Probabilistic Parsing for Indonesian Language

Rosa A. Sukamto, Dwi H. Widyantoro
School of Electrical Engineering & Informatics
Institute of Technology Bandung, Bandung INDONESIA
rosa_if_itb_01@yahoo.com, dwi@if.itb.ac.id

Abstract

This paper reports our work in using Collins' parser for parsing Indonesian language. Our approach is to adapt all input files, which were originally designed for English, so as to suit the parsing requirement for Indonesian. These include lexicon files, grammar and event files, as well as corpus files. Our preliminary experiment indicates that Collins' parse is able to parse Indonesian. One of the main obstacles in this attempt is providing treebank needed to calculate probability values. To overcome this problem we could translate the existing English treebank into Indonesian, or initiate a call for collaborative work to manually build and maintain a treebank for Indonesian language.

1. Introduction

Parse trees are trees representing the syntactic structure of a sentence with respect to a formal grammar. Parse trees have many useful applications. Some well known examples include grammar checking in word-processing systems, machine translation, question answering, information extraction, lexicography applications, and speech recognizers.

Parse trees can be automatically generated by a parser. Given a grammatically correct sentence, a parser will generate the corresponding parse tree. Many researchers have developed various parsing algorithms using various approaches. The search strategy can be goal-directed search (top-down) or data-directed search (bottom-up). A top-down parser searches a parse tree by starting from the root and growing the tree down to the leaves (i.e., words in sentence). Conversely, bottom-up parser starts from words and tries to grow the tree up to root.

The most recent advances in the field is probabilistic parsing. This approach builds probabilistic models of syntactic information and uses this probabilistic information to efficiently search for a parse tree. Probabilistic parsing offers efficient disambiguation (i.e., simply selecting the most-probable interpretation). Its important role in parsing and natural language understanding is unquestionable.

Collins' parser [6] is among the most popular probabilistic parsers to date. Although very useful, all available tools and resources have been designed and crafted for English. To our knowledge, no attempt has been made to use Collins parser to build parse trees for Indonesian. This paper reports our work to address the above issue.

The rest of the paper is organized as follows. In Section 2 we describe related work in probabilistic parsing. Section 3 provides an overview of Collins' parser. Next in Chapter 4, we describe steps to modify Collins parser for Indonesian language. Section 5 describes our preliminary experiment. Section 6 discusses issues in our work and suggests some alternative ways out, followed by conclusions in Section 7.

2. Related Work

Research in probabilistic parsing originates from the work by Schabes and Water who discussed Stochastic Lexicalized Context-Free Grammar (SLCFG) [11], also known as Probabilistic Lexicalized Context-Free Grammar (PLCFG), which is a model derivation of Probabilistic Context-Free Grammar (PCFG). Glen Carrol developed SINGER (Single Reader) that used rules as input and employed PCFG to generate additional new rules. Mark Johnson conducted similar work on PCFG and concluded that the performance of PCFG is good in most cases.

Charniak [3,4] built a bottom-up parser for English and used a *treebank* (a collection of annotated sentences) for calculating the probability of a parsed sentence. Later Charniak developed a top-down parser but employing Maximum Entropy and Penn *Treebank* [5], similar to using a decision tree. Collins built a statistical parser by calculating word dependency using bigram, and later developed head-driven based parser. Bikel developed statistical parsing model (parser framework) that take advantage of lexical parameters [2].

Aziz *at* al. attempted to parse Malay Language using CFG production rules [1]. Although looks very similar, Indonesian language and Malay Language differ in some respects that make their work cannot be directly applicable to Indonesian language. Lefuel and Ross proposed a hybrid parser, using both probabilistic parsing and genetic algorithm [9]. A more thorough discussion about probabilistic parsing is given by Jurafsky and Martin.

Similar work is also provided by Gusmita and Manurung [8]. They built the probabilistic models for Indonesian probabilistic parser using parse trees generated by PC-PATR parser. Our current work is much more similar to that of Collins, Hajic, Ramshaw and Tillmann [7]. They adapted Collins' parser to Czech from English. In our case, we adapted Collins' parser for Indonesian language.

3. Collins' Parser

Collins introduced three probabilistic parsing models. In the first model, PCFG has the following production rule:

$$P(h) \rightarrow L_n(l_n)...L_1(l_1)H(h)R_1(r_1)...R_m(r_m)$$

where H is the *head-child* from rule P (right hand side of production rule), whereas $L_n(l_n)...L_1(l_1)$ and $R_1(r_1)...R_m(r_m)$ are the left and right hand side of H, respectively. To avoid domination of either part of production rule, Collins incorporate distance parameter that consider the position of terminal & non terminal symbols at the right hand side.

The second model extends the first model so that it can make distinction between complements and adjuncts. This model also has parameters corresponding directly to probability distributions over sub-categorization of head-words. The third model is a further extension that provides probability treatment to extract relative clauses. How to calculate all these probabilities and other details can be found in [6].

Michael Collins' parser uses a PCFG to model grammar and employs a bottom-up chart parsing algorithm. Most of Collins' parser modules are for learning process. To create a language model, the system needs the following file as input:

1. Event file for storing heuristically generated event (probability of sentence elements dependency) from Penn WSJ (*Wall Street Journal*) *Treebank* using Collins' format. This file will be used for calculating the probability of grammars occurrence and dependency.
2. A corpus containing annotated sentences to be parsed.
3. File containing target language grammar for parsing references.
4. A file containing non terminal symbols.
5. A lexicon to check parts-of-speech tag.

4. Adapting Collins' Parser for Indonesian language

In this section we describe some modification needed in order to use Collins' parser for Indonesian language. Most of them relate to the adaptation of the input files. The main algorithm for training probability values and creating a parse tree remains unchanged.

4.1 Event File

Collins generates event file from sections in Penn WSJ *treebank* [10]. This part is the biggest challenge because currently we do not have a *treebank* using Indonesian language. We will discuss this obstacle in Section 6. The following is an example of event file containing Indonesian.

```
6 4 Yohanes NN memukul VB Bill NN . PU
3 memukul VB S NP 00000 00000
2 #STOP# #STOP# memukul VB #STOP# S NP
000000 110 0 0
2 #STOP# #STOP# memukul VB #STOP# S NP
000000 010 0 0
3 memukul VB NP VB 00000 00000
2 Yohanes NN memukul VB NN NP VB 000000 110
0 0
2 #STOP# #STOP# memukul VB #STOP# NP VB
000000 100 0 0
2 Bill NN memukul VB NN NP VB 000000 010 0
0
2 . PU memukul VB PU NP VB 000000 000 0 0
2 #STOP# #STOP# memukul VB #STOP# NP VB
000000 000 0 0
```

4.2 Corpus File

The sentence to be parsed needs to be annotated so that each word in the sentence must be tagged with part of speech. The corpus file has the following format:

```
#num_of_words word#1 tag#1 word#2 tag#2 …
```

For example,

```
4 Yohanes NN memukul VB Bill NN . PU
            (hits)
```

Figure 1 depicts the steps for the post tagging process, which could go through multiple stages. First, the word class will be determined using a dictionary. If the word is not found, then perform morphology analysis. If the tag is still unknown then predict the word parts-of-speech through bigram analysis using grammar rules. For example, in phrase "*sedang menggambar*" where we know that the POS tag of *sedang* is RB but do not know the POS tag of *menggambar* (drawing). If the grammar rule VP → RB VB happens to have high probability of occurrence then we can predict that the POS tag of *menggambar* is VB.

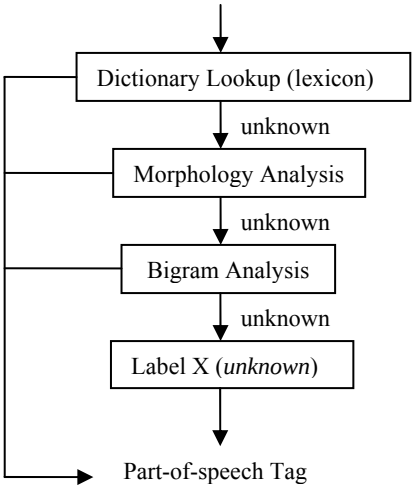

Figure 1. POS *Tagging*

Indonesian has morphological rules that can be used to predict the word class. For example, Table 1 provides affixes that would affect the part of speech tag. Predicting the part of speech using morphology rule is performed in the following order:

1. Numeral checking: if the word is a number (all are digit)
2. Abbreviation checking: if all letters in the word are capital then it must be a noun.
3. Prefix checking: for example, *menari* (to dance) is derived form of *tari* (dance) with prefix *meN*, so it will be predicted as a verb,
4. Suffix checking: for example, *terangi* (lighten up) is derived from *terang* (light) with suffix *i*, therefore, it must be a verb
5. Prefix and Suffix checking, examine if the word is a derived using prefix and suffix.
6. Repeat checking: for example, *buku-buku* (books) will have the same tag as its root word *buku* (book)
7. Name checking: a word with capital in the first letter indicates a name.

We use a total 33 prefix, 29 suffix and 17 (prefix & suffix) rules for morphology analysis.

Table 1. Example of Affixes and its Part of Speech

Affixes Pattern	Example	Part of Speech
meN + root word + *kan*	*mengajukan* (propose)	Verb
peN-ber + root word + *an*	*pelukis*, *pemburu*	Noun
ke-ber- + root word + *an*	*kebersamaan* (togetherness)	Noun
root word + *i*	*Terangi* (lighten up)	Verb
beR- + root word	*Bekerja* (work)	Verb
ter- + root word	*Tertidur* (fall a sleep)	Verb

4.3 Grammar dan Non-Terminal Symbols

The grammar file is generated from the *treebank*. Like the event file, this is also a problem. Grammar of Indonesian is similar to that of English in terms of the general structural patterns like subject-predicate-object although there are still some differences here and there. For example, Indonesian sentences do not recognize English tenses (i.e., all verbs have the same forms regardless of time of occurrences). Additionally, Indonesian has a DM (*Diterangkan*--word to modify – *Menerangkan*--modifier) pattern, for example *buku* (book) *biru* (blue), while english has the MD (opposite) pattern. Moreover, Indonesian noun does not distinguish between plural and singular.

We preserve Collins' parser grammars that match Indonesian grammar. A simple example of Indonesian grammar is as follows:

S → NP VP NN

NP → NN JJ (e.g., *anak kecil*, little kid)
VP → RB VB (e.g., *sedang menulis*, writing)

Several non terminal symbols from Collins' parser also need modification for Indonesian. Tables 2 and 3 provide the list of non terminal symbols needed for Indonesian.

Not all tags in Collins' parser are applicable in Indonesian. For example, tags such as NNP, NNPS, and NNS are used for plural noun in Collins' parser but Indonesian does not distinguish between plural and singular noun; i.e., all noun will be tagged as NN. Indonesian also does not recognize determiners such as "the" or "a", and various forms of time-dependent verb such as VBD, VBG, VBN, VBP, VBZ. All kinds of verb in Indonesian are tagged as VB.

Table 2. Partial List of Non-terminal Word Class

Simbol	Part of Speech	Example
JJ	Adjective	*cantik* (beautiful)
RB	Adverb	*sedang*, *nanti* (later), *sekarang* (now)
AR	Artikula	*si*, *sang*
CC	Coordinate Conjunction	*dan* (and), *lalu* (then)
CS	Subordinate Conjunction	*ketika* (when), *walaupun* (although)
PR	Pronoun	*saya* (I), *itu* (that)
WH	Question Word	*Siapakah* (who)
NN	Noun	*meja* (desk)
CD	Numeral	*seribu* (thousand)
IN	Preposition	*di* (at), *ke* (to), *dari* (from)
UH	Interjection	*ai, aah, ceile*
RP	Particle	*pun, per*
VB	Verb	*melempar* (throw)
MD	Modal	*boleh*
FW	Foreign Word	download, notebook
SYM	Symbol	+.%,$, #
PU	Punctuation	., , , :, (,), “, ‘, ”, ’
X	*unknown*	

Table 3. List of Non-terminal Phrase or Relative Clause

Simbol	Description
S	Sentence
ADJP	Adjective Phrase
ADVP	Adverb Phrase
NP	Noun Phrase
SBAR	Relative Clause
SBARQ	Relative Clause after question word
VP	Verb Phrase

4.4 Lexicon File

Collins' parser uses lexicon file as vocabulary (at least contains word and its part of speech). The content of this file, therefore, must support the Indonesian. Fortunately, there is KEBI (Electronic Indonesian-English Dictionary) that is available for free for research purpose. KEBI was developed by BPPT (Agency for Assessment and Application of Technology) and it contains about 29.396 entries.

KEBI recognizes only fifteen parts of speech: adjective, adverb, determiner, article, auxiliary, conjunction, interjection, noun, numeral, ordinal, particle, phatic, preposition, pronoun and verb. KEBI's format must first be converted into Collins' parser lexicon file format.

5. Preliminary Experiment

We have performed a preliminary experiment to see whether the Collins' parser is able to parse Indonesian. For this experiment, we manually build two very small Indonesian treebanks. The first treebank contains 45 parse trees of simple sentences. Using this treebank, Collin's parser is able to correctly parse new six out of seven simple sentences. Figure 2 depicts one of the correctly parsed tree.

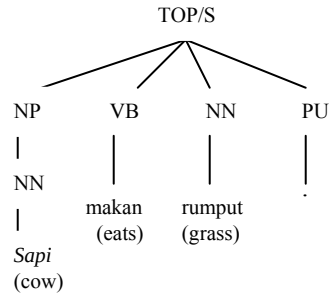

Figure 2. Simple sentence parse tree

The second *treebank* has 190 parse trees extracted from complex sentences. For the more complex *treebank* we provide 15 new complex sentences to parse. Collin's parser is able to parse only 8 sentences but fails to parse the other seven complex sentences. A quick observation reveals that Collins' parser failure to parse those sentences is due to the fact that parts of input sentences exhibit grammar rules unrecognizable by Collins' parser. In the eight sentences that have been successfully parsed, none of them is perfect. For example, given the complex sentence as follows:

Tema cerita Malin Kundang dari Sumatra Barat ini ternyata juga bisa ditemui di daerah lain di Indonesia.

(This story theme of Malin Kundang from West Sumatra can also be found in other regions of Indonesia)

Collins' parser output for the above sentence is depicted by Figure 3, which is incorrect. The correct parse tree is given by Figure 4.

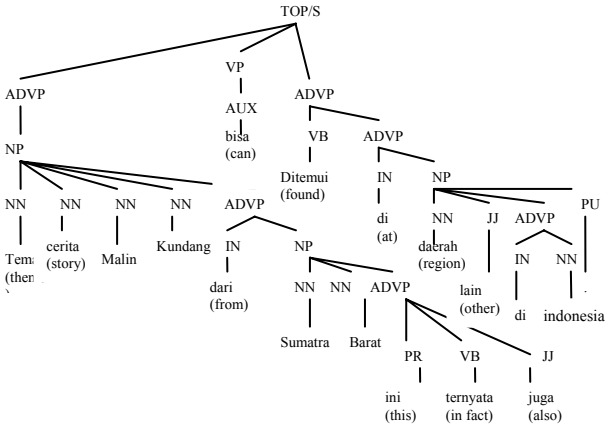

Figure 3. Collin's parser output for a more complex sentence.

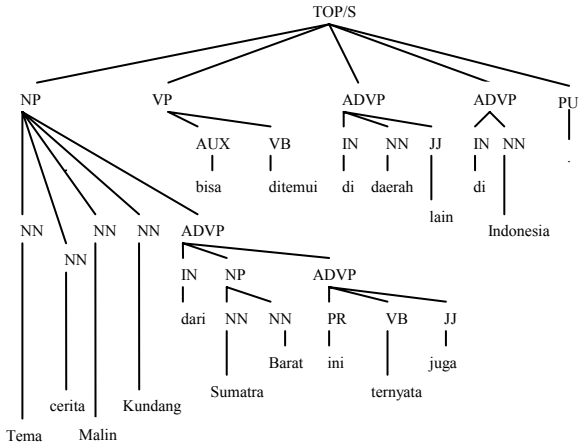

Figure 4. The correct parse tree.

6. Discussion

One of the main critical component as input is a *treebank* for calculating all probabilities. Unfortunately, this resource is currently not available for Indonesian. An attempt has been made to hand-craft a *treebank* with a tiny size (compared to Penn *treebank*). This is our biggest challenge; that is, creating a *treebank* that is large enough to provide reliable probability values. There are at least two alternatives for creating the needed *treebank* that we can think of.

The first alternative is to perform automatic language translation from Penn WSJ *treebank* into Indonesian language. This could be the easiest and most efficient way to obtain the Indonesian version of Penn WSJ *treebank*. Although the translation results might be inaccurate, the resulting probability values might still be reliable. Because there are some differences of grammar between English and Indonesian, the translation process should address this issue. For example, English part of speech that does not exist in Indonesian should be converted into its equivalent tag or just leave it blank. For example, a parse tree from the Penn *treebank*:

```
(S (NP-SBJ (NNP President)
           (NNP Reagan))
   (VP (VBD learned)
       (NP (DT that)
           (NN lesson)))
   (. .))
```

is translated into:

```
(S (NP (NN Presiden)
       (NN Reagan))
   (VP (VB belajar)
       (NP (NN pelajaran)
           (PR itu)))
   (. .))
```

Creating a *treebank* manually is an enormous effort that cannot be done individually. Because a *treebank* would be of useful to society nationally, it is also worth of effort to start a collaborative work among individual, community, organization and institution across the country to build a *treebank* for Indonesian language. As the first step, we are planning to set up a Web-based system for managing the creation and maintenance of *treebank*. The next step is to call for participation to contribute in this effort. Another possible action includes initiation a consortium comprising of various individual and institution interested in developing related and similar effort.

7. Conclusions

Theoretically speaking, it is possible to apply Collins' parser for Indonesian language provided that

all parser's inputs are appropriately adapted to conform to the target language. The attempt to apply probabilistic parsing for Indonesian has raised awareness and a need to have a *treebank* for Indonesian. Considering that statistical approach has gained much attention in natural language processing in general, other kind of Indonesian corpus might also be needed. Some one should initiate a national and regional collaborative work to create and maintain such corpora.

References

[1] Azis, Mohd Juzaiddin et al. (2006) Pola Grammar Technique for Grammatical Relation Extraction of Malay Language, Malaysian Journal of Computer Science, 19, 59-72

[2] Bikel, Daniel M. (2004) : *On The Parameter Space of Generative Lexicalized Statistical Parsing Models*, PhD dissertation, University of Pennsylvania. 1-20, 141-148

[3] Charniak, Eugene. (1993) : *Statistical Language Learning*, Massachusetts Institute of Technology.

[4] Charniak, Eugene. (1997) : Statistical Parsing with a Context-free Grammar and Word Statistics, *American Association for Artificial Intelligence*: *AAAI Press*. 1-6

[5] Charniak, Eugene. (2000) : A Maximum-Entropy-Inspired Parser, *Proceedings of NAACL-2000.* 132-139.

[6] Collins, Michael. (1999) : *Head-Driven Statistical Models for Natural Language Parsing*, Disertasi program Doctor of Philosophy, University of Pennsylvania. 1-265.

[7] Collins, Michael, Jan Hajic, Lance Ramshaw, Cristoph Tillmann (1999) : A Statistical Parser for Czech, *Proceedings of the 37th Annual Meeting of the ACL*.

[8] Gusmita, Ria Hari & Ruli Manurung (2008) Some initial experiments with Indonesian probabilistic parsing. Second MALINDO Workshop. 1-5.

[9] Lefuel, Ramon & Brian J. Ross (2004) Parsing Probabilistic Context Free Language with Multiple-Objective Genetic Algorithm, Technical Report, Brock University. 1-11.

[10] Marcus, Mitchell P. dkk (1992) : Building a Large Annotated Corpus of English: The Penn Treebank. Departmet of Computer and Information Science University of Pennsylvania.

[11] Schabes, Yves & Waters, Richard C (1993) Stochastic Lexicalized Context-Free Grammar, International Workshop on Parsing Technology. 1-10.