-THEN statement.

```
0. IF (ADJ.$W_i$ == English_demonstrative_pronoun || “the”)
1. THEN
2. {first_buffer ← ADJ.$W_i$’;
3.   IF (ADJ.$W_{i+2}$.left_connect.Contains(D))
4.   THEN second_buffer ← first_group_rules(ADJ.$W_{i+1}$, ADJ.$W_{i+2}$);
5.   swap between first_buffer and second_buffer; }
```

Line 0 identifies the demonstrative pronoun or determiner “the”. If it is true, line 2 stores the demonstrative pronoun or “the” into the first buffer. Line 3 identifies the noun phrase by checking whether the third English word ($ADJ.W_{i+2}$) has a disjunct which contains *D* left connector. If this condition is also true, line 4 applies the first group rules to the phrase that consists of the second and third word ($ADJ.W_{i+1}$ and $ADJ.W_{i+2}$). This first group rule determines that the phrase of “red car” is classified as a phrase which consists of a prenominal adjective modifying a noun. This is true because the word “red” has an empty left connector and an *A* right connector while the word “car” has an *A* left connector and an empty right connector. The first group rule yields a correct Indonesian noun phrase “mobil merah”. Line 5 swaps the positions of the Indonesian demonstrative pronoun (“ini”) stored in the first buffer and the Indonesian phrase (“mobil merah”) stored in the second buffer to obtain the grammatical Indonesian phrase “mobil merah ini”.

B. Rule for phrases consisting of possessive adjectives

Possessive adjectives always precede nouns or noun phrases in English grammar. This is not grammatically correct in Indonesian where possessive adjectives are located after nouns or noun phrases, such as “my red car” phrase with its link as in Figure 3.34.

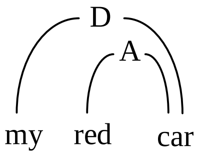


Figure 3.34: A phrase with *D* link connecting a possessive adjective and a noun phrase

The actual links for possessive adjectives vary with *Ds*, *Dmc*, etc. The same problem as in demonstrative pronouns arises where the use of the complete full link names are sometimes failed to identify the possessive adjectives. Thus, the combination of only the upper case letter for the link (*D*) and the possessive adjective themselves are used as the conditions in the implementation. *D* link in Figure 3.34 shows the possessive adjective "my" which precedes the noun phrase "red car". The mapping of the phrase "my red car" is given in Figure 3.35.

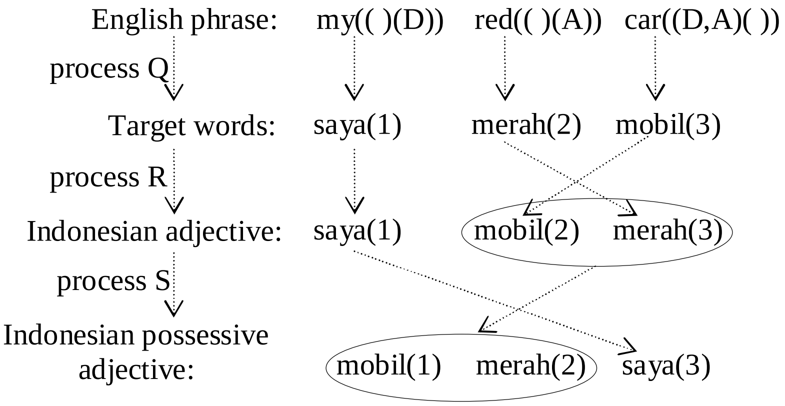


Figure 3.35: Translation of a possessive adjective preceding a noun phrase

As can be seen from Figure 3.35, as also already discussed previously in the rule for demonstrative pronouns and determiner "the", possessive adjectives share the same identification as that of demonstrative pronouns and determiner "the". They can be identified by an empty left connector and a *D* right connector. The transfer rule is done in three processes, if the following three conditions are satisfied:

1) the word “my” has an empty left connector and a *D* right connector,
2) the word “red” has an empty left connector and an *A* right connector,
3) the word “car” has two left connectors (*D*, *A*) and an empty right connector.

Process Q translates each word in the English phrase into the Indonesian target words “saya merah mobil”. Process R handles the allignment of the translated words “merah mobil” using the first group rules (see Sub Section 3.4.1) to get the correct phrase “mobil merah”. Subsequently, process S swaps the position of the target word “saya” and the phrase “mobil merah” to obtain the grammatical Indonesian phrase “mobil merah saya”. Thus, the IF-THEN statement of the English phrase with a possessive adjective mapping is implemented as follows.

```
0. IF (ADJ.$W_i$ == English_possessive_adjective)
1. THEN
2. {first_buffer ← ADJ.$W_i$’;
3.   IF (ADJ.$W_{i+2}$.left_connect.Contains(D))
4.   THEN second_buffer ← first_group_rules(ADJ.$W_{i+1}$, ADJ.$W_{i+2}$);
5.   swap between first_buffer and second_buffer; }
```

Line 0 identifies the possessive adjective. If it is true, line 2 stores the possessive adjective into the first buffer. Line 3 identifies the noun phrase by checking whether the third English word ($ADJ.W_{i+2}$) has a disjunct which contains *D* left connector. If this condition is also true, line 4 applies the first group rules to the phrase that consists of the second and third word ($ADJ.W_{i+1}$ and $ADJ.W_{i+2}$). This first group rule determines that the phrase “red car” is classified as a phrase which consists of a prenominal adjective modifying a noun. This is true because the word “red” has an empty left connector and an *A* right connector while the word “car” has an *A* left connector and an empty right connector. The first group rule yields a correct Indonesian noun phrase “mobil merah”. Line 5 swaps the positions of the Indonesian possessive adjective (“saya”) stored in the first buffer and the Indonesian phrase (“mobil merah”) stored in the second buffer to obtain the grammatical Indonesian phrase “mobil merah saya”.

C. Rule for phrases consisting of possessive nouns

Possessive nouns can be indicated by the use of apostrophe symbol in English grammar. This symbol is not used for the same purpose in Indonesian grammar. Instead, possessors are located after their possessions to show Indonesian possessive nouns. Consider an English phrase “Lutfi’s red car” and its link as seen in Figure 3.36.

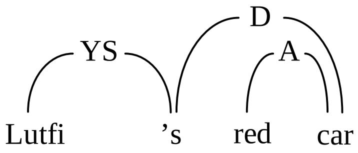


Figure 3.36: A phrase with *YS* link connecting apostrophe and a noun phrase

The English phrase has a *YS* link which shows the possessive noun indicator (apostrophe) which precedes the noun phrase “red car”. The word “Lutfi” is a name of an Indonesian person. The English-to-Indonesian words mapping is illustrated in Figure 3.37.

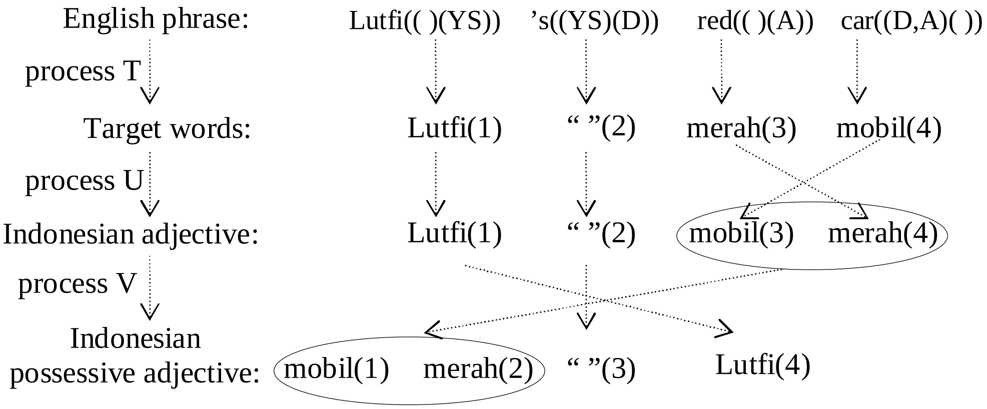


Figure 3.37: Translation of a phrase with a possessive noun

The mapping is divided into three processes if the following conditions are true:

1) the word “Lutfi” has an empty left connector and a *YS* right connector,
2) the apostrophe symbol has a *YS* left connector and a *D* right connector,
3) the word “red” has an empty left connector and an *A* right connector,
4) the word “car” has two left connectors (*D*, *A*) and an empty right connector.

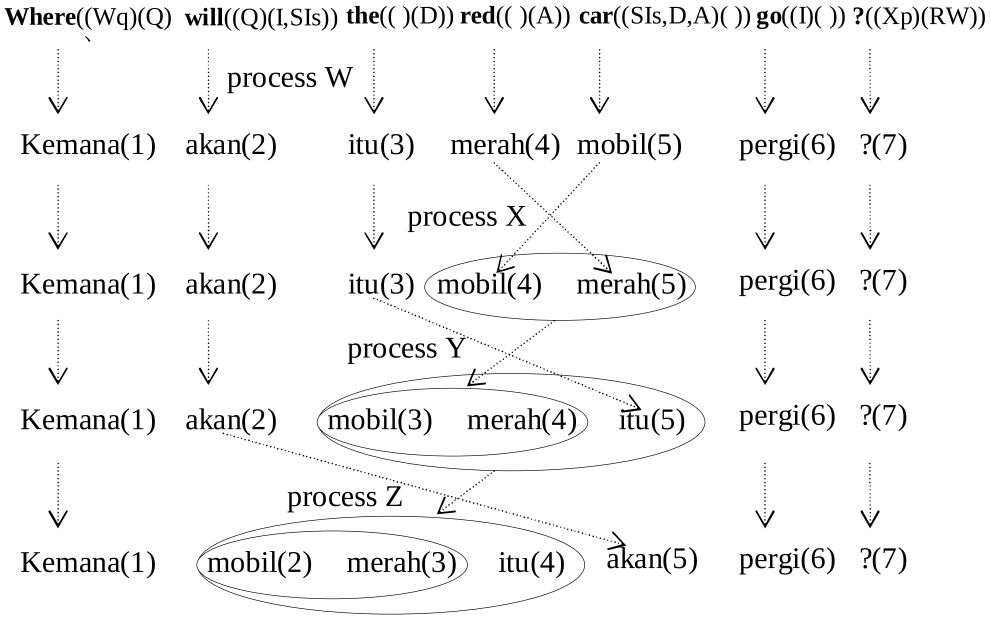
Process T translates each word in the English phrase into the Indonesian target words “Lutfi merah mobil”. Process U handles the allignment of the translated words “merah mobil” using the first group rules (see Sub Section 3.4.1) to get the correct phrase “mobil merah”. Subsequently, process V rearranges the position of the target words “ ” (blank), “Lutfi”, and the phrase “mobil merah” to obtain the grammatical Indonesian phrase “mobil merah Lutfi”. Thus, the IF-THEN statement of the English phrase with a possessive noun mapping is implemented as follows.

```
0. IF (ADJ.$W_i$.right_connect.Contains(YS) & ADJ.$W_{i+1}$.left_connect.Contains(YS))
1. THEN
2. {  first_buffer ← ADJ.$W_i$’;
3.    IF (ADJ.$W_{i+3}$.left_connect.Contains(D))
4.    THEN  second_buffer ← first_group_rules(ADJ.$W_{i+2}$, ADJ.$W_{i+3}$);
5.    swap between first_buffer and second_buffer;  }
```

Line 0 in identifies the possessive adjective. If it is true, line 2 stores the possessor (“Lutfi”) into the first buffer. Line 3 identifies the noun phrase by checking whether the fourth English word ($ADJ.W_{i+3}$) has a disjunct which contains *D* left connector. If this condition is also true, line 4 applies the first group rules to the phrase that consists of the third and fourth word ($ADJ.W_{i+2}$ and $ADJ.W_{i+3}$). This first group rule determines that the phrase “red car” is classified as a phrase which consists of a prenominal adjective modifying a noun. This is true because the word “red” has an empty left connector and an *A* right connector while the word “car” has an *A* left connector and an empty right connector. The first group rule yields a correct Indonesian noun phrase “mobil merah”. Line 5 swaps the positions of the possessor (“Lutfi”) stored in the first buffer and the Indonesian phrase (“mobil merah”) stored in the second buffer to obtain the grammatical Indonesian phrase “mobil merah Lutfi”.

3.4.3 Third Group Transfer Rules

The last group consists of rules with the most complexity, e.g. rules for handling phrases with interrogative words and modal auxiliaries. Figure 3.19 shows an English sentence along with its link while Figure 3.20 illustrates the English sentence mapping into an Indonesian sentence. The English sentence contains phrases with an interrogative word "Where", a modal auxiliary "will", a determiner "the", and a prenominal adjective "red". Hence, the mapping into Indonesian is done in three main stages; starts with the use of the first group rules to cope with the prenominal adjective, and then utilize the second group rules to handle the determiner "the", and ends with employing the third group rules to resolve the phrase "Where will". The complete English-to-Indonesian words mapping process is illustrated in Figure 3.38.



Where((Wq)(Q) **will**((Q)(I,SIs)) **the**(( )(D)) **red**(( )(A)) **car**((SIs,D,A)( )) **go**((I)( )) **?**((Xp)(RW))
process W
Kemana(1) akan(2) itu(3) merah(4) mobil(5) pergi(6) ?(7)
process X
Kemana(1) akan(2) itu(3) mobil(4) merah(5) pergi(6) ?(7)
process Y
Kemana(1) akan(2) mobil(3) merah(4) itu(5) pergi(6) ?(7)
process Z
Kemana(1) mobil(2) merah(3) itu(4) akan(5) pergi(6) ?(7)

Figure 3.38: Translation of phrases with interrogative words and modal auxiliaries

The mapping is divided into four processes if the following six conditions are fulfilled:

1) the word "Where" has a *Q* right connector,
2) the word "will" has a *Q* left connector and two right connectors (*I*, *SIs*),
3) the word "the" has an empty left connector and a *D* right connector,
4) the word "red" has an empty left connector and an *A* right connector,
5) the word "car" has three left connectors (*SIs*, *D*, *A*) and an empty right connector,
6) the word "go" has an *I* left connector and an empty right connector.

Process W translates each word in the English phrase into the Indonesian target words “Kemana akan itu merah mobil pergi?”. Process X handles the allignment of the translated words “merah mobil” using the first group rules (see Sub Section 3.4.1) to get the correct phrase “mobil merah”. Process Y handles the allignment of the translated words “itu merah mobil” using the second group rules (see Sub Section 3.4.2) to obtain the correct phrase “mobil merah”. Subsequently, process Z rearranges the position of the target words “Kemana”, “akan”, “pergi”, the question mark, and the phrase “mobil merah itu” to compose the grammatical Indonesian phrase “Kemana mobil merah itu akan pergi?”. Thus, the IF-THEN statement of the English phrase mapping is implemented as follows.

```
0. IF (ADJ.$W_i$.right_connect.Contains(Q) & ADJ.$W_{i+1}$.left_connect.Contains(Q) &
       ADJ.$W_{i+1}$.right_connect.Contains(I, SIs))
1. THEN
2. {   first_buffer ← ADJ.$W_{i+1}$’;
3.     IF (ADJ.$W_{i+4}$.left_connect.Contains(D))
4.     THEN  second_buffer ← second_group_rules(ADJ.$W_{i+2}$, ADJ.$W_{i+3}$, ADJ.$W_{i+4}$);
5.     swap between first_buffer and second_buffer;  }
```

Line 0 identifies the interrogative word and the modal auxiliary. If it is true, line 2 stores the modal auxiliary (“will”) into the first buffer. Line 3 identifies the noun phrase by checking whether the fifth English word ($ADJ.W_{i+4}$) has a disjunct which contains *D* left connector. If this condition is also true, line 4 applies the second group rules to the phrase that consists of the third, the fourth, and the fifth word ($ADJ.W_{i+2}$, $ADJ.W_{i+3}$, and $ADJ.W_{i+4}$). This second group rule determines that the phrase of “the red car” is classified as a phrase which starts with a determiner “the”. The second group rules yields a correct Indonesian noun phrase “mobil merah itu”. Line 5 swaps the positions of the modal auxiliary (“will”) stored in the first buffer and the Indonesian phrase (“mobil merah itu”) stored in the second buffer to obtain the grammatical Indonesian phrase “Kemana mobil merah itu akan pergi?”.

CHAPTER 4

EXPERIMENTAL SETUP

This chapter discusses the experimental set up of the research. Firstly, data collection methods are explained. These data includes dataset used in the transfer rules development, testing dataset, and translation results. Subsequently, tools employed during the system development including two dictionaries and a parser utilized by the develop MT system are highlighted. Finally, evaluation methods are described here.

4.1 Data Collection

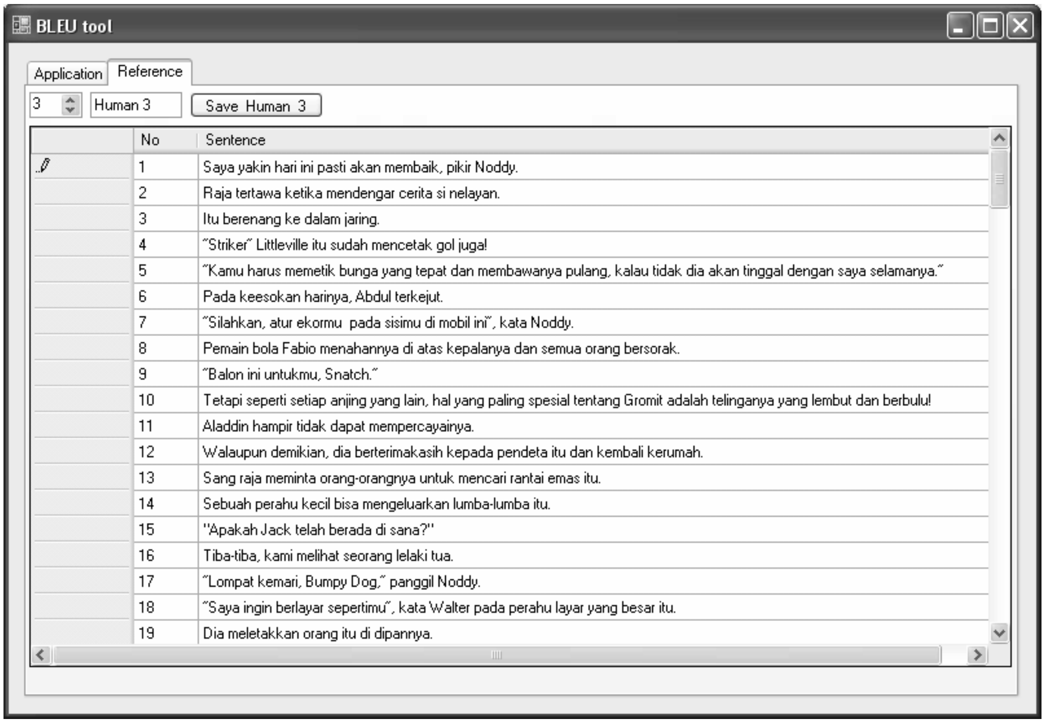
In this research, an English-Indonesian MT system is developed to prove that the proposed method i.e. the ADJ-based method for MT system is one of the appropriate methods for translating from a language with well-defined grammar formalism to a language with no grammar formalism. The ADJ-based method is a hybrid transfer method which requires bilingual or parallel English-Indonesian texts. These parallel texts are used for the transfer rules development (see Sub Section 4.1.1), for testing (see Sub Section 4.1.2), and for evaluation and comparison (see Sub Section 4.1.3).

4.1.1 Dataset Used for Transfer Rules Development

The dataset used for the transfer rules generation contains 300 sentences. Firstly, 150 sentences of them were randomly selected from 30 English storybooks. The titles vary such as Peter Pan, Snow White, Robin Hood, Lion King, and The Incredible. Some of the storybooks were borrowed from UTP library, some were borrowed from colleagues and friends, and the rest were bought from bookstores. It is assumed that readers of these books are elementary students and junior high school students. The sentences from the storybooks were translated manually into Indonesian sentences. Secondly, another 150 sentences were selected from two English grammar books [18], [83]; and translated manually. The dataset from the English grammar books was taken to generate more transfer rules in order to improve the system performance. The translation of all 300 sentences was done by considering the Indonesian grammar, which was explained by Dwipayana [35], Keraf [62], Widyamartaya [123], and Wilujeng [124]. Based on the grammatical relationship of each bilingual sentence pair, transfer rules are then developed. The 30 examples of the dataset used for the transfer rules development are given in Table B.1.

4.1.2 Testing Dataset Used for Evaluation and Comparison

A testing dataset consists of two parts: testing sentences in the *SL* and multiple human reference translations in the target language. To have enough coverage in the *SL*, a testing dataset usually has hundreds of sentences as also used in Papineni et al. [89]. In order to cover translation variations, typically 4 or more human references are used. 150 sentences were used for evaluation and comparison of the developed system over other MT system. All of the sentences were randomly selected from the same set of English storybooks used in the transfer rules development. The sentences were then translated into Indonesian language using four reference translations. One linguist and three Indonesian native speakers were involved as translators. The linguist is a lecturer at the Faculty of Language Study, State University of Yogyakarta. Two of the Indonesian native speakers are Ph.D. students of Universiti Teknologi Petronas, Malaysia. One of the Ph.D. students lived in English primary language coutries (USA and Australia) for 8 years and has a TOEFL score of 615. The other Ph.D. student lived in Australia for 2 years and holds a 585 of TOEFL score. The last native speaker holds a Masters degree from Queensland University, Australia, and has a TOEFL score of 603. The 30 examples of the translation by the Master holder are given in Table B.2 in the column of Indonesian sentences. All the reference translations were also used as input for a BLEU tool specifically designed for this research. The user interface of the tool so-called ‘BLEU tool’ which shows one of the reference translations by the Master holder translator is shown in Figure 4.1. Human 3 in the figure means the third reference translator i.e. the Master holder translator.



3 Human 3

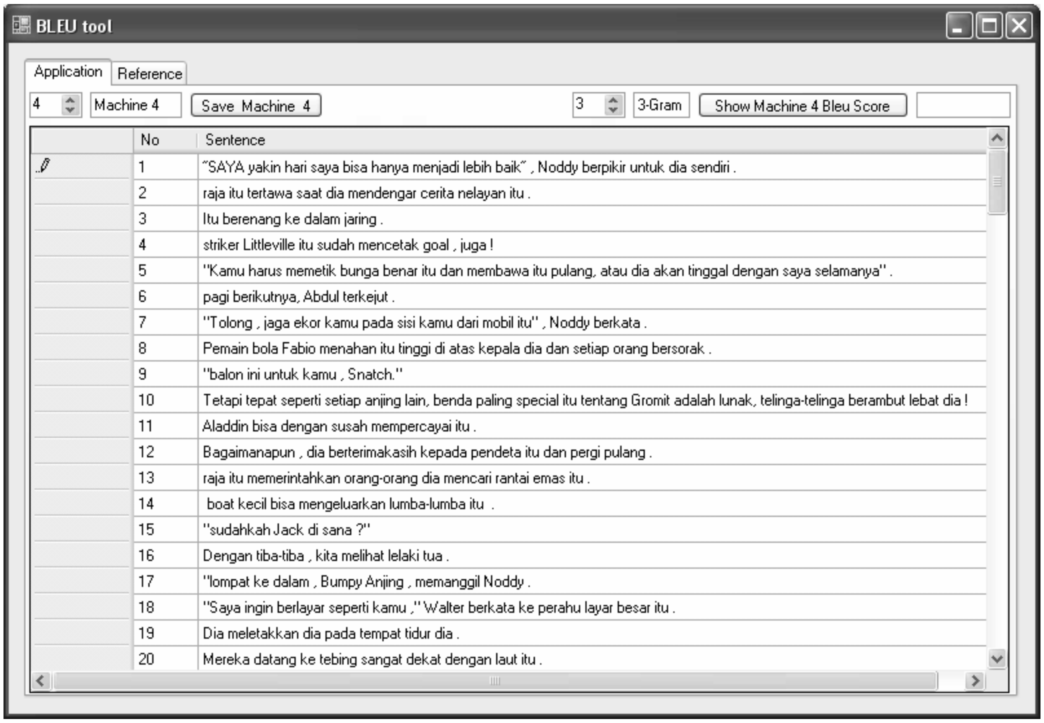
	No	Sentence
✎	1	Saya yakin hari ini pasti akan membaik, pikir Noddy.
	2	Raja tertawa ketika mendengar cerita si nelayan.
	3	Itu berenang ke dalam jaring.
	4	"Striker" Littleville itu sudah mencetak gol juga!
	5	"Kamu harus memetik bunga yang tepat dan membawanya pulang, kalau tidak dia akan tinggal dengan saya selamanya."
	6	Pada keesokan harinya, Abdul terkejut.
	7	"Silahkan, atur ekormu pada sisimu di mobil ini", kata Noddy.
	8	Pemain bola Fabio menahannya di atas kepalanya dan semua orang bersorak.
	9	"Balon ini untukmu, Snatch."
	10	Tetapi seperti setiap anjing yang lain, hal yang paling spesial tentang Gromit adalah telinganya yang lembut dan berbulu!
	11	Aladdin hampir tidak dapat mempercayainya.
	12	Walaupun demikian, dia berterimakasih kepada pendeta itu dan kembali kerumah.
	13	Sang raja meminta orang-orangnya untuk mencari rantai emas itu.
	14	Sebuah perahu kecil bisa mengeluarkan lumba-lumba itu.
	15	"Apakah Jack telah berada di sana?"
	16	Tiba-tiba, kami melihat seorang lelaki tua.
	17	"Lompat kemari, Bumpy Dog," panggil Noddy.
	18	"Saya ingin berlayar sepertimu", kata Walter pada perahu layar yang besar itu.
	19	Dia meletakkan orang itu di dipannya.

Figure 4.1: A reference translation of the third reference translator

4.1.3 Translations by Humans and MTs for Evaluation and Comparison

All the testing dataset details given in Sub Section 4.1.2 were used as inputs to test the developed system (see Figure 4.2). The same set of input was also tested on other systems i.e. Translator XP, Rekso Translator, Kataku^{TM}, and Google Translate. These dataset were used during the evaluation and comparison in Subjective Sentence Error Rate (SSER) and BLEU metrics.

In SSER evaluation, three linguists were assigned as human judges. Each judge was given the testing dataset and its translations by four systems: 1) Translator XP, 2) Rekso Translator, 3) Kataku^{TM}, and 4) the developed system. Note that Google Translate Beta version for English-Indonesian translation was not launched yet when SSER test was performed.



No	Sentence
1	"SAYA yakin hari saya bisa hanya menjadi lebih baik" , Noddy berpikir untuk dia sendiri .
2	raja itu tertawa saat dia mendengar cerita nelayan itu .
3	Itu berenang ke dalam jaring .
4	striker Littleville itu sudah mencetak goal , juga !
5	"Kamu harus memetik bunga benar itu dan membawa itu pulang, atau dia akan tinggal dengan saya selamanya" .
6	pagi berikutnya, Abdul terkejut .
7	"Tolong , jaga ekor kamu pada sisi kamu dari mobil itu" , Noddy berkata .
8	Pemain bola Fabio menahan itu tinggi di atas kepala dia dan setiap orang bersorak .
9	"balon ini untuk kamu , Snatch."
10	Tetapi tepat seperti setiap anjing lain, benda paling special itu tentang Gromit adalah lunak, telinga-telinga berambut lebat dia !
11	Aladdin bisa dengan susah mempercayai itu .
12	Bagaimanapun , dia berterimakasih kepada pendeta itu dan pergi pulang .
13	raja itu memerintahkan orang-orang dia mencari rantai emas itu .
14	boat kecil bisa mengeluarkan lumba-lumba itu .
15	"sudahkah Jack di sana ?"
16	Dengan tiba-tiba , kita melihat lelaki tua .
17	"lompat ke dalam , Bumpy Anjing , memanggil Noddy .
18	"Saya ingin berlayar seperti kamu ," Walter berkata ke perahu layar besar itu .
19	Dia meletakkan dia pada tempat tidur dia .
20	Mereka datang ke tebing sangat dekat dengan laut itu .

Figure 4.2: Translation results of the developed system

The table for listing the English sentences and their translations in Indonesian language are similar to the table in Table C.1. The different is in P3 rows, in which translation results of Google Translate Beta version were not filled in. Instead, translation results of Translator XP took place in P3 rows. Score with a range of 0 - 10 was given by the judge to each output, based on the deviation from the judge translation.

In the evaluation using BLEU metric, the testing dataset were tested to four systems i.e. Rekso Translator, Kataku^{TM}, Google Translate Beta version, and three version of the developed system. Translator XP system was not examined since its accuracy was the lowest compared to other systems according to the previous SSER test. Note that SSER test alone can be used for any MT system evaluation, as reported by Shaalan et al. [107]. This made Translator XP system was omitted for further testing. The data set examples and their translations are tabulated in Table C.1. The Indonesian sentences in this table were then used as input for the 'BLEU tool'. It can be seen from Figure 4.2 that 'BLEU tool' application shows the Indonesian sentences as translation results of Application 4 (or Machine 4). Application/Machine 4 refers to P4 i.e. the developed system (see Table C.1).

4.2 Tools

The developed hybrid transfer-based MT system needs dictionaries and a parser. The dictionaries are disjunct dictionary (see Sub Section 4.2.1) and annotated dictionary (see Sub Section 4.2.2). The parser is Link Parser [47] and discussed in Sub Section 4.2.3.

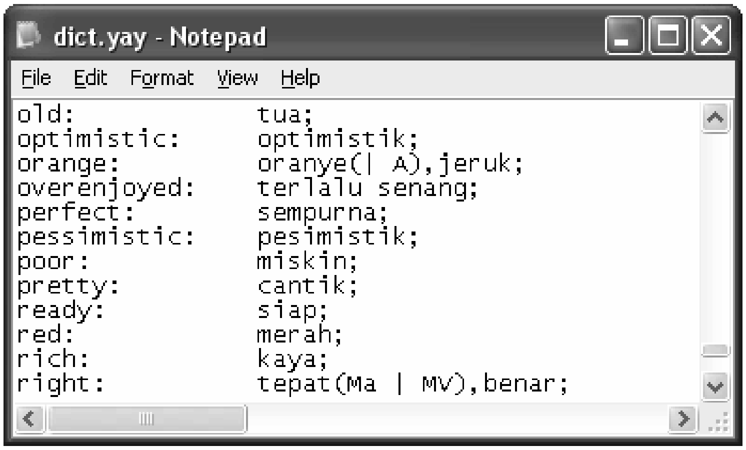
4.2.1 Disjunct Dictionary

The disjunct dictionary is a dictionary developed for the Link Parser. The dictionary consists of many files organized into two directories, namely ‘data’ and ‘words’ directories. These two directories must be placed in ‘L-Rapps-8\bin\Debug\’ directory. ‘L-Rapps-8’ is the name of the main directory where the developed MT project is located. The ‘data’ directory consists of 13 files developed by the Link Grammar authors such as ‘4.0.dict’, ‘4.1.dict’, and ‘tiny.dict’. ‘4.0.dict’ file lists the word disjuncts while ‘4.1.dict’ and ‘tiny.dict’ files are the tiny version of the word disjuncts as used in Sleator and Temperley [109]. These tiny version files are not used during the run time of the developed MT software. Each word disjunct is matched with the appropriate word category. Each word category and its members are recorded in several files e.g. ‘words.adj.1’ file, located in ‘words’ directory.

The ‘words’ directory consists of 50 files developed by Link Grammar authors such as ‘words.adj.1’ file which lists a set of adjectives, ‘words.adj.2’ file that lists another set of adjectives, ‘words.adv.1’ listing a set of adverbs, ‘words.n.1’ which records a set of nouns, ‘words.v.1’ that records a set of verbs, and ‘words.y’ recording years. These file names are written in ‘4.0.dict’ as a reference for the word disjuncts in ‘4.0.dict’ file to their matching words. These 50 files are available at http://www.link.cs.cmu.edu/link/.

4.2.2 Annotated Dictionary

The annotated dictionary file name is ‘dict.yay’. It is placed in ‘L-Rapps-8\bin\Debug\data’ directory. This file is built to list English lexicons, their translations in Indonesian, and their annotated disjuncts (for ADJ-based MT system) or their annotated connectors (for phrasal and hierarchical phrasal ADJ-based MT system). The reader should go back to Sub Section 2.1.3 for the explanation of disjuncts and connectors. These three elements are utilized for composing the ADJ set which in turn is used by the transfer rules to obtain the target sentences. The partial view of the annotated dictionary can be seen in Figure 4.3.



```
old:            tua;
optimistic:     optimistik;
orange:         oranye(| A),jeruk;
overenjoyed:    terlalu senang;
perfect:        sempurna;
pessimistic:    pesimistik;
poor:           miskin;
pretty:         cantik;
ready:          siap;
red:            merah;
rich:           kaya;
right:          tepat(Ma | MV),benar;
```

Figure 4.3: A partial view of the annotated dictionary

The third record from the top of the screen on Figure 4.3 explains an ADJ set of {(*orange*, *orange*, (| *A*)), (*orange*, *jeruk*, ( ))}. This means that the English word “orange” is translated into the Indonesian word “oranye” if *A* right connector is identified and translated into “jeruk” for any other connectors. The screen also shows the last record in the bottom describing another ADJ set of {(*right*, *tepat*, (*Ma* | *MV*)), (*right*, *benar*, ( ))}. It explains that the word “right” is translated into “tepat” if *Ma* left connector and *MV* right connector are identified and translated into “benar” for any other connectors. Note that “right” can also be interpreted as “kanan”, unfortunately the disjunct is not unique and cannot be used to distinguish the translation. The annotation process for building this annotated dictionary is already explained in Sub Sections 3.2.2 and 3.2.3.

4.2.3 Parser

The parser used here is Link Parser, an English parser developed for LG formalism [47]. The reasons for choosing Link Parser is because of its broad coverage in English grammar and lexicons [104] and its availability/openness.

The Link Parser was modified so that if an input English sentence is given then only the first linkage of the sentence is generated. The modified parser then extracts the linkage into words and their word disjuncts, which are assigned as the output of the parser. The modification of the parser was done using MinGW Developer Studio, an ANSI C compiler suitable for the parser written in C language. The modified user interface of the parser can be seen in Figure 4.4. Firstly, the screen shows that the parser module is opening '4.0.dict', 'words.n.p', 'words.n.1' files respectively until '4.0.affix' file. This is done for storing information about words, their categories, and their disjuncts into computer memories to which the parser can access.

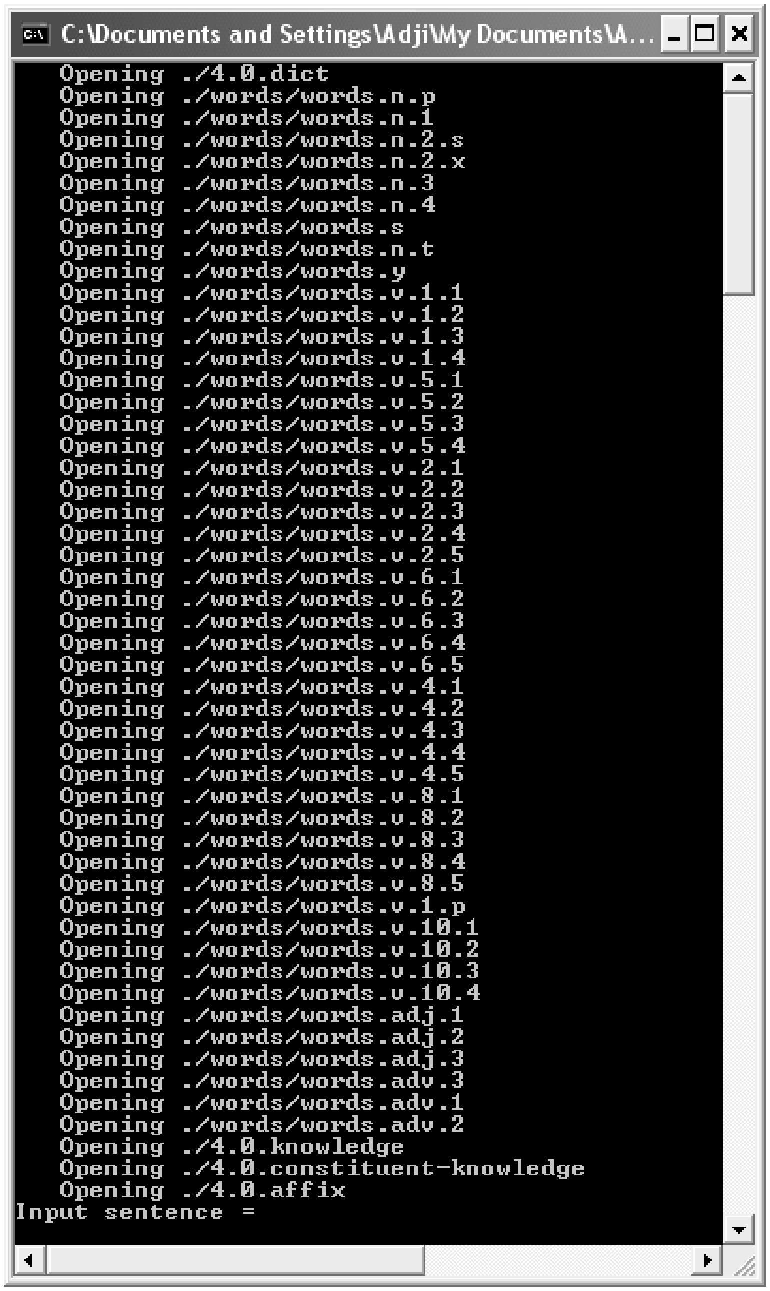
Secondly, the parser asks the user to type an input sentence. If an input sentence "She saw the red saw" is given then the user must press 'enter' key to show the output, which is captured and can be seen in Figure 4.5.

Finally, the parser shows four styles of outputs as seen in Figure 4.5 from up to down: list of words and their disjuncts, the linkage of the input sentence, list of words with word categories information e.g. "red.a" (.a explains "red" is an adjective), and 'check-4' variable comprising word disjuncts. 'check-4' variable consists of five disjuncts as follows:

Wd | Ss is read as ((*Wd*)(*Ss*)) disjunct for the word "She",
S | O is read as ((*S*)(*O*)) disjunct for the word "saw",
| D is read as (( )(*D*)) disjunct for the word "the",
| A is read as (( )(*A*)) disjunct for the word "red",
Os Ds A | is read as ((*Os*, *Ds*, *A*)( )) disjunct for the word "saw".

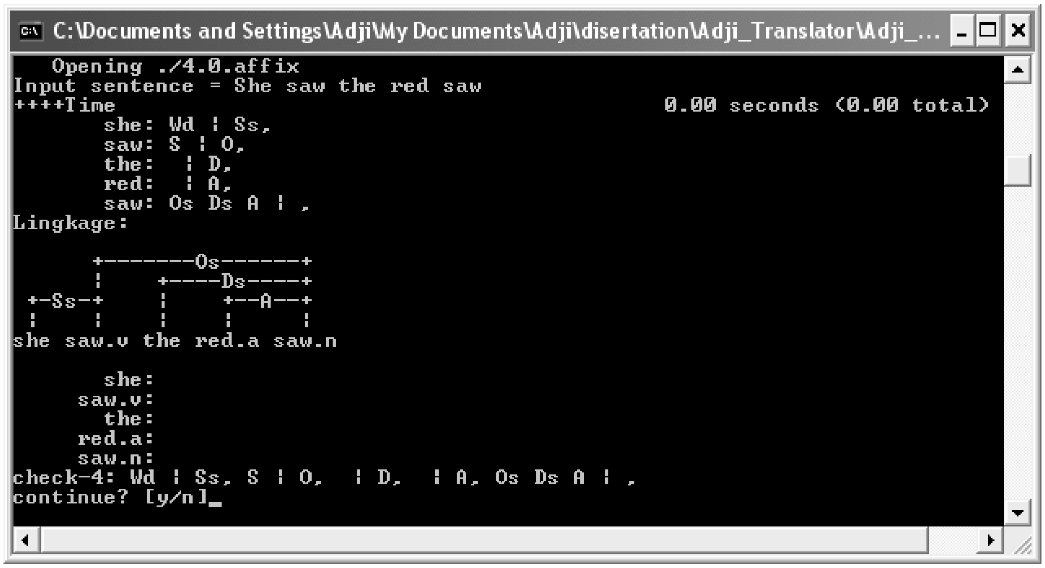
The modified parser was compiled into a dynamic linking library (dll) file. This enables the developed MT software written in C# to invoke the parser. The compiled file name is 'Adji_Trans.dll' and takes place in 'L-Rapps-8\bin\Debug\' directory.

The output of the file is only the ‘check-4’ variable i.e. the word disjuncts.



```
   Opening ./4.0.dict
   Opening ./words/words.n.p
   Opening ./words/words.n.1
   Opening ./words/words.n.2.s
   Opening ./words/words.n.2.x
   Opening ./words/words.n.3
   Opening ./words/words.n.4
   Opening ./words/words.s
   Opening ./words/words.n.t
   Opening ./words/words.y
   Opening ./words/words.v.1.1
   Opening ./words/words.v.1.2
   Opening ./words/words.v.1.3
   Opening ./words/words.v.1.4
   Opening ./words/words.v.5.1
   Opening ./words/words.v.5.2
   Opening ./words/words.v.5.3
   Opening ./words/words.v.5.4
   Opening ./words/words.v.2.1
   Opening ./words/words.v.2.2
   Opening ./words/words.v.2.3
   Opening ./words/words.v.2.4
   Opening ./words/words.v.2.5
   Opening ./words/words.v.6.1
   Opening ./words/words.v.6.2
   Opening ./words/words.v.6.3
   Opening ./words/words.v.6.4
   Opening ./words/words.v.6.5
   Opening ./words/words.v.4.1
   Opening ./words/words.v.4.2
   Opening ./words/words.v.4.3
   Opening ./words/words.v.4.4
   Opening ./words/words.v.4.5
   Opening ./words/words.v.8.1
   Opening ./words/words.v.8.2
   Opening ./words/words.v.8.3
   Opening ./words/words.v.8.4
   Opening ./words/words.v.8.5
   Opening ./words/words.v.1.p
   Opening ./words/words.v.10.1
   Opening ./words/words.v.10.2
   Opening ./words/words.v.10.3
   Opening ./words/words.v.10.4
   Opening ./words/words.adj.1
   Opening ./words/words.adj.2
   Opening ./words/words.adj.3
   Opening ./words/words.adv.3
   Opening ./words/words.adv.1
   Opening ./words/words.adv.2
   Opening ./4.0.knowledge
   Opening ./4.0.constituent-knowledge
   Opening ./4.0.affix
Input sentence =
```

Figure 4.4: User interface of the modified Link Parser


C:\Documents and Settings\Adji\My Documents\Adji\disertation\Adji_Translator\Adji_...

```
   Opening ./4.0.affix
Input sentence = She saw the red saw
++++Time                                          0.00 seconds (0.00 total)
       she: Wd ¦ Ss,
       saw: S ¦ O,
       the:  ¦ D,
       red:  ¦ A,
       saw: Os Ds A ¦ ,
Lingkage:

      +-------Os------+
      ¦    +----Ds----+
 +-Ss-+    ¦    +--A--+
 ¦    ¦    ¦    ¦     ¦
she saw.v the red.a saw.n

       she:
     saw.v:
       the:
     red.a:
     saw.n:
check-4: Wd ¦ Ss, S ¦ O,  ¦ D,  ¦ A, Os Ds A ¦ ,
continue? [y/n]
```

Figure 4.5: Output view of the modified Link Parser

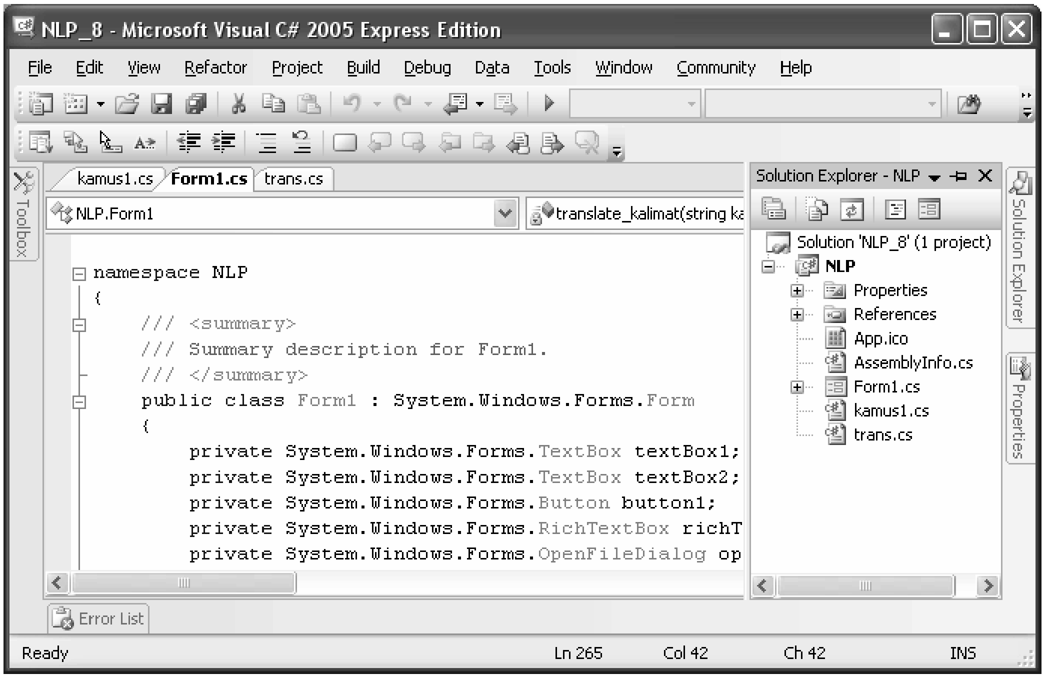
4.3 ADJ-Based MT System Files

The ADJ-based MT system is written in C# and requires the following files:

1) Disjunct Dictionary files,
2) Adji_Trans.dll (modified Link Parser),
3) dict.yay (annotated dictionary).

Name of ‘NLP_8’ is given to the solution for the system in the C# user interface (see Figure 4.6). The number 8 here means that the solution has been modified eight times. The most important files of the solution are as the followings:

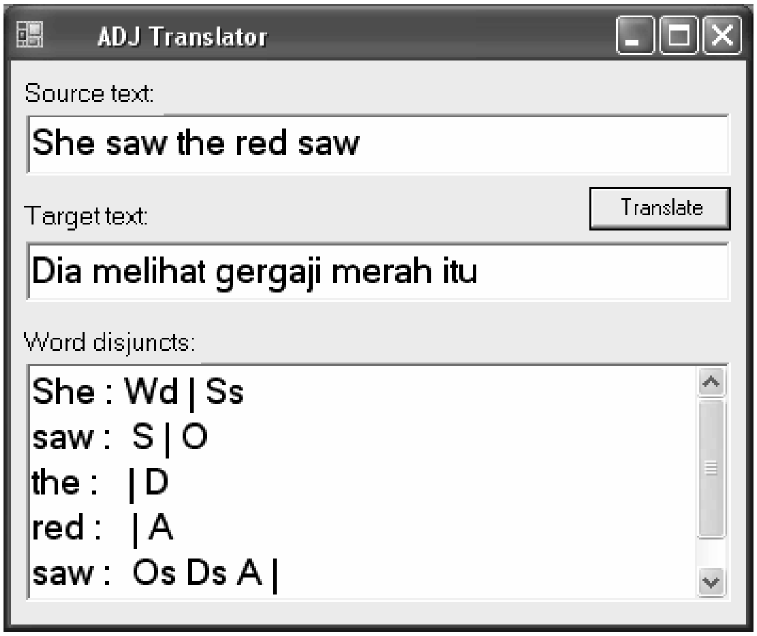
- Form1.cs,
- kamus.cs,
- trans.cs.



```
namespace NLP
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.TextBox textBox2;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.RichTextBox richT
        private System.Windows.Forms.OpenFileDialog op
```

Figure 4.6: C# user interface displaying ‘NLP_8’ solution

Form1.cs has a main job to create ‘ADJ Translator’ as the GUI of the developed MT system (see Figure 4.7). The GUI was used to run all the data for developing transfer rules (300 English sentences) and all the testing data (150 English sentences). Users can type an English input text in ‘Source text’ text box. Pressing ‘Translate’ button will trigger a function in ‘Form1.cs’ to call ‘Adji_Trans.dll’ to accept the input text and to generate all word disjuncts shown in ‘Word disjuncts’ text box. This text box helps for further analysis and for editing the annotation of the ADJ set (by any well-trained end user). If the input text consists of more than one sentence then the input text is divided into many sentences identified by several punctuation marks such as fullstop. Each sentence is fed into ‘Adji_Trans.dll’ one by one. Put differently, if the input consists of *n* sentences then ‘Adji_Trans.dll’ is called for *n* times. The result of the translation appears in the ‘Target text’ text box. ‘kamus.cs’ uses ‘dict.yay’ (annotated dictionary) to create on-the-fly look up table with two columns. The first column lists all source words and the second column lists the target words and their annotated disjuncts. ‘trans.cs’ has two main tasks. The first task is finding the target words of the given source words which appear in the ‘Source text’ text box. This is done by performing queries in the look up table the matching source words based on the annotated disjuncts. The second task is to execute transfer rules for repositioning each source word into its correct position in the target text. The transfer rules are already expained in detail in Sections 3.3 and 3.4.



Source text: She saw the red saw

Target text: Dia melihat gergaji merah itu

Word disjuncts:

She : Wd | Ss
saw : S | O
the : | D
red : | A
saw : Os Ds A |

Figure 4.7: ‘ADJ Translator’ GUI

4.4 Evaluation and Comparison Methods

The evaluation and comparison of the developed system is done by using human evaluation and automatic MT evaluation. Some automatic MT evaluation systems had been reported to achieve the improvement of correlation to human evaluation score [74], [91]. However, natural languages are rich and ambiguous which allow many possible different ways of interpreting [14] and translating them. In this sense, human evaluation on MT measures many aspects of translation including adequacy, fidelity, and fluency [50], [22]. The problem is that the evaluation approaches are quite expensive [50]. Moreover, they may take weeks or months and this is not good for MT developers since they need to monitor the effect of small changes to the MT systems for daily analysis and improvement [89]. Based on the advantages and disadvantages of both human and automatic evaluation, both evaluations are thus conducted for the developed system and explained separately in Sub Sections 4.4.1 and 4.4.2, respectively.

4.4.1 Evaluation and Comparison using Subjective Sentence Error Rate

Subjective Sentence Error Rate (SSER) metric is widely used for human evaluation on MT systems. Currently, the developed MT system has only a limited dictionary of 3000 pairs of common English-Indonesian words. So far, only one appropriate human evaluation method was found to be applicable in this work. The evaluation was for English-Arabian noun phrase translation tasks, which was discussed in Shaalan et al. [107]. In this evaluation method, 156 simple English noun phrases found in 50 selected thesis titles from the computer science domain are used for the evaluation. This method was customized for this research to allow evaluation and comparison of English-Indonesian MT systems in performing sentence-based translations (see Figure 4.8). The steps are as follows.

1. 30 English story books for elementary school students were randomly collected.
2. From these 30 books, 150 sentences were randomly selected. The sentences were manually translated into Indonesian language. Based on the English to Indonesian mapping patterns, the transfer rules were built and fed to the system.
3. Other 150 sentences taken from two English grammar books [18], [83]; were also collected and translated to generate more transfer rules to achieve better performance.
4. The developed transfer rules were tested by using unseen 150 different sentences, which were randomly selected from the same set of books. The same sentences were also used as input for other English-Indonesian MT systems.
5. The translation results generated by all the systems were sent to three linguists. The linguists assigned score (0 - 10) for each output based on the deviation from the linguist translations.
6. Overall score were finally computed using SSER metric, which gives the accuracy of the tested systems in percentage.

The accuracy of each MT was measured using SSER metric [80] which is given by Equation (4.1).

$$\mathrm{SSER}(s_1^n, t_1^n)\ [\%] = 100 - \frac{10}{n}\sum_{i=1}^{n} v(s_\mathrm{i}, t_\mathrm{i}) \qquad (4.1)$$

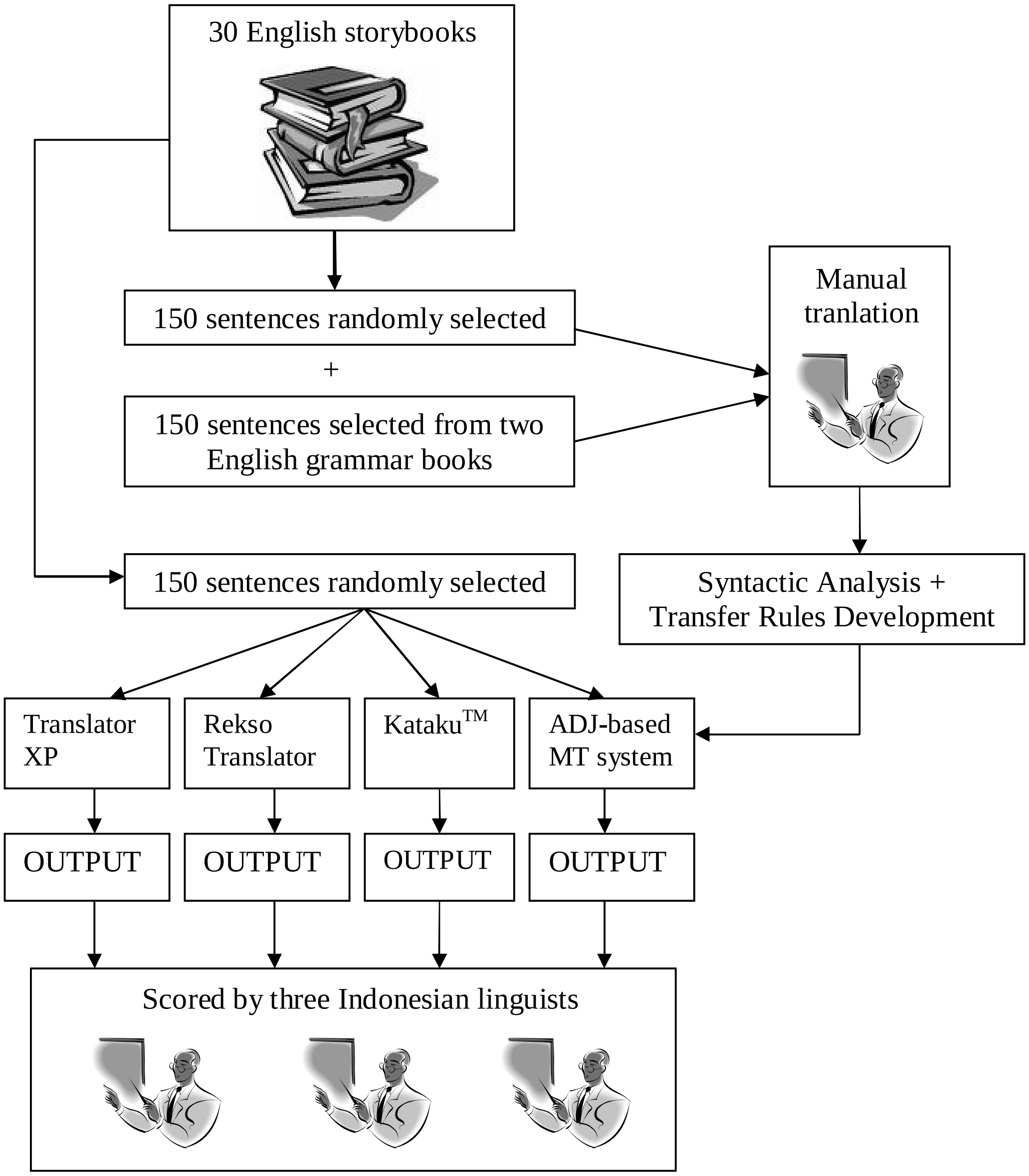

30 English storybooks
150 sentences randomly selected
+
150 sentences selected from two English grammar books
Manual tranlation
150 sentences randomly selected
Syntactic Analysis + Transfer Rules Development
Translator XP
Rekso Translator
Kataku™
ADJ-based MT system
OUTPUT
OUTPUT
OUTPUT
OUTPUT
Scored by three Indonesian linguists

Figure 4.8: SSER evaluation method for this research

In Equation (4.1), n is the total number of sentences, $t_1^n = t_1 \ldots t_n$ is a set of translations, $s_1^n = s_1 \ldots s_n$ is a set of test corpus, and $v(s_i, t_i)$ is the value/score for the ith sentence. If there are m human judges then the accuracy is given by Equation (4.2).

$$\text{Accuracy [\%]} = 100 - \frac{1}{m} \sum_{j=1}^{m} \text{SSER}_j \tag{4.2}$$

In this research, $n = 150$ and $m = 3$.

4.4.2 Evaluation and Comparison using BLEU metric

BLEU metric is one of the widely used MT evaluation metrics, besides NIST [34], [81]; Modified-BLEU [6], [132]; and METEOR [6]. BLEU metric is used by the IBM Statistical Machine Translation group [89] and used as evaluation metric in several MT reports [26], [32], [40], [88], [132]. The aim of the BLEU tool is to evaluate the system precision objectively. The BLEU metric is defined by Papineni et al. [89] as Equation (4.3).

$$\text{BLEU} = \text{BP} \bullet \exp \left(\sum_{n=1}^{N} w_n \log p_n \right) \tag{4.3}$$

- $BP = \begin{cases} 1 & if \quad c > r \\ e^{(1-r/c)} & if \quad c \leq r \end{cases}$, is the Brevity Penalty used to penalize candidates length (c) shorter than their reference translations length (r),
- c = the MT candidates/hypothesis/results length,
- r = effective reference translations length,
- $w_n = 1/N$, is uniform weight where usually $N = 4$ such as used in Papineni et al. [89] and Zhang and Vogel [132],

- N = maximum length of n-gram,
- $p_{\mathrm{n}} = \dfrac{\sum_{C\in\{Candidates\}} \sum_{n-gram\in C} Count_{clip}(n-gram)}{\sum_{C'\in\{Candidates\}} \sum_{n-gram'\in C'} Count(n-gram')}$, is modified n-gram precission.

To compute p_{n}, one first counts the maximum number of times an n-gram occurs in any single reference translation. Next, one clips the total count of each candidate (C) n-gram by its maximum reference count, adds these clipped counts up, and divides by the total (unclipped) number of candidate words.

Before using the BLEU metric in Equation (4.3), the followings situation has to be decided:

1) the amount of testing dataset used for statistical automatic evaluation of the MT systems,
2) the number of reference translations used for the testing dataset.

A work by Elliott et al. [39] explicitly attempts to solve the first problem by concerning human metrics fluency, adequacy and informativeness. The work focuses on the ranking of systems based on the results of the French/English, Spanish/English and Japanese/English DARPA 1994 MT evaluation campaign. The scores were compared for an increasing number of texts, starting with one and ending with 100 texts with the average length of texts being 350 words. Based on an empirical assessment of score variation, it was estimated that systems could be reliably ranked with around 40 texts (ca. 14,000 words), and that using ten texts already separate the highest and the lowest ranked systems. Zhang and Vogel [132] also studied the influence of the number of test data on the reliability of automatic evaluation metrics, focusing on confidence intervals for BLEU and NIST scores. They used the data of the Chinese/English track of the TIDES 2002 MT evaluation campaign (100 documents of 7-9 sentences each), with the output of the 7 participating systems and 4 reference translations. Their results show that BLEU and NIST scores become stable when using around 40% of the data (around 40 documents or 300 sentences). These two studies suggest that an evaluation can be reliably performed with less text than is often used [40]. This research used 150 sentences (around 1619 words) randomly collected from a random 30 English story books. This is still acceptable since the number of sentences used is more than what is used by Papineni et al. [89], which were 127 sentences.

Meanwhile, Zhang et al. [133] tries to clarify the second problem after an evaluation of several MT systems using NIST, BLEU, and Modified-BLEU metrics by stating a rough rule of thumb that doubling the testing data size narrows the confidence interval by 30%. The other result is that the relative confidence interval becomes narrower with more reference translations. In other words, increasing the testing data size as well as using more reference translations increases the precision of the evaluation metrics, i.e. narrows down the confidence interval. It was found during the observation that 100% testing data with 1 reference is equivalent to 80~90% of testing data with 2 references, or 70~80% of testing data with 3 references, or 60~70% of testing data with four references. That is to say, adding an additional reference translation will compensate the effects of removing 10~15% of the testing data on the relative confidence interval. Therefore, it seems more cost effective to have more test sentences but fewer reference translations.

It can be summarized based on these observation results into two things:

1) less number of testing dataset can be used with more number of reference translations,
2) oppositely, less number of reference translations can be used with more number of testing dataset.

In this research, less number of testing dataset but acceptable (i.e. greater or equal than 127 sentences as used by Papineni et al. [89]) with four reference translations as also employed in several works [26], [32], [40], [88], [132]; was used in this research. Four reference translations were used since they were available without cost.

For this automatic evaluation, an MT evaluation tool using BLEU metric was developed in C#. The solution in the C# user interface is called ‘BLEU-Grid’ solution (see Figure 4.9) and has the following files:

- Form1.cs with its design so-called Form1.cs[Design],
- Machine.cs,
- Ref.cs,
- Gram.cs.

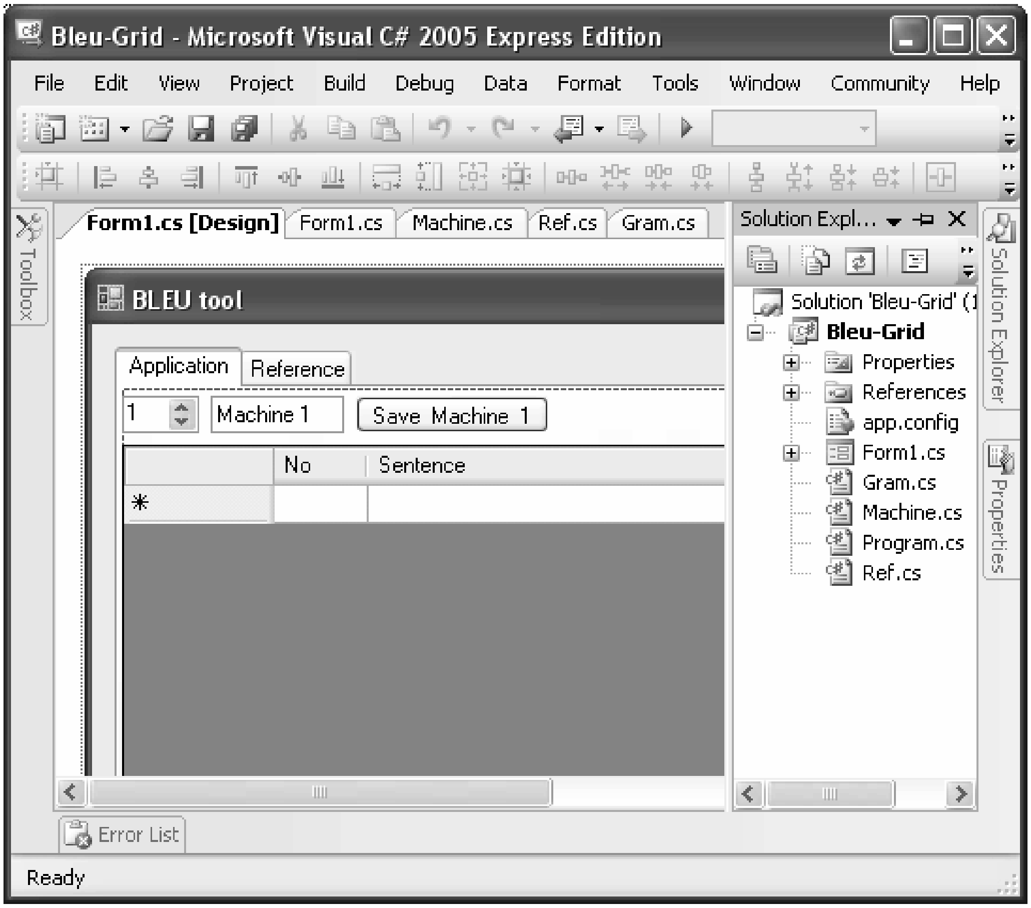


Figure 4.9: ‘BLEU-Grid’ solution

‘Form1.cs’ has a main job to create ‘BLEU tool’ as the GUI of the developed automatic MT evaluation tool (see Figure 4.2). It has two tabs, i.e. ‘Application’ and ‘Reference’. The ‘Application’ tab is set as the default initial screen which is related with all information about the MTs, which are defined by class ‘Machine’ in ‘Machine.cs’ file. Users can select an MT system from Machine 1 to Machine *N*, where *N* is the total number of MT systems to be tested. The selection is done by selecting an available number in the numeric up down box located below the ‘Application’ tab on the most left. If a user selects number 4 then “Machine 4” (showing the number of the MT system being tested) appears in the text box on the right of the up down box as well as the appearance of ‘Save Machine 4’ button, which is used to save the translation results of the MT system inserted by user in the data grid view located in the bottom of the screen. The other up down box located on the right is for selecting the maximum *n*-gram order of the BLEU metric, limited to 3 until 5 only as suggested by Papineni et al. [89]. If number 3 is selected then “3-gram” comes into view in the text box on the right. At the same time, ‘Show Machine 4 Bleu Score’ button becomes visible. User must click this button to see the BLEU score of the respected machine with a desired maximum *n*-gram order. ‘Gram.cs’ file defines the computation of the BLEU score.

‘Ref.cs’ file defines class ‘Ref’ used to initialize and define reference translations. In ‘BLEU tool’ user interface, if user click ‘Reference’ tab then the appearance of the BLEU tool is shown in Fig 4.1. This particular screen shows all information about the reference translations. Clicking the up down box on the most left will let the user to choose the translation of the source sentences by a particular human reference i.e. Human 1, Human 2, Human 3, and Human 4. For example, if user select number 3 then ‘Human 3’ comes to visible in the text box on the right. Simultaneously, ‘Save Human 3’ button appears to let the user save the translation by Human 3 shown in the data grid view in the bottom. The data grid view can also be used to insert and update the translation.

The method for evaluation using BLEU metric can be summarized as the following steps.

1. 30 English story books for elementary school students were randomly collected.
2. From these 30 books, 150 sentences were randomly selected. The sentences were manually translated into Indonesian language. Based on the English to Indonesian mapping patterns, the transfer rules were built and fed to the system.

3. Other 150 sentences taken from two English grammar books [18], [83]; were also collected and translated to generate more transfer rules to achieve better performance.
4. Three different version of the developed MT system, namely ADJ-based sentence translation MT system, ADJ-based phrase translation MT system, and ADJ-based hierarchical phrase translation MT system, were tested by using unseen 150 different sentences, which were randomly selected from the same set of books. The same sentences were also used as input for other English-Indonesian MT systems.
5. The translation results generated by all the systems were used as input for 'BLEU tool' interface under 'Application' tab as seen in Figure 4.2. Whilst 4 reference translations were used as input for this 'BLEU tool' interface under 'Reference' tab as shown in Figure 4.1.
6. Execution of 'BLEU tool' application to calculate the precision on phrase length of 3-, 4-, and 5-gram for each system in percentage.

Note that in this research, steps 1 until 3 was already conducted in SSER testing so that these steps can be skipped and proceeded to step 4.

CHAPTER 5

RESULTS AND DISCUSSIONS

In this chapter, evaluation and comparison process of the developed MT system is conducted. During our experiments, changes to the MT software had been made to improve its performance. The changes were not only based on the errors of the translation, but also based on the new techniques or methods obtained from other works. Three versions of the MT systems have been developed to which evaluation and comparison with other available systems is performed. The first version is the sentence-based MT system which is evaluated in Section 5.1. The second version is the phrase-based MT system and is evaluated in Section 5.2. The last version – the hierarchical phrase-based MT system – is evaluated in Section 5.3. Section 5.4 summarizes evaluation and comparison of all versions of the developed system with other available systems.

5.1 Evaluation of the Sentence-Based MT System Using Annotated Disjunct

The first evaluation and comparison was done by using an MT performance evaluation method, which was discussed by Shaalan et al. [107]. This method was customized for this research to allow evaluation and comparison of English Indonesian MT systems in performing sentence translations. The performance (in terms of accuracy) of all MTs was calculated in percentage using Equation (4.2). In this research, *m* (the total number of human judges) equals to three. The reader need to go to Sub Section 4.4.1 for the detail of the SSER evaluation set up. The tested systems for this evaluation and comparison are Translator XP, Rekso Translator, Kataku^{TM}, and the developed system (ADJ-based system). The accuracy of all tested systems is shown in Figure 5.1.

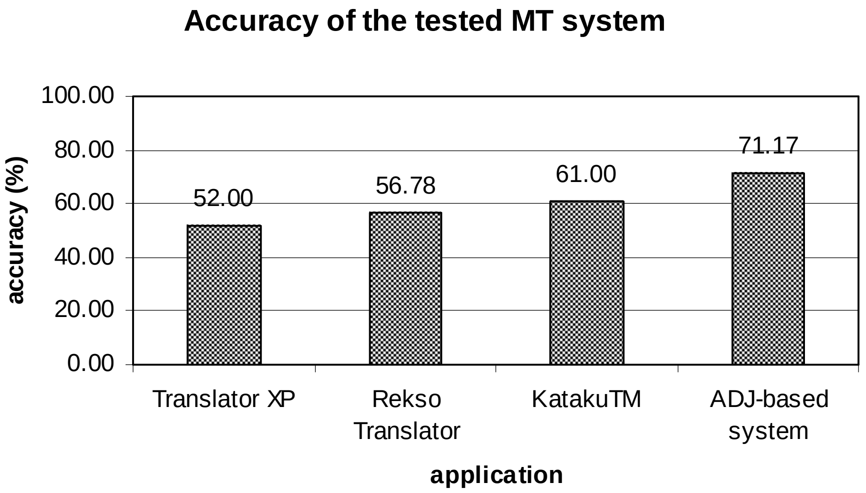


Figure 5.1: Accuracy of all the tested MT systems using SSER

Although the result of our developed system is better than the results of the other softwares, unfortunately, the error rate is still considered high (28.83%). However, it seems quite possible that there will be a significant improvement in the accuracy if the amount of training data is large [113], since the MT development only used 300 example data. In addition, the accuracy is also very much reflected by the generation of more transfer rules. Possible causes of the high error rate were analyzed based on the data in Table 5.1 as given in the following lines.

Table 5.1: Performance analysis data

Category	Linguists' Score	Total Sentences	Total Sentences (%)
I	$5 \leq$ score < 6	2	1.33
II	$6 \leq$ score < 7	55	36.67
III	$7 \leq$ score < 8	58	38.67
IV	$8 \leq$ score < 9	30	20.00
V	$9 \leq$ score $\leq$ 10	5	3.33

Scores assigned by the human judges were grouped into five categories together with the number of sentences (from the total of 150 tested sentences) fall within each category. It was found that 13.33% of the tested English sentences were not in LG formalism. Hence some of their words have no disjuncts which have made the system fail to produce correct translations. These have contributed to the lower scores given by the judges for sentences in Category I-IV. For example, all sentences in Category I involves idiomatic phrases or sayings such as “What a wonderful day!”. The system mapped “What a wonderful day!” into “Apa bagus hari !” with low average score of 5.33 from the judges. The correct translation should be “Alangkah indahnya hari ini!”. To achieve higher score for the translation of this kind of saying, proverbs, or greetings, a database of all sentences in those categories which consists of sentence-by-sentence mapping, that translates the whole *SS* into the whole *TS* as it is, is needed. Another method to obtain higher scores for those cases is to combine the ADJ-based method with the statistical-based method. Meanwhile Category V shows that the judges were happy with the translated results which provide the best level of system performance.

It was also found that Category II, III, and IV contributed to the most error i.e. 95.34% of the tested English sentences, which prompted us to further explore the causes. The sentences in those categories contained certain noun phrases (e.g. “the next morning”, “Footballer Beckham”), verb phrases with particles (e.g. “look for”, “dream of”), and translations that require morphological analysis. Further morphological studies on Indonesian language are vital since the language employs affixes with more complexity than English [4], [17]. Some of the tested sentences were in the English interrogative forms and passive forms which need morphological analysis. For example, “paid” in the passive sentence “You will be paid” was translated into the Indonesian inflectional verb “dibayar”. Meanwhile, “paid” is frequently translated into the inflectional verb “membayar” in active sentences like “I paid you”. For this example, surprisingly the ADJ Algorithm can generate different disjuncts for the word “paid” in both forms. Two opportunities exist to improve the work, i.e. adding a word stemmer and a morphological analyzer for the ADJ approach, which is likely to solve the mentioned problems.

The second evaluation and comparison was conducted by using a BLEU metric tool, which was developed in C# (see Sub Section 4.4.2). While four reference translations for each of 127 source sentences were used in Papineni et al. [89], this report used four reference translations and 150 source sentences, which were selected randomly from 30 story books. Precision of all tested systems is shown in Figure 5.2.

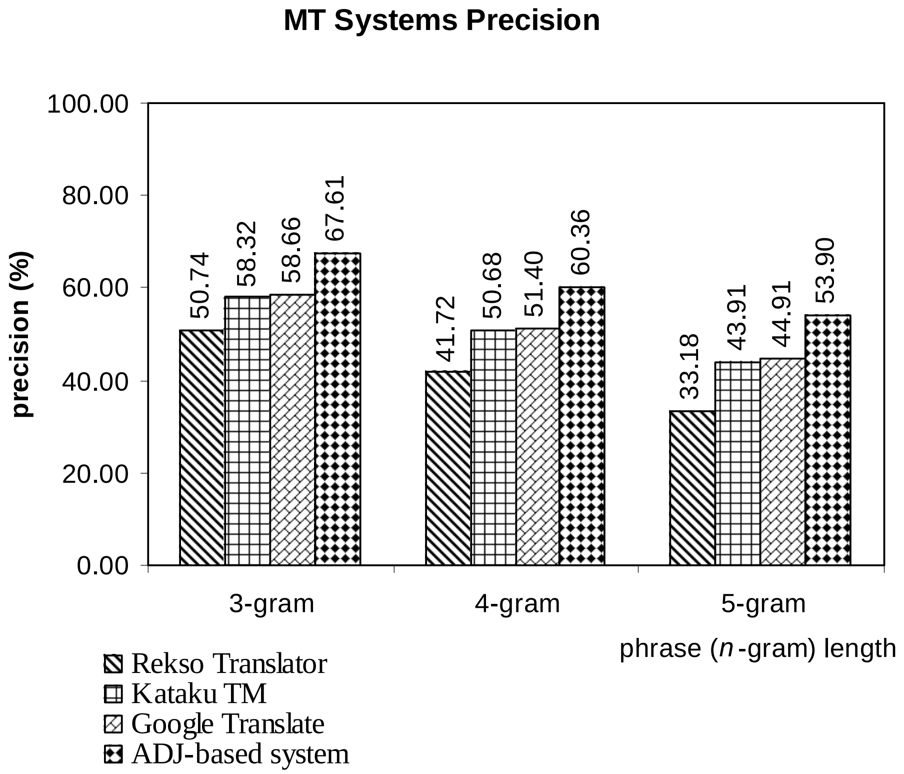


Figure 5.2: Precision of English-Indonesian MT systems using BLEU metric

It must be noted here that in this BLEU evaluation, Translator XP was not examined since its accuracy was very low according to the previous SSER test. Instead, Google Translate Beta version which is developed more recent was evaluated and has second best performance as shown in Figure 5.2.

Firstly, Rekso Translator and Kataku^{TM} were not evaluated extensively since no reports were found on the discussion of their translation methods and their precisions were arguably lower. It can be seen from Figure 5.2 that Rekso Translator shows the worst precision. This was due to the appearance of unrelated symbols such as "[", "(", and "-" (dashed) in front of some translated words. For example, "-" was appeared after the first double quote when the *SS* was in double quotes i.e. "Has Jack been there?" was translated into "-Has Jack been there?". Another cause of the lowest precision was, if there exists word disambiguation, all possible target words were displayed, separated with "/" symbol. For example, "fairy" was translated into "peri" and “dongeng”. Hence the translation result was written as “peri/dongeng”, which makes the *n*-gram precision module giving zero match score. Meanwhile, Kataku^{TM} precision were the third rank, although it could translate some phrases such as “Once upon a time”, “It’s too late”, “we could be friends”, and “We cannot fly yet” into “Pada suatu ketika”, “Terlalu terlambat”, “kami bisa menjadi teman”, and “Kami belum bisa terbang” respectively with highest fluency.

Secondly, Google Translate were evaluated more intensively since many publications discussed the approach adopted by this system. There was an interesting finding with the Google Translate. At the beginning, it produced the worst precision. It was then found that this application had a problem in translating sentences within double quotes. Since this problem was not fundamental then these double quotes were removed. After reevaluating the results, the precision rate by Google Translate improved and was ranked second in the list. It was expected that this statistical-based Google Translate performed well. However, in this research, its accuracy was still very low, due to the small bilingual corpora used by the application. In addition, there are no available good English-Indonesian corpora at present. This application had a problem in identifying some of single quotes such as in phrases/sentences “Grandmother’s bed”, “I’m sorry”, “wouldn’t”, “Don’t you?”, “Gaston’s done it.”, and “Well, if you don’t, I do!” which caused no mapping for these whole phrases/sentences to be generated. Many words like “swam”, “too”, “dolphin”, “magician”, “princes”, “Prince”, “mouse”, “Mr.”, “Grandmother”, “puppies”, “Beast”, and “Bring”, “monsters” were not translated. The corpus were also lacking of verbs in past form such as “got”, “cheered”, “rushed”, “filed”, “crept”, “wandered”, “grabbed”, “sighed”, “whistled”, “gasped”, and “whispered”. Some of the noun phrases consist of adjective nouns such as “Footballer Fabio”, “Baby Bear”, and “ape nesting” which could not be mapped into Indonesian. This application also failed to translate a single word of sentences such as “She has done it”, “What do you want?”, and “What a wonderful day!”. However, some difficult tasks on translating such as “said the second”, “He’s not one of us”, and “It is better than the other one” were accomplished with perfect matches into “berkata yang kedua”, “Dia bukan salah satu dari kami”, and “Itu lebih baik daripada yang lainnya”.

The ADJ-based system translated simple, compound, and complex English sentences in present, present continues, present perfect, past, past perfect, and future tenses with better precision than the other systems. This was typical since the generated transfer rules were mostly based on the tenses used in the example data, which involved the tenses. That is to say, the tenses found in the first 300 example data were also found in the 150 testing data. Thus, an evaluation on whether the precision will decrease should be conducted when the *SS* were in other tenses in complex or compound forms.

The system translated “What do you want?” into “Apa yang kamu ingin?”, whereas the correct one should be “Apa yang kamu inginkan?”. Thus, “want” in this interrogative sentence must be translated into an Indonesian inflectional verb “inginkan” (from the root word “ingin” (= want) with suffix “kan”). Meanwhile, “want” is frequently translated into the inflectional verb “menginginkan” (root word “ingin” with prefix “meng” and suffix “kan”) in affirmative sentence like “You want this book”. Surprisingly the ADJ Algorithm can generate different disjuncts for the word “want” in both forms. Two opportunities exist to improve the work i.e. adding a word stemmer and a morphological analyzer for the ADJ approach, which is expected to solve the mentioned problem.

In this research, the example data set has an average sentence length of 10.79 words per sentence, ranging from two to 29 words per sentence. It can be said that the ADJ-based system consistently outperformed the other applications. The precision of all systems evaluated using BLEU metric is lower in all length of *n*-gram as compared to the precision results obtained through human judgment. This is typical since a linguist can judge better than an MT evaluation tool in terms of adequacy, fidelity, and fluency of the translations. Nonetheless, BLEU metric is highly correlated with human judgment [89]. It was also found that the longer the word-length of *n*-gram, the higher reduction of the system precision is.

An interesting result was obtained from ADJ module. Many POS of source words which are indicated by the ADJ set coincidently are also valid as the same POS of the target words. For example, Figure 4.5 explains that the source word “the” is a determiner of “saw”, as indicated by *Ds* connector. There is a matching with the target words where the Indonesian word “itu” (the) is a determiner of the noun “gergaji” (saw). Furthermore, the target word “merah” (red) is an Indonesian adjective of the Indonesian noun “gergaji” (saw). The word “sangat” (very) is an adverb of the word “besar” (big). This finding provides an interesting motivation to boost the development of new reversed transfer rules from Indonesian to English.

It was also found that the annotated disjunct can distinguish the associated disjuncts of different pairs of words such as for two English-Indonesian pairs of words “saw → gergaji”, which are ((*Os*,*Ds*)( )) and ((*Ds*)(*Ss*)), when the system translated “She saw the saw. What saw is that?” (see Figure 3.6 and Figure 3.8). The first annotated disjunct consists of *Os* left connector explaining that the target word “gergaji” is an object and the second with *Ss* right connector explains that “gergaji” is a subject. These results will be beneficial for different direction of research such as NER or IE which intensively observe the POS of words and the structure of sentences.

The comparison to other proprietary softwares indicated that the ADJ method applied in our system is still promising since the data used in the transfer rules development are only 300 sentences. According to [113], the accuracy of the real MT software should be trained on a minimum of 100,000 sentences. Thus, with more training data fed to our MT system, not only the better precision will be achieved but also the more robust transfer rules are expected to be created.

5.2 Evaluation of the Phrase-Based MT System Using Annotated Disjunct

The precision of all tested systems in BLEU metric is shown in Figure 5.3. This figure shows consistent results where the phrase-based translation system precision increased slightly about 2% higher than the previous ADJ-based system precision for all *n*-gram lengths and outperformed other system precision with more than 10% different. Cases (among 150 testing data) that contribute to the increase of the accuracy (solved cases category) and cases those are still unsolved are classified into ten categories as shown in Table 5.2. A single case is defined as a sentence which contains at least a problematic phrase that is either solved (correctly translated) or not solved in this work. The solved cases mean cases which were correctly translated by the developed Phrase-Based Transfer Rules Algorithm.

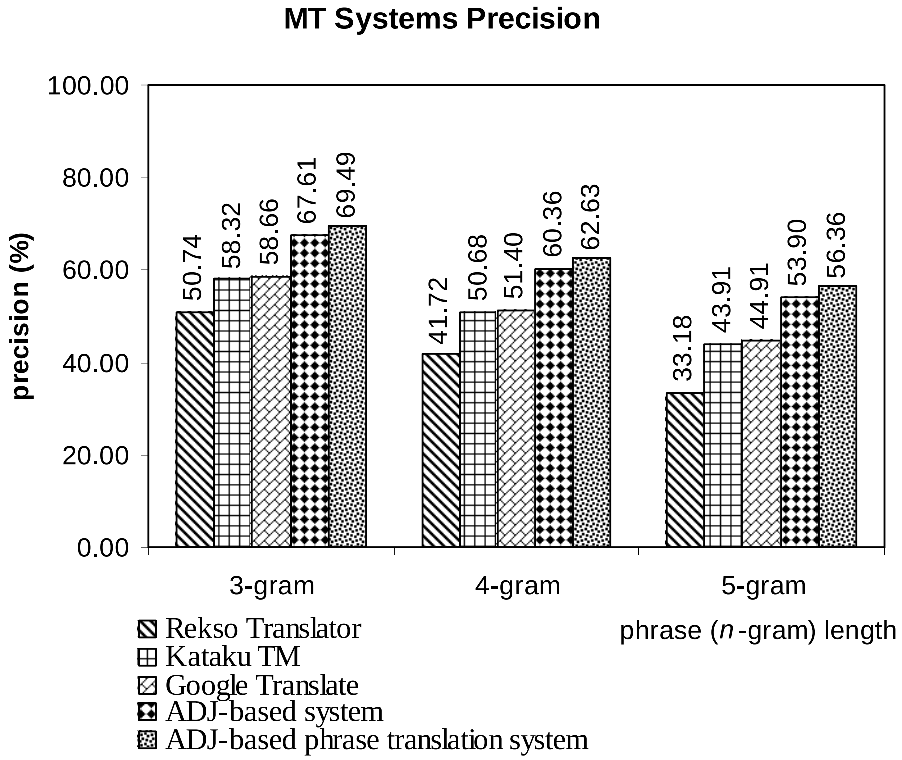


Figure 5.3: Comparison of phrase-based system with other MT systems

Table 5.2: The number of solved and unsolved cases for each phrase category

Category	English Phrase Category	Total Solved Cases (%)	Total Unsolved Cases (%)
I	Idiomatic time phrases	1.33	0.00
II	Infinitive phrases	4.67	1.33
III	“ing” form phrases	2.00	2.00
IV	Phrases with pronoun “one”	0.00	2.00
V	Phrases in interrogative sentences	0.60	2.67
VI	Possessive noun phrases	2.67	4.00
VII	Phrases in adjective clauses	0.00	4.00
VIII	Phrases in negative sentences	5.33	4.00
IX	Phrases in passive sentences	0.00	5.33
X	Other phrases	0.00	7.33
	Total	16.60	32.67

The total number of unsolved cases is about twice of the total number of solved cases. This result prompted us to explore the causes. It was found that Category X (other phrases category) contributed to the highest unsolved cases with total cases of 7.33% and no solved cases. Most of this category were phrasal verbs like “dream of” and “go in”; ambiguous phrases such as “there were three bears”, “May I have it?”, “a few years”, and “baby Jumbo looked so funny”; and sayings like “What a wonderful day”. The disjuncts generated by the Link Parser for this kind of cases could not be utilized to translate the phrases correctly. Hence, to get the correct translation of the sayings, ambiguous words, and phrasal verbs then a database of all sentences in this category which consist of phrase-by-phrase mapping, that translates the whole source phrase into the whole target phrase as it is, is needed. Another way can be a combination of the developed phrase-based method with the phrase-based statistical method. However, phrases consist of noun-modifiers such as “the waiting truck” and noun phrases with determiners such as “every other dog”, which were also in this category, have connectors that could be used to translate correctly. “waiting” has an *AN* right connector and was translated by the system into one Indonesian word “menunggu” while the correct translation is in three words “yang sedang menunggu”. In our transfer rules, the *AN* connector was utilized to solve noun-modifiers in root form of nouns only, not for noun-modifiers in “ing” form of verbs. Thus, adding a rule for “ing” form of verbs to the existing noun-modifiers transfer rule will hopefully solve the problem.

Other high percentages of unsolved cases were contributed by Category V or phrases in interrogative sentence (4.00%) like “Are you looking for Noddy?”, “May we go”, “what was Winnie doing”, and “What magic do you use?”; Category VI or possessive noun phrases (4.00%) like “the fisherman’s story”, “the lion’s cry”, “Baby Bear’s little bed”, “the handsome prince’s future”, “the Incredible family’s problems”, and “a girl’s game”; Category VII or phrases in adjective clauses (4.00%) such as “the money he had”, “Phil, who was half man”, “the moment I dreamed of”, “Princess Aurora and Prince Philip themselves, who lived happily”, “Robin Hood, who slipped away”, and “The old woman, who was really the fairy”; Category VIII or phrases in negative sentences such as “don’t stop”, “Why don’t you ask”, “Wouldn’t you choose the company”, “Don’t you?”, “will not go”, and “is not a human”; and Category IX or phrases in passive sentences like “was being dressed up”, “will be well paid”, “had been promised”, “was watched by her”, “was filled with joy”, “ought to be called – Dumbo”, “was grabbed”, and “had been seen”.

Category V, VII, and IX need to be evaluated extensively since all cases were unsolved except one case (0.60%) in Category V. These categories need morphological analysis and/or construction to be incorporated in the system. For example, a word “May” in “May we go” (Category V) must be translated into the Indonesian inflectional interrogative word “Bolehkah”. Meanwhile, “may” in a declarative sentence “You may go” is translated into “boleh”, without the suffix “kah”.

Surprisingly the ADJ Algorithm can generate different disjuncts for the word “may” in both forms. The generated ADJ sets are {(*May*,*bolehkah*,((*Q*)(*I*,*SI*)))} and {(*may*,*boleh*,((*S*)(*I*)))} respectively (see Figure 5.4). Thus, adding a morphological construction to compose interrogative word “bolehkah” can be done simply by adding “kah” to the target word “boleh” when the algorithm identifies *Q* connector on the left of the word “boleh”. Thus, similar to what was suggested in the phrase-based system, adding a word stemmer and a morphological analyzer/construction for the ADJ approach can solve phrases in interrogative sentences, phrases in adjective clauses, and phrases in passive sentences.

Interesting results obtained from Category VI and VIII where some cases could be resolved but some could not. For instance, in Category VI, a possessive noun phrase of “Aladdin’s good fortune” could be solved by the developed transfer rules while another possessive noun phrase of “the fisherman’s story” could not be solved. It was found that the developed transfer rules only considered a single word of possessor, e.g. “Aladdin”. Hence, we need to find a mechanism that will not ignore the possessor that contains two words or more in the possessive noun phrase e.g. “the fisherman” in the phrase “the fisherman’s story”. However, the mechanism seems to be complicated. Therefore, the hierarchical phrase-based method will be one of the good options to work out this problem.

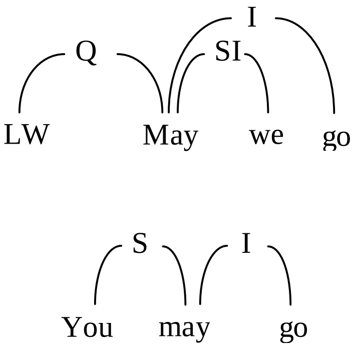


Figure 5.4: Connectors of “may” in phrases “May we go” and “You may go”

The lower percentages of unsolved cases belong to Category II (i.e. infinitive phrases) and Category III (i.e. “ing” form phrases). Most cases in the infinitive phrases were solved. An infinitive is composed from “to” followed by the simple form of a verb. The unsolved case arises when the infinitive must be translated into an Indonesian passive verb. For instance, “to wear” in “for Cinderella to wear to the Ball” must be translated into the Indonesian verb “dipakai”. Meanwhile, “to wear” was translated into an active verb “memakai” by the developed system. The other unsolved cases are found when ambiguous infinitives appear e.g. “to”, which must be translated into “untuk” (for), was translated into “ke” (to). One way to solve the cases in Category II is by combining phrase-based statistical translation with the developed system.

Cases in the “ing” form phrases can be solved if they are in the present or past continuous forms. The unsolved cases were found in present continuous forms which uses apostrophe like “Nothing’s coming” and which is in negative forms like “is not moving”; and in present perfect continuous forms such as “has been doing it”. These cases can be solved by some modification on the existing phrase-based transfer rules. For example, the existing transfer rules incorrectly translated “is not moving” into the Indonesian phrase “sedang tidak bergerak” while the correct translation is “tidak sedang bergerak”. There must be a swapping attempt between the word “sedang” and “tidak”. The generated ADJ set for this negative “ing” form phrases is unique e.g. {(*is*,*sedang*,((*Ss*)(*Pa*,*EBm*))), (*not*,*tidak*,((*EBm*)( )))} (see Figure 5.5) so that the words swapping will be made possible if there is a mechanism to identify all the connectors.

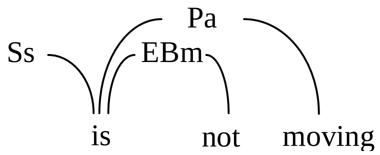


Figure 5.5: Connectors of the phrase “is not” in “is not moving”

The last categories to be discussed are Category I and IV. The phrases found in Category I were “the next morning” and “next time” and were already solved by the system, showing that the developed phrase-based transfer rules worked well with idiomatic time phrases. Meanwhile, all phrases in Category IV or phrases with pronoun “one” were not properly translated. However, it was found that the generated ADJ set for this form of phrases is also unique and thus correct phrase translations can be achieved by using the generated connectors.

Although incorporating phrase-base module in our previous ADJ-based system does not significantly increase the accuracy, there are other benefits can be obtained. The number of transfer rules generated in the phrase translation system is very few compared with those in the previous ADJ-based system. More than one hundred transfer rules in the previous ADJ-based system have been developed and we decided to stop the attempt since the tasks required us to observe the complete disjuncts of all the words in target sentences. This became difficult in the way that each word can have different disjunct in different sentence or context. It was found that generalization of the transfer rules will decrease the total number of transfer rules which in turn will ease the effort of generating the transfer rules. The phrase-based module answered the problems as the generated transfer rules became around 60 only.

However, there is a drawback with the phrase-based module in terms of the algorithm complexity. The Phrase-Based Transfer Rules Algorithm (see Sub Section 3.3.2) has the complexity of $O(n^5)$ while the sentence-based is $O(n^4)$. In this calculation, the Link Parser algorithm with the complexity of $O(n^3)$ is taken into account since this parser is called by both transfer rules.

5.3 Evaluation of the Hierarchical Phrase-Based MT System Using ADJ

The precision of all tested systems in BLEU metric is demonstrated in Figure 5.6. It is shown that the hierarchical phrase-based system precision increased with a slight difference higher than the previous phrase-based system precision for 3-gram, 4-gram, and 5-gram with 0.38%, 0.34%, and 0.28%, respectively. Most of the solved and unsolved cases that were found in the previous phrase-based system were also found in the hierarchical phrase-based system. However, there were cases that could be translated by the hierarchical phrase-based system with higher precision which were found in interrogative sentences like “What magic do you use?”; in adjective clauses such as “the money he had”, “the moment I dreamed of”, “Robin Hood, who slipped away”, and “The old woman, who was really the fairy”; and in passive sentences for examples “will be well paid”, “had been promised”. In addition, the use of hierarchical phrase-based module, as suggested in the discussion of phrase-based MT system, could correctly translate cases in possessive noun phrases. Possessors with two words or more were no longer ignored by the Hierarchical Phrase-Based Transfer Rules Algorithm. For instances in possessive noun phrases “the fisherman’s story”, “the lion’s cry”, “Baby Bear’s little bed”, “the handsome prince’s future”, “the Incredible family’s problems”, and “a girl’s game”.

It can be seen from Table 5.2 in which phrases in interrogative sentences, possessive noun phrases, phrases in adjective clauses, and phrases in passive sentences contributed significantly to the unsolved cases in the phrase-based MT system using ADJ. Thus, incorporating the hierarchical phrase-base module is valuable. The numbers of transfer rules generated in the hierarchical phrase-based translation system are 32 only. These are fewer than those generated in the phrase-based system. Moreover, the algorithm hierarchy makes ease the effort of generating and managing the transfer rules as the rules can be classified into simple (or the first group), more complex (or the second group), and most complex (or the third group) rules.

The disadvantage with the hierarchical phrase-based translation is the algorithm complexity of $O(n^6)$ compared with the phrase-based of $O(n^5)$. However, it is indeed possible to gain insight of the hierarchical phrase-based transfer rules and adapt them into the phrase-based MT system so that we can get all the advantages of the hierarchical phrase-based module and at the same time obtain the algorithm complexity of $O(n^5)$.

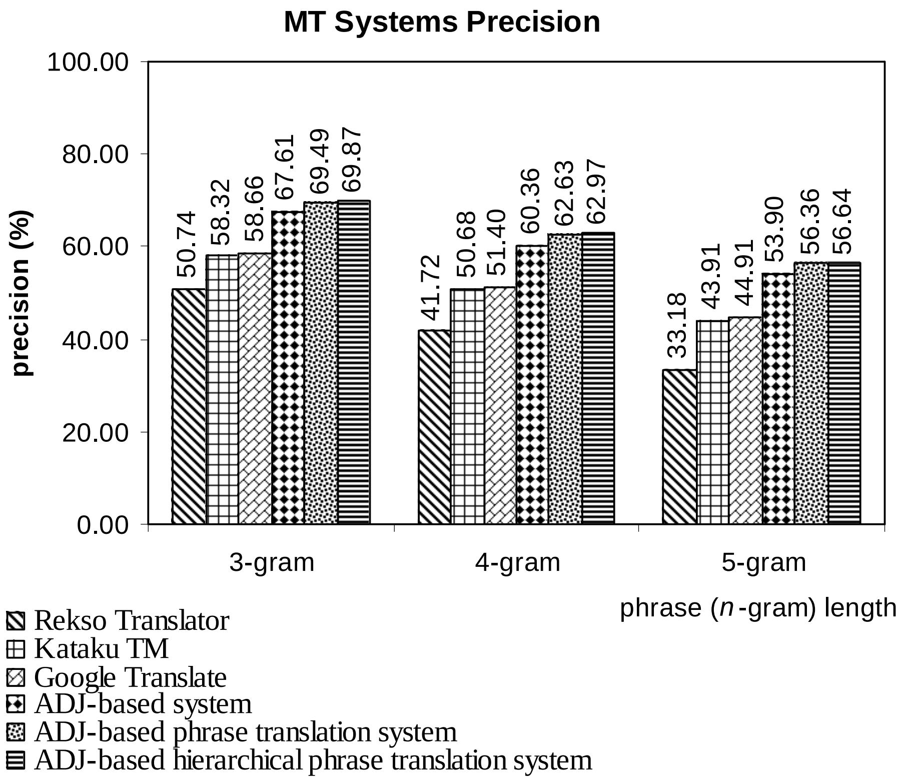


Figure 5.6: Comparison of hierarchical phrase-based system with other systems

5.4 Summary

This section evaluates and compares the developed English-Indonesian MT system with three available transfer rules algorithms over other available English-Indonesian systems. To assure valid evaluation and comparison, two methods namely SSER and BLUE metrics were used. Strength and weakness of each system was discussed here to give readers idea on how to develop a good English-Indonesian MT system or to enhance the existing methods used by the available systems. From the results of this study, the following order of accuracy using SSER test was constructed: ADJ-based system > Kataku^{TM} > Rekso Translator > Translator XP. Although the developed system seemed to outperform other tested systems, its accuracy is still low (71.17%).

With more training data fed to our MT system, not only the higher accuracy will be achieved but also the more robust transfer rules are expected to be created. Based on the scores of the human judges, the improvement of the accuracy can be achieved with the use of a word stemmer, morphological analyzer, POS, ontology, and phrase-based statistical translation module.

The order of the precision for the BLEU test was: ADJ-based hierarchical phrase translation system > ADJ-based phrase translation system > ADJ-based system > Google Translate > KatakuTM > Rekso Translator. Although the hierarchical phrase-based module can correctly translate possessive noun phrases, there are many cases that still could not be solved. Those unsolved cases were mostly found in interrogative sentences, adjective clauses, negative sentences, and passive sentences. Comparison of the three transfer rules, namely sentence-based transfer rules, phrase-based transfer rules, and hierarchical phrase-based transfer rules, which are used in our ADJ-based MT system, is conducted. The evaluation results of the precision versus the algorithm complexity of these three transfer rules are depicted in Figure 5.7.

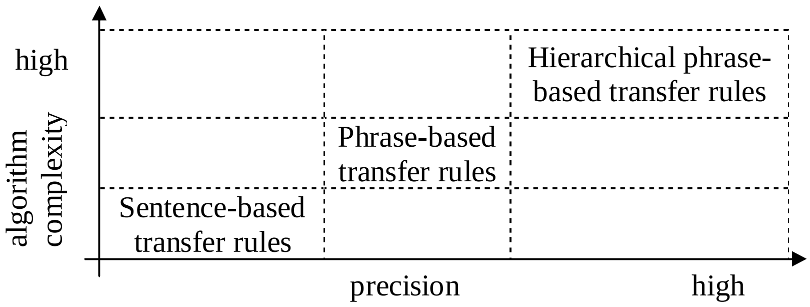

high
algorithm complexity
Hierarchical phrase-based transfer rules
Phrase-based transfer rules
Sentence-based transfer rules
precision
high

Figure 5.7: Precision versus algorithm complexity of three kind transfer rules

It can be seen from Figure 5.7 that the hierarchical phrase-based transfer rules could achieve the highest precision but also need the highest algorithm complexity. On the other hand, the sentence-based transfer rules gained the lowest precision but has the lowest algorithm complexity.

The evaluation of the number of the generated transfer rules versus the transfer rules building effort is shown in Figure 5.8. The figure demonstrates that the number of hierarchical phrase-based transfer rules is the greatest but the development effort is the simplest. On the contrary, the number of sentence-based transfer rules is the smallest but the development effort is the most complex.

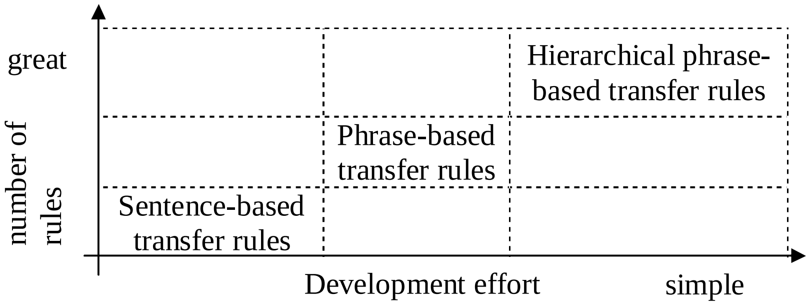

great
number of rules
Hierarchical phrase-based transfer rules
Phrase-based transfer rules
Sentence-based transfer rules
Development effort
simple

Figure 5.8: The number of transfer rules versus development effort

If all bilingual word pairs are completely annotated, the development of a reversed transfer rules from Indonesian to English is indeed possible since English and Indonesian languages share many similar word POS. Many POS of source words which are indicated by the ADJ set coincidently are also valid as the same POS of the target words. This finding will boost the development of an Indonesian parser which in turn might lead to different directions of research such as NER or IE of Indonesian language because those kinds of researches intensively observe the POS and the structure of Indonesian sentences.

CHAPTER 6

CONCLUSIONS

This chapter sums up all research outcomes covering the proposed MT method, contributions, limitations, and future works.

6.1 Annotated Disjunct for Machine Translation

Annotated Disjunct in LG formalism is a new hybrid transfer approach for bilingual MT system which performs translation tasks from major language to less-resourced language texts. The proposed method is appropriate when the pair of source-target languages does not have bilingual corpora and the target language does not have available grammar formalism and parser. In this method, LG formalism is used as a platform where the Link Parser – a free English parser built in LG formalism – is modified and utilized to generate word disjuncts of a given input sentence. The word disjuncts together with English words and the target words in Indonesian are annotated as ADJ set and are used as parameter in the translation process. This method is referred as ADJ-based method. Transfer rules for the ADJ-based MT system are then built. To prove that the method works, an ADJ-based English-Indonesian MT system is built and its performance compared with other English-Indonesian MT systems.

6.1.1 Annotated Disjunct in LG formalism

The conclusions of the disjunct annotation for machine translation are as follows.

1. The annotated disjunct in LG formalism is a suitable method for a bilingual MT system on conditions that there is no available bilingual corpus and the language pair is major/less-resourced language pair.
2. Translation tasks which are of one-to-one, one-to-many, and many-to-many in nature can theoretically be handled by this MT system.
3. The transfer rules development in this MT system will not need extra work since this model does not consider the head, constituent and dependent levels.
4. LG formalism can be used as a platform of the MT model which is in line with linguistics intuition better than other grammar formalisms. The fact that it does not recognize the head, constituent and dependent levels will also ease and simplify the transfer rules development.

6.1.2 Transfer Rules for the ADJ-Based MT System

The development of transfer rules based on ADJ method has the following conclusions.

1. The order of accuracy using SSER test is: ADJ-based system > Kataku^{TM} > Rekso Translator > Translator XP.
2. The order of the precision using BLEU test is: ADJ-based hierarchical phrase translation system > ADJ-based phrase translation system > ADJ-based system > Google Translate > KatakuTM > Rekso Translator.
3. The hierarchical phrase-based transfer rules can generalize the transfer rules for similar translation cases which in turn reduce the number of transfer rules thus will ease the effort of transfer rules generation.
4. The hierarchical phrase-based module can translate possessive noun phrases.

6.1.3 ADJ-Based English-Indonesian MT System

The development of transfer rules based on ADJ method has the following conclusions.

1. The accuracy depends on the number of transfer rules.
2. Higher accuracy is expected to be achieved with more example data fed to the MT system for generating transfer rules, instead of using only 300 bilingual sentences.
3. The developed system outperforms other tested systems, however its accuracy is still low (71.17%).
4. Generally, the developed ADJ-based MT system translated simple, compound, and complex English sentences in present, present continuous, present perfect, past, past perfect, and future tenses with better precision than other systems.

6.2 Limitations of the Method, the Transfer Rules, and the MT System

The following lines are the limitations of the ADJ-based method, the transfer rules, and ADJ-based MT system.

1. The ADJ-based method cannot solve non-projective cases (the problems arise in the dependent or constituent levels).
2. The translation in the ADJ-based method is only in one direction, which is from English to the *TL* (e.g. Indonesian language).
3. The hierarchical phrase-based module used in the latest version of the ADJ-based MT system still cannot solve several phrases which are found mostly in interrogative sentences, adjective clauses, negative sentences, and passive sentences.
4. The hierarchical phrase-based module also has very high algorithm complexity, thus the system is applicable for translation tasks involving only few English sentences, for example for translating some English sentences found in the Internet.

6.3 Future Works

In future works, the problem arise from non-standard cases should be addressed. It is possible by introducing word class parameter in this model, or by utilizing more than one English parser that can result in English sentence c-structure.

There are more issues that can be discussed in this research which were not found in the example data, such as sayings and phrasal verbs. Hence, a database of all sentences in both categories which consists of phrase-by-phrase mapping, that translates the whole source phrase into the whole target phrase as it is, is needed. Another way is by the combination of the developed hierarchical phrase-based method with the hierarchical phrase-based SMT method. The ambiguous Indonesian meaning of the English words for some cases could not be solved with the disjunct annotation. Thus, ontology and the use of other parameters, such as the word class, are possible approaches to address this problem.

Some of the tested sentences consist of words in English interrogative sentences which are translated differently when they are in passive or affirmative sentences. Two opportunities exist to improve the work, i.e. adding a word stemmer and a morphological analyzer for the ADJ-based method, which are expected to solve the mentioned problem.

Provided that all bilingual word pairs are already annotated, the development of a new parser for Indonesian language is indeed possible since English and Indonesian languages shared many similar word elements. The existence of an Indonesian parser will boost the development of new reversed transfer rules from Indonesian to English. The Indonesian parser will also lead to different directions of research, such as NER or IE of Indonesian language, because those kinds of researches intensively observe the POS and the structure of Indonesian sentences.

The results that the developed ADJ-based MT system translates simple, compound, and complex English sentences in present, present continues, present perfect, past, past perfect, and future tenses with better precision than the other systems are typical since the tenses found in the first 300 training data were also found in the 150 testing data. An evaluation on whether the precision will decrease should thus be conducted when the *SS* were in other tenses.

REFERENCES

[1] Abney, S., “Partial parsing via finite-state cascades,” *Natural Language Engineering*, vol. 2, no. 4, pp. 337-344, 1996.

[2] Abu-Salem, H., Al-Omari, M. and Evens, M.W., “Stemming Methodologies over Individual Query Words for An Arabic Information Retrieval System,” *Journal of the American Society for Information Science*, vol. 50, no. 6, pp. 524, Paper 9, 1999.

[3] Adriani, M., “Using Statistical Term Similarity for Sense Disambiguation in Cross-Language Information Retrieval,” *Information Retrieval*, Kluwer Academic Publishers, Manufactured in The Netherlands, vol. 2, pp. 69-80, 2000.

[4] Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S.M.M., and Williams, H.E., “Stemming Indonesian: A Confix-Stripping Approach,” *ACM Trans. on Asian Language Information Processing*, vol. 6, no. 4, Article 13, 2007.

[5] Adriani, M. and Croft, W.B., “The Effectiveness of a Dictionary-Based Technique for Indonesian-English Cross-Language Text Retrieval,” University of Massachusetts, Amherst, CIIR Tech. Rep. IR-170, 1997.

[6] Agarwal, A. and Lavie, A., “Meteor, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output,” in *Proc. Third Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Columbus, Ohio, Jun. 2008, pp. 115-118.

[7] Ahmad, F., Yusoff, M. and Sembok, T.M.T., “Experiments with a stemming algorithm for Malay words,” *Journal of the American Society for Information Science*, vol. 47, no. 12, pp. 909, Paper 18, 1996.

[8] Al-Adhaileh, M.H. and Tang, E.K., “Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema,” in *Proc. Machine Translation SUMMIT VII (MTS-VII)*, Singapore, Sep. 1999, pp. 244-249.

[9] Al-Adhaileh, M.H. and Tang, E.K., “Synchronous Structured String-Tree Correspondence (S-SSTC),” in *Proc. 20th IASTED02 Int. Conf.*, Innsbruck, Austria, Feb., pp. 270-275.

[10] Al-Adhaileh, M.H., Tang, E.K., and Zaharin, Y., “A Synchronization Structure of SSTC and Its Applications in Machine Translation,” presented at the COLING Post-Conf. Workshop on Machine Translation in Asia, Taipei, Taiwan, 2002, vol. 16, pp. 1-8.

[11] Alemayehu, N., “Development of a stemming algorithm for Amharic language text retrieval,” Ph.D dissertation, University of Sheffield, Sheffield, 1999.

[12] Alemayehu, N. and Willett, P., “The Effectiveness of Stemming for Information Retrieval in Amharic,” *Emerald*, vol. 37, no. 4, pp. 254-259, 2003.

[13] Al-Kharashi, I.A. and Evens, M.W., “Comparing Words, Stems and Roots as Index Terms in an Arabic Information Retrieval System,” *Journal of the American Society for Information Science*, vol. 45, no. 8, pp. 548, Paper 60, 1994.

[14] Amigo, E., Gimenez, J., Gonzalo, J., and Marquez, L., “MT Evaluation: Human-like vs. Human Acceptable,” in *2006 Proc. COLING/ACL Main Conf. Poster Sessions*, Association for Computational Linguistics, Sydney, Jul, pp. 17-24.

[15] Argamon, S., Dagan, I., and Krymolowski, Y., “A memory-based approach to learning shallow natural language patterns,” in *1998 Proc. COLING/ACL*, Montreal, pp. 67-73.

[16] Asian, J., Williams, H.E., and Tahaghoghi, S.M.M., “A Testbed for Indonesian Text Retrieval,” in *Proc. 9th Australasian Document Computing Symp.*, Melbourne, Australia, 13 Dec. 2004.

[17] Asian, J., Williams, H.E., and Tahaghoghi, S.M.M., “Stemming Indonesian,” in *ACM Int. Conf. Proc. Series vol. 102 archive*, in *Proc. 28th Australasian Conf. on Computer Science*, Newcastle, Australia, 2005, vol. 38, pp. 307-314.

[18] Azar, B.S., *Fundamental of English Grammar*, 3rd ed. 10 Bank Street, White Plains, New York: Longman-Pearson Education, 2003.

[19] Backus, J. W., “The syntax and semantics of the proposed international algebraic language of the Zurch ACM-GAMM Conf.,” In *Information Processing: Proc. Int. Conf. on Information Processing*, Paris, UNESCO, 1959, pp. 125-132.

[20] Bali, R., and Mohamad, S.K., “Automatic Identification of Close Languages: Case study: Malay and Indonesian,” *ECTI Trans. on Computer and Information Technology*, vol. 2, no. 2, pp. 126-134, Nov. 2006.

[21] Berment, V., “Several Directions for Minority Languages Computerization,” in *2002 Proc. COLING/ACL.*, Taipei, Taiwan, vol. 2, pp. 1-5.

[22] Bond, F. and Shirai, S., “A Hybrid Rule and Example-based Method for Machine Translation,” in *Recent Advances in Example-Based Machine Translation*, Springer, 2003, ch. 7.

[23] Bond, F., Oepen, S., Siegel, M., Copestake, A., and Flickinger, D., “Open source machine translation with DELPH-IN,” in *Proc. Open-Source Machine Translation workshop at the 10th Machine Translation Summit*, Phuket, Thailand, 2005, pp. 15-22.

[24] Cablitz, G., Ringersmaand, J., and Kemps-Snijders, M., “Visualizing endangered indigenous languages of French Polynesia with LEXUS,” in *Proc. 11th Int. Conf. Information Visualization (IV'07)*, IEEE Computer Society, pp. 409-414.

[25] Caseli, H.M, Nunes, M.G.V, and Forcada, M.L., "Automatic induction of bilingual resources from aligned parallel corpora: application to shallow-transfer machine translation," *Machine Translation*, Springer Netherlands, vol. 20, no. 4, pp. 227-245, 2008.

[26] Chiang, D., "Hierarchical Phrase-Based Translation," *Computational Linguistics*, Association for Computational Linguistics, vol. 33, no. 2, pp. 201-228, 2007.

[27] Chomsky, N., "Three models for the description of language," *IRI Trans. on Information Theory*, vol. 2, no. 3, pp. 113-124, 1956.

[28] Collins, M. J., Hajic, J., Ramshaw, L. A., and Tillmann, C., "A statistical parser for Czech," in *Proc. Annu. Meeting on ACL-99*, College Park, Maryland, pp. 505-512.

[29] Covington, M.A., "Parsing Discontinues Constituents in Dependency Grammar," *Computational Linguistics*, Association for Computational Linguistics, vol. 16, pp. 234-236, 1990.

[30] Covington, M., "An Empirically Motivated Reinterpretation of Dependency Grammar," Research Rep. AI1994-01, University of Georgia, 1994.

[31] Dave, S., Parikh, J., and Bhattacharyya, P., "Interlingua-based English–Hindi Machine Translation and Language Divergence," *Machine Translation*, Kluwer Academic Publishers, Printed in the Netherlands, vol. 16, pp. 251-304, 2003.

[32] DeNeefe, S., Knight, K., Wang, W., and Marcu, D., "What Can Syntax-based MT Learn from Phrase-based MT?," in *2007 Proc. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ACL, Prague, June, pp. 755–763.

[33] Ding, J., Berleant, D., Xu, J., and Fulmer, A.W., "Extracting Biochemical Interaction from MEDLINE Using a Link Grammar Parser," in *Proc. 15th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'03)*, pp. 467.

[34] Doddington, G., "Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics," in *2002 Proc. HLT (Human Language Technology Conf.)*, San Diego, CA, pp. 228-231.

[35] Dwipayana, G., *Sari Kata Bahasa Indonesia (The Essence of Indonesian Language)*, Surabaya: Terbit Terang, 2001.

[36] Earley, J., "An effcient context-free parsing algorithm," *Commun. of the ACM*, vol. 6, no. 8, pp. 451-455, 1970.

[37] Ejerhed, E.I., "Finding clauses in unrestricted text by finitary and stochastic methods," in *Proc. 2nd Conf. on Applied Natural Language Processing*, ACL, 1988, pp. 219-227.

[38] Ekmekcioglu, F.C. and Willett, P., "Effectiveness of Stemming for Turkish Text Retrieval," *The Association for Information Management*, vol. 34, no. 2, pp. 195-200, Apr. 2000.

[39] Elliott, D., Hartley, A., and Atwell, E., "Rationale for a multilingual aligned corpus for machine translation evaluation," in *Proc. Int. Conf. on Corpus Linguistics (CL2003), Lancanster*, UK, pp. 191-200.

[40] Estrella, P., Hamon, O., and Popescu-Belis, A., “How much Data is Needed for Reliable MT Evaluation? Using Bootstrapping to Study Human and Automatic Metrics,” in *Proc. MT Summit XI*, Copenhagen, Denmark, 2007, pp. 167-174.

[41] Forcada, M.L., “Open-Source Machine Translation: An Opportunity for Minor Languages,” in *2006 Proc. 5th SALTMIL Workshop*, LREC, Genoa, Italy, 23 May.

[42] Frakes, W.B., ‘Stemming algorithms,” in *Information Retrieval: Data Structures & Algorithms*, Frakes, W.B. and Baeza-Yates, R., Eds. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 161-218.

[43] Fung, P., “A pattern matching method for finding noun and proper noun translations from noisy parallel corpora,” in *Proc. 33rd Annu. Meeting of the Association for Computational Linguistics*, Cambridge, MA, 1995, pp. 236-243.

[44] Galley, M., Hopkins, M., Knight, K., and Marcu, D., “What’s in a translation rule?,” in *Proc. HLT-NAACL*, 2004.

[45] Ghani, R., Jones, R., and Mladenic, D., “Building Minority Language Corpora by Learning to Generate Web Search Queries,” *Knowledge and Information Systems*, Springer-Verlag London Ltd., vol. 7, pp. 56-83, 2005.

[46] Google Translate (2006). [Online]. Available: http://translate.google.com.

[47] Grinberg, D., Lafferty, J., and Sleator, D., “A Robust Parsing Algorithm for A Link Grammar,” in *Proc. 4th Int. Workshop on Parsing Technologies*, Prague, 1995.

[48] Hajic, J., “Building a Syntactically Annotated Corpus: The Prague Dependency Treebank,” pp. 106-132, Karolinum, Prague, Praha, 1998.

[49] Hardjadibrata, R.R., “An Indonesian Newspaper Wordcount,” Department of Indonesian and Malay, Faculty of Arts, Monash University, Cayton, Victoria, Rep. 1969.

[50] Hovy, E.H., “Toward finely differentiated evaluation metrics for machine translation,” in *Proc. Eagles Workshop on Standards and Evaluation*, Pisa, Italy, 1999.

[51] Hudson, R., “Word Grammar,” in *Concise Encyclopedia of Syntactic Theories*, Brown, K. and Miller, J., Oxford: Elsevier, 1996, pp. 368-372.

[52] Hutchins, J. and Somers, H., *An introduction to machine translation*, London: Academic Press, 1992.

[53] Indradjaja, L.S. and Bressan, S., “Automatic Learning of Stemming Rules for the Indonesian Language,” in *Proc. 17th Pacific Asia Conf.*, Sentosa, Singapore, 1-3 Oct. 2003, pp. 62-68.

[54] Järvinen, T. and Tapanainen, P., “Towards an Implementable Dependency Grammar,” in *1998 Proc. COLING-ACL Workshop: Processing of Dependency-Based Grammars*, pp. 1-10.

[55] Jurafsky, D. and Martin, J.H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Prentice Hall, 21 Aug. 2008.

[56] Kaji, H., Kida, Y., and Morimoto, Y., “Learning Translation Templates from Bilingual Text,” *Proc. 14th Int. Conf. On Computational Linguistics (COLING-92)*, vol. 2, pp. 672-678.

[57] Kalamboukis, T.Z., “Suffix stripping with modern Greek,” *Program*, vol. 29, no. 4, Paper 21, pp. 313, 1995.

[58] Kaplan, R.M., “A general syntactic processor,” in *Natural Language Processing*, Rustin, R., Ed. New York: Algorithmics Press, 1973, pp. 193-241.

[59] Kasami, T., “An ef_cient recognition and syntax analysis algorithm for context-free languages,” Air Force Cambridge Research Laboratory, Bedford, MA, Tech. Rep. AFCRL-65-758, 1965.

[60] Kataku^{TM} (2007). [Online]. Available: http://www.toggletext.com/kataku_trial.php.

[61] Kay, M., “Algorithm schemata and data structures in syntactic processing,” in *Readings in natural language processing*, San Francisco, CA: Morgan Kaufmann Publishers Inc, 1986, pp. 35-70.

[62] Keraf, G., *Tata Bahasa Rujukan Bahasa Indonesia*, Jakarta: PT Gramedia Widiasarana, 1999.

[63] Koehn, P. and Knight, K., “Learning a translation lexicon from monolingual corpora,” in *Proc. Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, Philadelphia, PA, 2002, pp. 9-16.

[64] Koehn, P., Och, F.J., and Marcu, D., “Statistical Phrase-Based Translation,” in *Proc. HLT–NAACL*, Main Papers, Edmonton, May-Jun. 2003, pp. 48-54.

[65] Kraaij, W. and Pohlmann, R., “Viewing stemming as recall enhancement,” in *Proc. 19th Annu. Int. ACM/SIGIR Conf.*, Association for Computing Machinery, New York, 1996, Paper 8, pp. 40.

[66] Kruijff, G.J.M. (2007). Formal and Computational Aspects of Dependency Grammar – Heads, Dependents, and Dependency Structures. [Online]. Available: Computational Linguistics Department, University of the Saarland, Saarbrucken, Germany.

[67] Langlais, P., Foster, G., and Lapalme, G., “Integrating bilingual lexicons in a probabilistic translation assistant,” in *Proc. MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, 2001, pp. 197–202.

[68] Lavoie, B., White, M., and Korelsky, T., “Learning Domain-SpecificTransfer Rules: An Experiment with Korean to English Translation,” in *Proc. 19th Int. Conf. on Computational Linguistics: Workshop on Machine Translation in Asia*, 2002, vol. 16, pp 1-7.

[69] Lennon, M., Pierce, D.S., Tarry, B.D. and Willett, P., “An Evaluation of Some Conflation Algorithms for Information Retrieval,” *Journal of Information Science*, vol. 3, no. 4, Paper 83, pp. 177, 1981.

[70] Lopez, A., “Hierarchical Phrase-Based Translation with Suffix Arrays,” in *2007 Proc. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Jun., pp. 976–985.

[71] Marcu, D. and Wong, W., “A phrase-based, joint probability model for statistical machine translation,” in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Philadelphia, Jul. 2002, pp. 133-139.

[72] Marcu, D., Wang, W., Echihabi, A., and Knight, K., “SPMT: Statistical Machine Translation with Syntactified Target Language Phrases,” in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Sidney, Jul. 2006, pp. 44-52.

[73] Maxwell, M. (2003). Incremental Grammar Development using Finite State Tools. [Online]. Available: ACL Anthology database.

[74] Melamed, I.D., Green, R., Turian, J.P., “Precision and Recall of Machine Translation,” in *2003 Proc. NAACL HLT*, Edmonton, Canada, vol. 2, pp. 61-63.

[75] Melcuk, I.A., *Dependency Syntax: Theory and Practice*, Albany, New York: SUNY Press, 1988.

[76] Muhadjir, “Menjaring Data dari Teks,” *Lembaran Sastra Universitas Indonesia, special edition: Tautan Sastra & Komputer*, Faculty of Letters, University of Indonesia, Depok, pp. 81-91, 1995.

[77] Munoz, M., Punyakanok, V., Roth, D., and Zimak, D., “A learning approach to shallow parsing,” in *Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, ACL, College Park, MD, pp. 168-178.

[78] Nakamura, J., Tsujii, J., and Nagao, M., “Solutions for problems of MT parser: methods used in Mu-machine translation project,” in *Proc. 11th Int. Conf. on Computational Linguistics (COLING-86)*, Bonn, Germany, pp. 133 – 135.

[79] Nazief, B., “Development of Computational Linguistics Research: a Challenge for Indonesia. Faculty of Computer Science,” University of Indonesia, Tech. Rep., 1996.

[80] Nießen, S., Och, F.J., Leusch, G., and Ney, H., “An evaluation tool for machine translation: Fast evaluation for MT research,” in *Proc. 2nd Int. Conf. on Language Resources and Evaluation (LREC)*, Athens, Greece, 2000, pp. 39-45.

[81] NIST (2006). NIST Open Machine Translation (MT) Evaluation. [Online]. Information Access Division (IAD), Information Technology Laboratory, U.S. Department of Commerce. Available: http://www.itl.nist.gov/iad/mig/tests/mt/

[82] Noone, G., “Machine Translation A Transfer Approach,” Computer Science, Linguistics and a Language (CSLL) Department, University of Dublin, Trinity College, Final Rep., May 2003.

[83] Noor, N.M., Paul-Evanson, C., Mat, H., Karuthan, K., Nashir, S.M., Khuan, A.L.C., Ismail, M., Ghazali, A.R., Shaari, F., Misdan, M., and Jais, I.R.M., *Vision: Focus on Grammar*, Academy of Language Studies Universiti Teknologi Mara: McGraw-Hill Education (Malaysia), 2003.

[84] Novento, F., “Perangkat Lunak Penerjemah Kalimat Inggris-Indonesia Menggunakan Metode Loading Data Sementara,” Electrical Engineering Department, Gadjah Mada University, Final Rep., 2003.

[85] Nusai, C., Suzuki, Y., and Yamazaki, H., “Estimating Word Translation Probabilities for Thai – English Machine Translation using EM Algorithm,” *International Journal of Computational Intelligence*, WASET, vol. 4, no. 3, Summer 2008.

[86] Och, F. J., Tillmann, C., and Ney, H., “Improved alignment models for statistical machine translation,” in *Proc. Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 20-28.

[87] Och, F. J. and Ney, H., “Improved statistical alignment models,” in *Proc. 38th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Hong Kong, 2000, pp. 440-447.

[88] Och, F. J. and Ney, H., “The alignment template approach to statistical machine translation,” *Computational Linguistics*, Association for Computational Linguistics, MIT Press, Cambrige, MA, USA, vol. 30, pp. 417-449, 2004.

[89] Papineni, K., Roukos, S., Ward, T., and Zhu, W.J., “BLEU: a method for automatic evaluation of machine translation,” in *Proc. 40th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Pennsylvania, 2002, pp. 311-318.

[90] Paulik, M., Rottmann, K., Niehues, J., Hildebrand, S., and Vogel, V., “The ISL phrase-based MT system for the 2007 ACL workshop on statistical MT,” in *Proc. Association of Computational Linguistics Workshop on Statistical Machine Translation*, 2007, pp. 197-202.

[91] Pepescu-Belis, A., “An Experiment in Comparative Evaluation: Humans vs. Computers,” in *Proc. MT Summit IX*, New Orleans, LA USA, 2003, pp. 307-314.

[92] Popovic, M. and Willett, P., “The effectiveness of stemming for natural language access to Slovene textual data,” *Journal of the American Society for Information Science*, vol. 43, no. 5, Paper 90, pp. 384, 1992.

[93] Poulsen., C.S. (2007). Translator XP. [Online]. CV Media Internusa Enterprice, Yogyakarta, Indonesia. Available: http://translatorxp.com.

[94] Probst, K., “Learning Transfer Rules for Machine Translation with Limited Data,” Ph.D. dissertation, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 15 Aug. 2005.

[95] Purwarianti, A., Tsuchiya, M., and Nakagawa, S., “Indonesian-Japanese CLIR Using Only Limited Resource,” in *Proc. Workshop on How Can Computational Linguistics Improve Information Retrieval?*, ACL, Sydney, Jul. 2006, pp. 1-8.

[96] Ramshaw, L. A. and Marcus, M. P., “Text chunking using transformation-based learning,” in *Proc. Third Annu. Workshop on Very Large Corpora*, ACL, 1995, pp. 82-94.

[97] Riza, H., “Resources Report on Languages of Indonesia,” in *Proc. 6th Workshop on Asian Language Resources*, Hyderabad, India, 11-12 Jan. 2008, pp. 93-94.

[98] Rekso Translator (2007). [Online]. Indonesia Commerce, Indonesia. Available: http://reksotranslator.com/.

[99] Sadler, V. and Vendelmans, R., “Pilot Implementation of a Bilingual Knowledge Bank,” in *Proc. COLING-90*, Helsinki, Finland, vol. 3, pp. 449-451.

[100] Sakti, S., Kelana, E., Riza, H., Sakai, S., Markov, K., and Nakamura, S., “Development of Indonesian Large Vocabulary Continuous Speech Recognition System within A-STAR Project,” in *Proc. Workshop on Technologies and Corpora for Asia-Pacific Speech Translation*, Hyderabad, India, 2008.

[101] Sanchez-Martinez, F., Armentano-Oller, C., P´erez-Ortiz, J.A., and Forcada, M.L., “Training Part-of-Speech Taggers to build Machine Translation Systems for Less-Resourced Language Pairs,” in *Proc. 23th Congr. of Spanish Society of Natural Language Processing*, University of Sevilla, 10-12 Sep. 2007, pp. 257-264.

[102] Sari, Y., Hassan, M.F., and Zamin, N., “A Hybrid Approach to Semi-Supervised Named Entity Recognition in Health, Safety and Environment Reports,” in *Proc. Int. Conf. on Future Computer and Communication (ICFCC 2009)*, Kuala Lumpur, Malaysia, 3-5 Apr, pp. 599-602.

[103] Schafer, C. and Yarowsky, D., “Inducing translation lexicons via diverse similarity measures and bridge languages,” in *Proc. Int. Conf. On Computational Linguistics (CoNLL-2002)*, Taipei, Taiwan, pp 1-7.

[104] Schneider, G., “A Linguistic Comparison of Constituency, Dependency and Link Grammar,” M.S. thesis, University of Zurich, 1998.

[105] Senellart, J., Dienes, P., and V´aradi, T., “New generation systran translation system,” in *Proc. MT Summit VIII*, Spain, 18-22 Sep. 2001.

[106] Sgall, P. and Panevova, J., “Dependency Syntax – A chalenge,” *Theoritical Linguistics*, vol. 15, no. 1, pp. 73-86, 1989.

[107] Shaalan, K., Rafea, A., Moneim, A. A., and Baraka, H., “Machine Translation of English Noun Phrases into Arabic,” *International Journal of Computer Processing of Oriental Languages*, World Scientific, vol. 17, No. 2, pp. 121-134, 2004.

[108] Siregar, N.E.F., “Pencari Kata Berimbuhan pada Kamus Besar Bahasa Indonesia dengan Menggunakan Algoritma Stemming,” University of Indonesia. Jakarta, Indonesia, Final Rep., 1995.

[109] Sleator, D.D. and Temperley, D., “Parsing English with A Link Grammars,” in *Proc. 3rd Int. Workshop on Parsing Technologies, ACL - SIGPARSE Conf.*, University of Tilburg, The Netherlands, 1993.

[110] Solak, A. and Oflazer, K., “Design and implementation of a spelling checker for Turkish,” *Literary and Linguistic Computing*, vol. 8, no. 3, Paper 30, pp. 113, 1993.

[111] Somers, H., “Machine Translation and Welsh: The Way Forward,” A Report for the Welsh Language Board, Centre for Computational Linguistics, UMIST, Manchester, Jul. 2004.

[112] Sproat, R., *Morphology and Computation*, Cambridge, MA: MIT Press, 1992.

[113] Tanaka, Y., “Example data for machine translation systems,” in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, 7-10 Oct. 2001, vol. 2, pp. 915-920.

[114] Tang, E.K. and Al-Adhaileh, M.H., “Converting a Bilingual Dictionary into a Bilingual Knowledge Bank based on the Synchronous SSTC”, in *Proceedings of Machine Translation Summit VIII*, Spain, 18 Sep. 2001, pp. 351-356.

[115] Trujillo, A., *Translation Engines: Techniques for Machine Translation*, London: Springer-Verlag, 1999.

[116] Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S., and Badia, T., "METIS-II: Machine Translation for Low Resource Languages," in *Proc. 5th Int. Conf. on Language Resources and Evaluation*, Genoa, Italy, 2006, pp. 1284-1289.

[117] Venable, P., "Modeling Syntax for Parsing and Translation," Ph.D. dissertation, School of Computer Science, Computer Science Department, Carnegie Mellon University, 15 Dec., 2003.

[118] Venugopal, A., Zollmann, A., and Vogel, S., "An Efficient Two-Pass Approach to Synchronous-CFG Driven Statistical MT," in *Proc. of NAACL HLT 2007*, Rochester, NY, Apr., pp. 500–507.

[119] Vinsensius B.V.S.N., "Information Retrieval for the Indonesian Language," M.S. thesis, National University of Singapore, Singapore, 2001.

[120] Watanabe, H. and Takeda, K., "A pattern-based machine translation system extended by example-based processing," in *Proc. 17th Int. Conf. on Computational Linguistics (COLING-98)*, Montreal, Quebec, Canada, vol. 2, pp. 1369-1373.

[121] Weber, H.J., *Dependenzgrammatik. Ein interaktives Arbeitsbuch*, Auflage 2, Tübingen: Gunter Narr Verlag, 1997.

[122] White, J.S. and O'Connell, T., "The ARPA MT evaluation methodologies: evolution, lessons, and future approaches," in *Proc. First Conf. of the Association for Machine Translation in the Americas (AMTA-1994)*, Columbia, Maryland, pp. 193-205.

[123] Widyamartaya, A., *Seni Menerjemahkan*, 13th ed. Yogyakarta: Kanisius, 2003.

[124] Wilujeng, A., *Inti Sari Kata Bahasa Indonesia Lengkap*, Surabaya: Serba Jaya, 2002.

[125] Wu, D. and Xia, X., "Learning an English–Chinese lexicon from parallel corpus," in *Proc. 1st Conf. of the Association for Machine Translation in the Americas (AMTA-1994)*, Columbia, MD, pp. 206–213.

[126] Xu, J. L., "Multilingual Search on the World Wide Web," in *Proc. Hawaii Int. Conf. on System Sciences HICSS-33*, Maui, Hawaii, Jan. 2000.

[127] Yamada, K. and Knight, K., "A syntax-based statistical translation model," in *Proc. 39th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Toulouse, France, 2001, pp. 523-530.

[128] Yamada, K. and Knight, K., "A Decoder for Syntax-based Statistical MT," in *Proc. 40th Annu. Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Jul. 2002, pp. 303-310.

[129] Younger, D.H., "Recognition and parsing of context-free languages in time n^3," *Information and Control*, vol. 10, pp. 189-208, 1967.

[130] Yusuf, H., "An Analysis of Indonesian Language for Interlingual Machine-Translation System," in *Proc. 14th Conf. on Computational Linguistic (COLING)*, Nantes, France, 23-28 Aug. 1992, vol. 4, pp. 1228-1232.

[131] Zamin, N., “Information Extraction Using Link Grammar,” in *Proc. World Congr. On Computer Science and Information Engineering*, Los Angeles, CA, 31 Mar.-2 Apr. 2009, pp. 149-153.

[132] Zhang, Y. and Vogel, S., “Measuring Confidence Intervals for the Machine Translation Evaluation Metrics,” in *Proc. Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI 2004)*, Baltimore, MD, 4-6 Oct.

[133] Zhang, Y., Vogel, S., and Waibel, A., “Interpreting BLEU/NIST scores: How much improvement do we need to have a better system?,” in *Proc. Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, pp. 2051-2054.

[134] Zens, R., Och, F.J., and Ney, H., “Phrase-Based Statistical Machine Translation,” in *Lecture Notes in Artificial Intelligence 2479*, Springer-Verlag Berlin Heidelberg, 2002, pp. 18-32.

[135] Zollmann, A., Venugopal, A., Paulik, M., and Vogel, S., “The Syntax Augmented MT (SAMT) System for the Shared Task in the 2007,” in *Proc. Second Workshop on Statistical Machine Translation*, ACL, Prague, Czech Republic, Jun. 2007, pp. 216-219.

Attended Conferences

1. Adji, T.B., Baharudin, B., and Zamin, N., “English-Indonesian Machine Translation using Link Grammar,” presented at the Int. Conf. on Computational Science (ICCS), Bandung, 3-4 Dec. 2007.
2. Adji, T.B., Baharudin, B., and Zamin, N., “Annotated Disjunct in Link Grammar for Machine Translation,” in *Proc. Int. Conf. on Intelligent & Advanced Systems (ICIAS)*, KL Convention Centre, Kuala Lumpur, 25-28 Nov. 2007.
3. Zamin, N., Adji, T.B., and Baharudin, B., “Development of Automated Text Sumarization System for Malay Language,” in *Proc. Int. Conf. on Intelligent & Advanced Systems (ICIAS)*, KL Convention Centre, Kuala Lumpur, 25-28 Nov. 2007.
4. Adji, T.B., Baharudin, B., and Zamin, N., “Building Transfer Rules using Annotated Disjunct: An Approach for Machine Translation,” in *Proc. 5th Student Conf. on Research and Development (SCOReD)*, UKM, Bangi, 11-12 Dec. 2007.
5. Adji, T.B., Baharudin, B., and Zamin, N., “Applying Link Grammar Formalism in the Development of English-Indonesian Machine Translation System,” in *9th Artificial Intelligence and Symbolic Computation (AISC)*, University of Birmingham, UK, 31 Jul.-1 Aug. 2008.

Publications

1. Adji, T.B., Baharudin, B., and Zamin, N., “Applying Link Grammar Formalism in the Development of English-Indonesian Machine Translation System,” in *Book Chapter in Intelligent Computer Mathematics, LNAI 5144 LNCS*, Autexier, S., Champbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F., Eds. Springer Berlin / Heidelberg, 2008, pp. 17-23.
2. Adji, T.B., Baharudin, B., and Zamin, N., “The Development of Phrase-Based Transfer Rules for ADJ-Based Machine Translation,” *International Journal of Recent Trends in Engineering*, Academy Publisher, Nov. 2009, vol.2, no. 1.

Awards

1. 2nd Runner Up of Best Presenter Postgraduate on Engineering Design Exhibition 2008.
2. Silver of Computer & Information Sciences Postgraduate on Engineering Design Exhibition 2008.

Appendix A

List of Link Types at a Glance

This appendix lists and describes LG connectors. This list is taken from Schneider [1998].

A / connects pre-noun ("attributive") adjectives to following nouns: "The BIG DOG chased me", "The BIG BLACK UGLY DOG chased me".

AA / is used in the construction "How [adj] a [noun] was it?". It connects the adjective to the following "a".

AF / connectives adjectives to verbs in cases where the adjective is fronted, such as questions and indirect questions: "How BIG IS it?"

AL / connects a few determiners like "all" or "both" to following determiners: "ALL THE people are here".

AN / connects noun-modifiers to following nouns: "The TAX PROPOSAL was rejected".

AZ \ connects the word "as" back to certain verbs that can take "[obj] as [adj]" as a complement: "He VIEWED him AS stupid".

B \ serves various functions involving relative clauses and questions. It connects transitive verbs back to their objects in cases like relative clauses and questions ("WHO did you HIT?"); it also connects the main noun to the finite verb in subject-type relative clauses ("The DOG who CHASED me was black").

THIS IS A PROBLEM CASE, AS SOMETIMES – SEEMS TO FIT BETTER

BI \ connects form of the verb "be" to certain idiomatic expressions: for example, cases like "He IS PRESIDENT of the company".

BT / is used with time expressions acting as fronted objects: "How many YEARS did it LAST?".

BW / connects "what" to various verbs like "think", which are not really transitive but can connect back to "what" in questions: "WHAT do you THINK?"

C / links conjunctions to subjects of subordinate clauses ("He left WHEN HE saw me"). It also links certain verbs to subjects of embedded clauses ("He SAID HE was sorry"). FIRST MAY BE FUNCTIONAL HEAD

CC - connects clauses to following coordinating conjunctions ("SHE left BUT we stayed").

CO / connects "openers" to subjects of clauses: "APPARENTLY / ON Tuesday, THEY went to a movie". SHOULD RATHER MODIFY THE VERB

CQ - connects to auxiliaries in comparative constructions involving s-v inversion: "SHE has more money THAN DOES Joe".

CX - is used in comparative constructions where the right half of the comparative contains only an auxiliary: "She has more money THAN he DOES".

EITHER INTRODUCES CLAUSE \ (LIKE COORD) OR SUBSTITUTES OBJ / D / connects determiners to nouns: "THE DOG chased A CAT and SOME BIRDS".

UNLESS UNDER DP HYPOTHESIS

DD / connects definite determiners ("the", "his") to number expressions certain things like number expressions and adjectives acting as nouns: "THE POOR", "THE TWO he mentioned". UNLESS UNDER DP HYPOTHESIS

DG / connects the word "The" with proper nouns: "the Riviera", "the Mississippi".

DP / connects possessive determiners to gerunds: "YOUR TELLING John to leave was stupid".

DT / connects determiners to nouns in idiomatic time expressions: "NEXT WEEK", "NEXT THURSDAY".

E / is used for verb-modifying adverbs which precede the verb: "He APPARENTLY not COMING".

EA / connects adverbs to adjectives: "She is a VERY GOOD player".

EB \ connects adverbs to forms of "be" before an object or prepositional phrase: "He IS APPARENTLY a good programmer". SHOULD PERHAPS MODIFY NOUN

EC / connects adverbs to comparative adjectives: "It is MUCH BIGGER"

EE / connects adverbs to other adverbs: "He ran VERY QUICKLY".

EF / connects the word "enough" to preceding adjectives and adverbs: "He didn’t run QUICKLY ENOUGH".

EI / connects a few adverbs to "after" and "before": "I left SOON AFTER I saw you".

EN / connects certain adverbs to expressions of quantity: "The class has NEARLY FIFTY students".

ER - is used the expression "The x-er..., the y-er...". it connects the two halfs of the expression together, via the comparative words (e.g. "The FASTER it is, the MORE they will like it").

FM \ connects the preposition "from" to various other prepositions: "We heard a scream FROM INSIDE the house".

G - connects proper noun words together in series: "GEORGE HERBERT WALKER
BUSH is here."

- OR / OR \ : UNIMPORTANT

GN - (stage 2 only) connects a proper noun to a preceding common noun which introduces it: "The ACTOR Eddie MURPHY attended the event".

H / connects "how" to "much" or "many": "HOW MUCH money do you have".

I \ connects certain words with infinitive verb forms, such as modal verbs and "to": "You MUST DO it", "I want TO DO it".

IN \ connects the preposition "in" to certain time expressions: "We did it IN DECEMBER".

J \ connects prepositions to their objects: "The man WITH the HAT is here".

JG \ connects certain prepositions to proper-noun objects: "The Emir OF KUWAIT is here".

JQ / connects prepositions to question-word determiners in "prepositional questions": "IN WHICH room were you sleeping?"

JT \ connects certain conjunctions to time-expressions like "last week": "UNTIL last WEEK, I thought she liked me".

K / connects certain verbs with particles like "in", "out", "up" and the like: "He STOOD UP and WALKED OUT".

L / connects certain determiners to superlative adjectives: "He has THE BIGGEST room".

LE \ is used in comparative constructions to connect an adjective to the second half of the comparative expression beyond a complement phrase: "It is more LIKELY that Joe will go THAN that Fred will go".

M \ connects nouns to various kinds of post-noun modifiers: prepositional phrases ("The MAN WITH the hat"), participle modifiers ("The WOMAN CARRYING the box"), prepositional relatives ("The MAN TO whom I was speaking"), and other kinds.

MG \ allows certain prepositions to modify proper nouns: "The EMIR OF Kuwait is here".

MV \ connects verbs and adjectives to modifying phrases that follow, like adverbs ("The dog RAN QUICKLY"), prepositional phrases ("The dog RAN IN the yard"), subordinating conjunctions ("He LEFT WHEN he saw me"), comparatives, participle phrases with commas, and other things.

MX - OR \ connects modifying phrases with commas to preceding nouns: "The DOG,
a POODLE, was black". "JOHN, IN a black suit, looked great".

N \ connects the word "not" to preceding auxiliaries: "He DID NOT go".

ND / connects numbers with expressions that require numerical determiners: "I saw him THREE WEEKS ago".

NF / is used with NJ in idiomatic number expressions involving "of": "He lives two THIRDS OF a mile from here".

NI \ OR - is used in a few special idiomatic number phrases: "I have BETWEEN 5 AND 20 dogs".

NN - or / connects number words together in series: "FOUR HUNDRED THOUSAND people live here".

NR / connects fraction words with superlatives: "It is the THIRD BIGGEST city in China".

NS / connects singular numbers (one, 1, a) to idiomatic expressions requiring number determiners: "I saw him ONE WEEK ago".

NW / is used in idiomatic fraction expressions: "TWO THIRDS of the students were women".

O \ connects transitive verbs to their objects, direct or indirect: "She SAW ME", "I GAVE HIM the BOOK".

OD \ is used for verbs like "rise" and "fall" which can take expressions of distance as complements: "It FELL five FEET".

OF / connects certain verbs and adjectives to the word "of": "She ACCUSED him OF the crime", "I’m PROUD OF you".

OT \ is used for verbs like "last" which can take time expressions as objects: "It LASTED five HOURS".

P \ connects forms of the verb "be" to various words that can be its complements: prepositions, adjectives, and passive and progressive participles: "He WAS [ANGRY / IN the yard / CHOSEN / RUNNING]".

PF / is used in certain questions with "be", when the complement need of "be" is satisfied by a preceding question word: "WHERE ARE you?", "WHEN will it BE?" PP \ connects forms of "have" with past participles: "He HAS GONE".

Q / is used in questions. It connects the wall to the auxiliary in simple yes-no questions ("///// DID you go?"); it connects the question word to the auxiliary in where-when-how questions ("WHERE DID you go").

QI \ connects certain verbs and adjectives to question-words, forming indirect questions: "He WONDERED WHAT she would say".

R \ connects nouns to relative clauses. In subject-type relatives, it connects to the relative pronoun ("The DOG WHO chased me was black"); in object-type relatives, it connects either to the relative pronoun or to the subject of the relative clause ("The DOG THAT we chased was black", "The DOG WE chased was black").

RS \ is used in subject-type relative clauses to connect the relative pronoun to the verb: "The dog WHO CHASED me was black".

RW - connects the right-wall to the left-wall in cases where the right-wall is not needed for punctuation purposes.

S / connects subject nouns to finite verbs: "The DOG CHASED the cat": "The DOG [IS chasing / HAS chased / WILL chase] the cat".

ROOT LINK TO SUBJ MAY NECESSITATE \

SF / is a special connector used to connect "filler" subjects like "it" and "there" to finite verbs: "THERE IS a problem", "IT IS likely that he will go".

SFI \ connects "filler" subjects like "it" and "there" to verbs in cases with subject-verb inversion: "IS THERE a problem?", "IS IT likely that he will go?"

SI \ connects subject nouns to finite verbs in cases of subject-verb inversion: "IS JOHN coming?", "Who DID HE see?"

TA / is used to connect adjectives like "late" to month names: "We did it in LATE DECEMBER".

TD \ connects day-of-the-week words to time expressions like "morning": "We’ll do it MONDAY MORNING".

TH \ connects words that take "that [clause]" complements with the word "that". These include verbs ("She TOLD him THAT..."), nouns ("The IDEA THAT..."), and adjectives ("We are CERTAIN THAT").

TI \ is used for titles like "president", which can be used in certain cirumstances without a determiner: "AS PRESIDENT of the company, it is my decision".

TM \ is used to connect month names to day numbers: "It happened on JANUARY 21".

TO \ connects verbs and adjectives which take infinitival complements to the word "to": "We TRIED TO start the car", "We are EAGER TO do it".

TQ / is the determiner connector for time expressions acting as fronted objects: "How MANY YEARS did it last".

TS \ connects certain verbs that can take subjunctive clauses as complements - "suggest", "require" - to the word that: "We SUGGESTED THAT he go".

TY \ is used for certain idiomatic usages of year numbers: "I saw him on January 21 , 1990 ". (In this case it connects the day number to the year number.)

U / is a special connector on nouns, which is disjoined with both the determiner and subject-object connectors. It is used in idiomatic expressions like "What KIND_OF DOG did you buy?"

UN \ connects the words "until" and "since" to certain time phrases like "after [clause]": "You should wait UNTIL AFTER you talk to me".

V \ connects various verbs to idiomatic expressions that may be non-adjacent: "We TOOK him FOR_GRANTED", "We HELD her RESPONSIBLE".

W \ connects the subjects of main clauses to the wall, in ordinary declaratives, imperatives, and most questions (except yes-no questions). It also connects coordinating conjunctions to following clauses: "We left BUT SHE stayed".

WN \ connects the word "when" to time nouns like "year": "The YEAR WHEN we lived in England was wonderful".

WR / connects the word "where" to a few verbs like "put" in questions like "WHERE did you PUT it?".

X - is used with punctuation, to connect punctuation symbols either to words or to each other. For example, in this case, POODLE connects to commas on either side: "The dog , a POODLE , was black."

Y / is used in certain idiomatic time and place expressions, to connect quantity expressions to the head word of the expression: "He left three HOURS AGO", "She lives three MILES FROM the station".

YP / connects plural noun forms ending in s to "’" in possessive constructions: "The STUDENTS ’ rooms are large".

YS / connects nouns to the possessive suffix "’s": "JOHN ’S dog is black". BOTH ABOVE LIKE PREP.

Z \ connects the preposition "as" to certain verbs: "AS we EXPECTED, he was late".

Appendix B

List of Bilingual Datasets

In this appendix, two bilingual datasets is given. The first dataset contains 30 sentences from 150 bilingual English-Indonesian sentences which are used in the development of the transfer rules. The second dataset contains another 30 sentences from 150 bilingual English-Indonesian sentences as a testing data.

B.1 Bilingual English-Indonesian Dataset Used for Transfer Rules Development

The English sentences were taken randomly from a random 30 English story books, while the Indonesian sentences were the manual translation.

Table B.1: Example data used for transfer rules development

No	English sentences	Indonesian sentences
1	He told us that he knew Grandpa.	Dia memberitahu kita bahwa dia mengenal kakek.
2	Noddy didn’t like Martha’s tail at all, it kept tickling him.	Noddy tidak menyukai ekor Martha sama sekali, itu selalu menggelitik dia.
3	Back in changing room, Coach Ken passed oranges to the players.	Kembali dalam ruang ganti, Pelatih Ken memberikan jeruk-jeruk ke pemain-pemain itu.
4	He doesn’t have a collar or a lead.	Dia tidak mempunyai kalung atau tanda.
5	Mr Sparks was in a very lively mood that day and he couldn’t stop talking.	Tuan Sparks berada dalam kesenangan sangat tinggi hari itu dan dia tidak bisa berhenti berbicara.
6	Where can they have gone?	Di mana mereka sudah bisa pergi?
7	The mouse promised to help the lion if he was in trouble.	Tikus itu berjanji untuk membantu singa itu jika dia berada dalam kesulitan.
8	That night, Bala went to look for the giant.	Malam itu, Bala pergi untuk mencari raksasa itu.
9	Once there was a pretty princess.	Dahulu kala ada puteri cantik.

No	English sentences	Indonesian sentences
10	“We have to get the dolphin out of the net.”	“Kita harus mengeluarkan lumba-lumba itu dari jaring itu.”
11	Dan dribbled and passed to Fabio.	Dan menggiring dan mengoper ke Fabio.
12	He even takes himself for walkies.	Dia bahkan membawa dia sendiri untuk jalan-jalan.
13	Let’s clean it before we light it.	Marilah membersihkannya sebelum kita menyalakannya.
14	Smutty and Primrose ran off with the dog biscuits, leaving Horace to look after Mark who was feeling rather dizzy.	Smutty dan Pimrose melarikan diri beserta biskuit anjing itu, meninggalkan Horace untuk memelihara Mark yang sedang merasa agak pusing.
15	“You can’t sail without sails, Walter.”	“Kamu tidak bisa berlayar tanpa layar, Walter.”
16	The old man is very angry.	Lelaki tua itu sangat marah.
17	The prince and princess were very happy to be together again.	Pangeran dan sang puteri sangat bahagia karena bersama lagi.
18	The next thing he knew, the genie turned him into one!	Hal berikutnya yang dia tahu, jin itu mengubah dia menjadi satu!
19	What makes Gromit so special?	Apa yang membuat Gromit sangat special?
20	“Where is my princess?”	“Di mana sang puteri saya?”
21	We thought it would be much more fun for you and Scruffy to chase a balloon instead of us cats.	Kita berpikir hal itu akan menjadi lebih menyenangkan untuk kamu dan Scruffy saat mengejar balon dibandingkan kucing-kucing seperti kita.
22	Walter, the Walter taxi, was a little boat.	Walter, taksi Walter itu, adalah perahu kecil.
23	A feast appeared, just as the emperor’s parade passed by.	Pesta berlangsung, begitu parade kaisar itu singgah.
24	The fairy told him what to do.	Peri itu memberitahu dia apa yang dikerjakan.
25	Fabio was very excited.	Fabio menjadi sangat tersemangati.
26	Snatch was soon wide awake and ready to find his friends for a game of chase, but they had seen him coming and had gone to hide again.	Snatch segera beranjak bangun dan siap untuk menemukan teman-teman dia dalam permainan kejar-kejaran, tetapi mereka sudah melihat dia datang dan sudah pergi untuk bersembunyi lagi.
27	He stayed in a little hut.	Dia tinggal dalam pondok kecil.
28	It was half-time and the score was one-all!	Saat itu separuh-waktu dan skornya adalah satu-semua!
29	The frog went out to search for the gold chain.	Katak itu pergi keluar untuk mencari rantai emas itu.
30	She was jealous.	Dia cemburu.

B.2 Bilingual English-Indonesian Dataset Used for Evaluation and Comparison

Other set of English sentences were taken randomly from a random 30 English story books. The Indonesian sentences were one of four reference translations.

Table B.2: Example of the testing data

No	English sentences	Indonesian sentences
1	“I am sure my day can only get better”, Noddy thought to himself.	“Saya yakin hari ini pasti akan membaik, pikir Noddy.”
2	The king laughed when he heard the fisherman’s story.	Raja tertawa ketika mendengar cerita si nelayan.
3	It swam into a net.	Itu berenang ke dalam jaring.
4	The Littleville striker had scored a goal, too!	“Striker” Littleville itu sudah mencetak gol juga!
5	“You must pick the right flower and take it home, or she will stay with me forever.”	“Kamu harus memetik bunga yang tepat dan membawanya pulang, kalau tidak dia akan tinggal dengan saya selamanya.”
6	The next morning, Abdul was surprised.	Pada keesokan harinya, Abdul terkejut.
7	“Please, keep your tail on your side of the car”, Noddy said.	“Silahkan, atur ekormu pada sisimu di mobil ini”, kata Noddy.
8	Footballer Fabio held it high above his head and everyone cheered.	Pemain bola Fabio menahannya di atas kepalanya dan semua orang bersorak.
9	“This balloon is for you, Snatch.”	“Balon ini untukmu, Snatch.”
10	But just like every other dog, the most special thing about Gromit is his soft, furry ears!	Tetapi seperti setiap anjing yang lain, hal yang paling spesial tentang Gromit adalah telinganya yang lembut dan berbulu!
11	Aladdin could hardly believe it.	Aladdin hampir tidak dapat mempercayainya.
12	However, he thanked the priest and went home.	Walaupun demikian, dia berterimakasih kepada pendeta itu dan kembali kerumah.
13	The king ordered his men to look for the gold chain.	Sang raja meminta orang-orangnya untuk mencari rantai emas itu.
14	A little boat can get the dolphin out.	Sebuah perahu kecil bisa mengeluarkan lumba-lumba itu.
15	"Has Jack been there?"	"sudahkah Jack di sana ?"
16	Suddenly, we saw an old man.	Tiba-tiba, kami melihat seorang lelaki tua.
17	“Jump in, Bumpy Dog”, called Noddy.	“Lompat kemari, Bumpy Dog,” panggil Noddy.
18	“I want to sail like you”, Walter said to the big sailboat.	“Saya ingin berlayar sepertimu”, kata Walter pada perahu layar yang besar itu.
19	She put him on her bed.	Dia meletakkan orang itu di dipannya.
20	They came to a cliff very close to the sea.	Mereka sampai ke tebing yang sangat dekat dengan laut itu.
21	The magician was clever, but so was the princess.	Pesulap itu pandai, begitu juga dengan sang puteri.
22	Doctor Daisy rushed out to help Johnny.	Dokter Daisy segera menolong Johnny.
23	After two days, the bear goes back to the forest.	Setelah dua hari, beruang itu kembali ke hutan.

No	English sentences	Indonesian sentences
24	The mouse heard the lion’s cry.	Tikus itu mendengar tangisan singa.
25	“You are too little to carry lots of people”, said the ferryboat.	“Kamu terlalu kecil untuk mengangkut banyak orang,” kata perahu feri itu.
26	The frog was very sad.	Katak itu sangat sedih.
27	“Are you looking for Noddy?” Mr Plod asked Big-Ears.	“Apakah kamu sedang mencari Noddy?” Tuan Plod bertanya pada Big-Ears.
28	There were many flowers in the woods.	Ada banyak bunga di pepohonan itu.
29	Scruffy was by now rather bored with being dressed up as a birthday present so he climbed out of the box.	Scruffy mulai bosan dengan dirias sebagai kado ulang tahun sehingga dia memanjat keluar dari kotak itu.
30	When the magician heard about Aladdin’s good fortune, he was furious.	Ketika penyihir itu mendengar tentang keberuntungan Aladdin, dia menjadi sangat marah.

Appendix C

Translation Results

The English sentences of Table C.1 are the testing data taken from Table B.2. The Indonesian sentences are the translations by P1, P2, P3, and P4 respectively from up to down. P1 is Kataku^{TM}, P2 is Rekso Translator, P3 is Google Translate Beta Version, and P4 is ADJ-based MT system.

Table C.1: Example of translation results of four tested MT systems

No	English sentences	Program	Indonesian sentences
1	“I am sure my day can only get better," Noddy thought to himself.	P1	"Saya yakin hari saya hanya bisa jadi lebih baik," pikir Noddy kepada sendiri.
		P2	"- Aku merasa pasti hari ku hanya dapat mendapat lebih baik", Orang bodoh berpikir kepada dirinya.
		P3	"Saya yakin hari saya hanya bisa mendapatkan lebih baik", Noddy berpikir untuk dirinya sendiri."
		P4	“SAYA yakin hari saya bisa hanya menjadi lebih baik” , Noddy berpikir untuk dia sendiri .
2	The king laughed when he heard the fisherman’s story.	P1	Raja tertawa ketika dia mendengar cerita nelayan.
		P2	Raja tertawa[kan ketika ia mendengar kisah nelayan itu.
		P3	Raja tertawa ketika dia mendengar cerita dari nelayan.
		P4	raja itu tertawa saat dia mendengar cerita nelayan itu .
3	It swam into a net.	P1	Berenang ke dalam sebidang jala.
		P2	Itu berenang ke dalam suatu jaring.
		P3	It swam menjadi bersih.
		P4	Itu berenang ke dalam jaring .
4	The Littleville striker had scored a goal, too!	P1	Pemogok Littleville sudah memasukkan gol, juga!
		P2	striker Littleville telah mencetak gol, juga!
		P3	Yang telah dinilai Littleville pemogok tujuan, too!
		P4	striker Littleville itu sudah mencetak goal , juga !

No	English sentences	Program	Indonesian sentences
5	“You must pick the right flower and take it home, or she will stay with me forever.”	P1	“Anda harus memetik bunga dan pengambilan benar itu rumah, atau dia akan tinggal dengan saya selama-lamanya.
		P2	"- Anda harus memungut bunga yang benar dan mengambilnya rumah, atau dia akan tinggal dengan aku selamanya."
		P3	"Anda harus memilih bunga yang tepat dan bawa pulang, atau ia akan tinggal dengan saya selama-lamanya."
		P4	"Kamu harus memetik bunga benar itu dan membawa pulang itu , atau dia akan tinggal dengan saya selamanya" .
6	The next morning, Abdul was surprised.	P1	Keesokan paginya, Abdul heran.
		P2	Besoknya, Abdul dikejutkan.
		P3	Pagi berikutnya, Abdul yang terkejut.
		P4	berikutnya itu pagi , Abdul terkejut .
7	“Please, keep your tail on your side of the car”, Noddy said.	P1	"Silahkan, menyimpan ekor anda di pihak anda mobil," kata Noddy.
		P2	"-tolong, menyimpan(pelihara ekor mu di sisi mu dari mobil", Orang bodoh berkata.
		P3	"Silakan, Anda tetap ekor di samping mobil", kata Noddy.
		P4	"Tolong , jaga ekor kamu pada sisi kamu dari mobil itu" , Noddy berkata .
8	Footballer Fabio held it high above his head and everyone cheered.	P1	Footballer Fabio memegangnya tinggi di atas kepalanya dan tiap orang menyoraki.
		P2	Sepak Bola Fabio [mengadakan;memegang] nya ketinggian di atas kepala dan setiap orang nya bersorak.
		P3	Footballer Fabio diadakan itu tinggi di atas kepalanya dan semua orang cheered.
		P4	Pemain bola Fabio menahan itu tinggi di atas kepala dia dan setiap orang bersorak.
9	“This balloon is for you, Snatch.”	P1	"Balon ini bagi anda, Snatch."
		P2	"- Balon ini adalah untuk anda, Rengutan."
		P3	"Ini adalah untuk Anda balon, menggigit."
		P4	"balloon ini untuk kamu , Snatch."
10	But just like every other dog, the most special thing about Gromit is his soft, furry ears!	P1	Tetapi tepat seperti setiap anjing lain, hal yang paling istimewa di sekitar Gromit adalah telinganya yang berbulu lembut yang halus!
		P2	Hanya seperti semua anjing yang lain, hal paling khusus tentang Gromit adalah lembut nya, telinga-telinga berbulu lembut!
		P3	Tetapi seperti setiap anjing, yang paling khusus tentang hal Gromit adalah lembut, berbulu telinga!
		P4	Tetapi tepat seperti setiap yang lain anjing, benda paling special itu tentang Gromit

No	English sentences	Program	Indonesian sentences
			adalah lunak, telinga-telinga berambut lebat dia !
11	Aladdin could hardly believe it.	P1	Aladdin hampir tidak bisa percaya kepadanya.
		P2	Aladdin bisa dengan susah percaya nya.
		P3	Aladdin hampir dapat percaya.
		P4	Aladdin bisa dengan susah mempercayai itu.
12	However, he thanked the priest and went home.	P1	Tetapi, dia berterima kasih kepada pendeta dan pulang.
		P2	Bagaimanapun, ia berterimakasih imam dan pergi rumah.
		P3	Namun, ia mengucapkan terima kasih imam dan pulang.
		P4	Bagaimanapun, dia berterimakasih kepada pendeta itu dan pergi pulang .
13	The king ordered his men to look for the gold chain.	P1	Raja menyuruh laki-lakinya mencari rantai emas.
		P2	Raja memerintahkan (memesan orang nya untuk mencari rantai emas.
		P3	Raja memerintahkan orang-orangnya untuk mencari emas rantai.
		P4	raja itu memerintahkan orang-orang dia mencari rantai emas itu .
14	A little boat can get the dolphin out.	P1	Sekapal kapal kecil bisa mengeluarkan ikan lumba-lumba.
		P2	Suatu perahu yang kecil dapat mendapat dolfin ke luar.
		P3	Sebuah perahu kecil yang dapat dolphin keluar.
		P4	boat kecil bisa mengeluarkan lumba-lumba itu .
15	"Has Jack been there?"	P1	Apakah "Jack di sana?"
		P2	"- Mempunyai Dongkrak di sana?"
		P3	"Apakah Jack telah ada?"
		P4	"sudahkah Jack di sana ?"
16	Suddenly, we saw an old man.	P1	Tiba-tiba, kami melihat seorang laki-laki tua.
		P2	Tiba-tiba, kita melihat satu ayah/suami.
		P3	Tiba-tiba, kami melihat seorang laki-laki tua.
		P4	Dengan tiba-tiba, kita melihat lelaki tua.
17	“Jump in, Bumpy Dog”, called Noddy.	P1	"Terjun, Bumpy Dog, yang" dinamai" Noddy.
		P2	"- Sela, Anjing Tidak Rata", memanggil (hubungi Noddy.
		P3	"Langsung di, Bumpy Dog", bernama Noddy.
		P4	"lompat ke dalam , Bumpy Anjing , memanggil Noddy .

No	English sentences	Program	Indonesian sentences
18	"I want to sail like you," Walter said to the big sailboat.	P1	"Saya mau berlayar seperti anda," kata Walter ke perahu layar besar.
		P2	"- Aku ingin berlayar seperti anda", Walter berkata kepada perahu layar yang besar.
		P3	"Saya ingin berlayar seperti Anda", kata Walter ke layar besar.
		P4	"Saya ingin berlayar seperti kamu," Walter berkata ke perahu layar besar itu.
19	She put him on her bed.	P1	Dia menaruhnya di atas tempat tidurnya.
		P2	Dia menaruh dia di tempat tidur nya.
		P3	Dia menempatkan dia di tempat tidur dia.
		P4	Dia meletakkan dia pada tempat tidur dia.
20	They came to a cliff very close to the sea.	P1	Mereka datang ke jurang sangat dekat laut.
		P2	Mereka datang ke suatu karang sangat dekat dengan laut.
		P3	Mereka datang ke jurang yang sangat dekat dengan laut.
		P4	Mereka datang ke tebing sangat dekat dengan laut itu.
21	The magician was clever, but so was the princess.	P1	Tukang sihir pandai, tetapi oleh sebab itu adalah puteri.
		P2	Tukang sihir itu pandai, tetapi maka adalah puteri.
		P3	The magician telah pandai, tetapi yang jadi princess.
		P4	pesulap itu pandai , tetapi begitu juga ratu itu .
22	Doctor Daisy rushed out to help Johnny.	P1	Dokter Daisy buru-buru keluar untuk menolong Johnny.
		P2	Bunga aster Dokter buru-buru ke luar untuk menolong Johnny.
		P3	Dokter Daisy rushed out untuk membantu Johnny.
		P4	Doctor Daisy buru-buru keluar untuk menolong Johnny.
23	After two days, the bear goes back to the forest.	P1	Sesudah dua hari, beruang kembali ke hutan.
		P2	Setelah dua hari, beruang kembali ke hutan.
		P3	Setelah dua hari, yang melahirkan akan kembali ke hutan.
		P4	Setelah dua hari, beruang itu pergi kembali ke hutan itu.
24	The mouse heard the lion’s cry.	P1	Tikus mendengar jeritan singa.
		P2	Tikusan mendengar tangis singa itu.
		P3	Mouse mendengar teriakan dari singa.
		P4	tikus itu mendengar tangisan singa itu .
25	"You are too little to carry lots of people," said the ferryboat.	P1	"Anda adalah terlalu sedikit untuk memajukan banyak orang," kata kapal feri.

No	English sentences	Program	Indonesian sentences
		P2	"- Anda adalah terlalu kecil untuk membawa banyak orang-orang", berkata kapal tambang.
		P3	"Anda terlalu kecil untuk membawa banyak orang", kata tambangan.
		P4	"Kamu terlalu kecil untuk membawa banyak dari orang-orang," berkata perahu feri itu.
26	The frog was very sad.	P1	Katak sangat menyedihkan.
		P2	Kodok itu adalah sangat sedih.
		P3	Katak yang sangat sedih.
		P4	katak itu sangat sedih .
27	“Are you looking for Noddy?” Mr Plod asked Big-Ears.	P1	Apakah "anda mencari Noddy?" Mr Plod bertanya Big-Ears.
		P2	"- Adalah anda mencari Noddy?" Mr Plod minta (tanya Big-Ears.
		P3	"Apakah Anda mencari Noddy?" Mr mengarungi ditanyakan Big-Ears.
		P4	"apakah kamu mencari Noddy ?" Tuan Plod bertanya Big-ears- .
28	There were many flowers in the woods.	P1	Ada banyak bunga di hutan.
		P2	Ada banyak berbunga di dalam hutan.
		P3	Ada banyak bunga di dalam hutan.
		P4	Di sana banyak bunga-bunga dalam hutan itu.
29	Scruffy was by now rather bored with being dressed up as a birthday present so he climbed out of the box.	P1	Scruffy oleh sekarang agak bosan dengan dipakaikan sebagai hadiah hari ulang tahun oleh sebab itu dia merayap keluar kotak.
		P2	Scruffy sekarang juga agak bosan dengan mahluk berdandan sebagai suatu harilahir menyajikan maka ia memanjat karena kotak.
		P3	Scruffy itu oleh sekarang bukan sedang bosan dengan berpakaian sebagai hadiah ulang tahun, jadi dia naik dari kotak.
		P4	Scruffy juga oleh sekarang agak bosan dengan dirias sebagai hadiah hari ulang tahun maka dia memanjat keluar dari kotak itu.
30	When the magician heard about Aladdin’s good fortune, he was furious.	P1	Ketika tukang sihir mendengar tentang keberuntungan Aladdin, dia geram.
		P2	Ketika tukang sihir mendengar tentang untung/kekayaan baik Aladdin, ia sangat marah/hebat.
		P3	Ketika mendengar tentang magician Aladdin dari pukulan, dia sangat marah.
		P4	Ketika pesulap itu mendengar tentang keberuntungan baik Aladdin, dia sangat marah.