Some initial experiments with Indonesian probabilistic parsing

Ria Hari Gusmita
Faculty of Computer Science
Universitas Mercu Buana
ria_hg@yahoo.com

Ruli Manurung
Faculty of Computer Science
University of Indonesia
maruli@cs.ui.ac.id

Abstract

This paper presents initial experiments in constructing a probabilistic parser for Indonesian. Due to the unavailability of a large manually parsed corpus, we start from an existing symbolic parser [4] to parse a balanced collection of Indonesian sentences. A probabilistic CFG language model is extracted, ignoring explicit linguistic information encoded in feature structures, and is subsequently used to parse an unseen collection of sentences. The resulting parse trees are evaluated against the set of candidate parses returned by the symbolic parser. The initial results indicate that the PCFG is failing to accurately capture verb subcategorization information.

1. Introduction

Syntactic analysis is a crucial element of automatic processing of language, and as such, parsers are fundamental systems which enable better understanding of sentences in a particular language.

Previous work in parsing Indonesian has, to our knowledge, been limited to the traditional rationalist approach of handcrafted grammars that discretely partition sentences into valid and invalid ones.

Since the late 1980s, thanks to the availability of large collections of textual data and increasing computational power, statistical approaches to parsing have yielded robust and wide-coverage parsing, and brought about a rigid, well-defined engineering methodology to the research field.

In this paper we present an initial experiment in constructing probabilistic parsers for Indonesian.

In Section 2 we discuss existing work on parsing Indonesian, and in Section 3 we discuss some shortcomings of these approaches, namely the issue of *ambiguity*, and how they are addressed by probabilistic techniques. Section 4 describes our design for an initial experiment utilizing probabilistic parsing for Indonesian, and the results are presented and discussed in Section 5.

2. Parsing Indonesian

The first documented work on parsing Indonesian dates back to 1995 [13], in which an initial grammar for Indonesian was written for LL and LR-parsers, formalisms more commonly used for defining computer programming languages. As such, the grammar was not linguistically motivated. This work was continued in [2, 11], where context-free grammars were constructed based on prescriptive rules defined in the *Tata Baku Bahasa Indonesia*, or official grammar of Indonesian [1].

In [4], the previous work was extended with more sophisticated linguistic analysis, namely an attempt to account for subcategorization and selectional restrictions. The grammars utilized feature structures in the PATR-II formalism [14]. The symbolic grammar reported in this work forms the basis of the experiments presented in this paper.

However, these works by and large contained many linguistic infelicities, as the overall goal of the research seemed to be to achieve high accuracy on a given corpus, with little regard to true wide-coverage and linguistic generality.

3. Ambiguity and probabilistic parsing

Ambiguity is one of the main problems encountered in parsing natural language, and Indonesian is no exception. With respect to parsing, **structural ambiguity**, such as the attachment of adjunct phrases, becomes a problem. Consider the example sentence, taken from [4]:

(1)	*presiden*	*memegang*		*kekuasaan*		*tertinggi*	*atas*
	the president	holds		authority		highest	over
	angkatan	*darat*	,	*angkatan*	*laut*	*dan*	*angkatan*
	forces	ground	,	forces	sea	and	forces
	udara						
	air						

'The president holds the highest authority over the army, navy, and air force.'

Structural ambiguity arises when considering what constituent the prepositional phrase *"atas angkatan darat,*

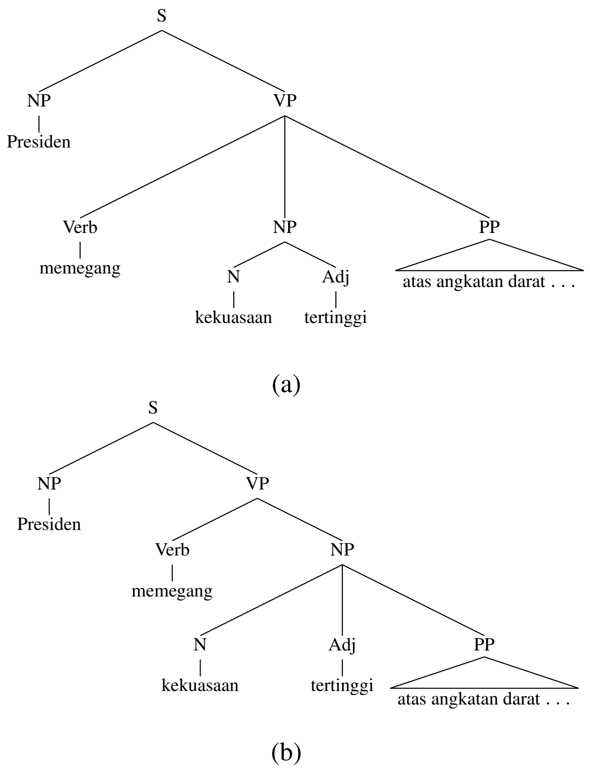

Figure 1. Alternative PP-attachment parses

angkatan laut, dan angkatan udara" attaches to, or modifies. On one hand, it could modify the verb phrase (as in, for example, *"presiden memegang kekuasaan tertinggi atas permintaan rakyat"*), yielding the parse tree in Figure 1(a). Alternatively, it could modify the object noun phrase, yielding the parse tree in Figure 1(b).

Most Indonesian speakers would have little difficulty in recognizing the latter analysis as being correct, but this requires contextual information usually inaccessible to a traditional parser. Indeed, the parser we adopted as our starting point [4] returns variants of the above trees as plausible parses, but makes no attempt to determine which one is correct.

Probabilistic syntax models provide a finer-grained explanation than the binary distinction of valid and invalid sentences afforded by standard CFGs. As a result, probabilistic models can thus be used to **disambiguate** between alternative parse trees.

A well-known formalism for constructing probabilistic syntax models is the probabilistic context-free grammar, or PCFG [6]. A PCFG is a 5-tuple $G = (N, \Sigma, P, S, D)$, consisting of

- a set of non-terminal symbols N
- a set of terminal symbols Σ (where $N \cap \Sigma = \emptyset$)
- a set of *production rules* P, each of the form $A \rightarrow \alpha$, where $A \in N$, and $\alpha \in (\Sigma \cup N)*$
- a *start symbol* $S \in N$
- a function D that states a probability [0,1] for each rule $A \rightarrow \alpha \in P$

The probability of parse tree T for sentence S is the product of probabilities of all rules r that are used to expand all nodes $n \in T$. More formally, $P(T, S) = \prod_{n \in T} p(r(n))$. Probabilistic parsing involves selecting the parse tree that maximizes this probability. Most traditional CFG parsing algorithms can be extended to achieve this. A well-known variant is the probabilistic CKY algorithm [3].

A PCFG model can be constructed from a corpus that has been manually parsed, sometimes referred to as a *treebank* [8]. Since such a resource already contains evidence of rule expansions, we simply count the number of occurrences a particular rule is applied, divided by the number of times the non-terminal symbol being expanded appears in the corpus. More formally, the probability of a rule can be computed as follows:

$$P(\alpha \rightarrow \beta|\alpha) = \frac{Count(\alpha \rightarrow \beta)}{\sum_{\gamma} Count(\alpha \rightarrow \gamma)} = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

Unfortunately, constructing a treebank is a costly (expert) labour-intensive task, and such a resource does not yet exist for Indonesian. One approach is to use a symbolic parser to parse all the sentences in a raw, unannotated corpus, and to count the rule expansions for all parse trees. This is the approach we use in this paper. Unfortunately this does make the assumption that all alternative parses are equally probable, when this is very likely not the case. A more sophisticated approach would be to apply the *inside-outside* algorithm, an instance of the Expectation Maximization algorithm for CFGs [7].

4. Experiment design

Due to the lack of a large manually parsed Indonesian corpus, we start from an existing symbolic parser developed by Joice [4] to parse a balanced collection of Indonesian sentences. Using the resulting parse trees, a PCFG-based stochastic language model is extracted. This language model deliberately ignores certain sophisticated linguistic details from Joice's output, e.g. subcategorization details contained within feature structures. The model is subsequently tested on an unseen collection of sentences, employing the probabilistic CKY algorithm. The results are evaluated against the parse trees returned by Joice's symbolic parser for those unseen sentences.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average (%)
Case 1	19	18	18	14	13	16.4 (6.56%)
Case 2	172	138	164	149	114	147.4 (58.96%)
Case 3	21	41	3	20	78	32.6 (13.04%)
Case 4	36	44	64	66	39	49.8 (19.92%)
Case 5	2	9	1	1	6	3.8 (1.52%)

Table 1. Results of parsing experiment

The purpose of this experiment is to investigate how well the PCFG formalism is able to implicitly represent explicit linguistic knowledge encoded as feature structures in the original symbolic grammar. Joice's grammar heavily relies on subcategorization information in the feature structures to rule out incorrect parses. As such, the CFG rewrite rules overgenerate considerably. If the probabilistic parser is able to select parse trees suggested by Joice's constraint-based parser without any access to feature structures, it would indicate that the statistical model indeed implicitly encodes some of the more sophisticated linguistic details.

4.1. Resources used

1. **Input sentences.** We used 1250 sentences reported in [4]. These sentences form a fairly balanced collection of Indonesian sentences, taken from various online media resources, with domains including (a) academic papers in the fields of psychology, agriculture, and science, (b) online entertainment news articles, (c) official governmental regulations.

2. **Symbolic grammar and lexicon.** We used the syntactic and lexical rules defined in [4]. However, this resource is not without linguistic infelicities. The syntactic rules overgenerate considerably, especially without the subcategorization information, and a number of the syntactic rules are not well-founded. Moreover, many proper nouns and idiomatic phrases are also listed as lexical items, e.g. *"lembaran negara republik indonesia tahun 2001 nomor 30"*. Such phrases were taken directly from the above corpus, as the goal of the previous research seemed to be to simply achieve high accuracy on the given corpus.

3. **Software tools**. To use the above grammar, which is written in the PATR-II formalism [14], we used the PC-PATR application, a freely available parser [9]. For parsing with the PCFG, we used an implementation of the probabilistic CKY algorithm used in [3][1]. The PCFG learning program was written in Perl, as were the training and testing scripts.

4.2. Experimental procedure

Firstly, we parsed the entire corpus using Joice's grammar and lexicon in PC-PATR. The results were analysed by our PCFG learning program, but the learnt grammar rules were discarded, leaving just the lexical rules. This was necessary so that during the testing stage, the probabilistic parser would be able to recognize the lexical items.

We then performed five-fold cross-validation as follows: the corpus was divided into 5 folds, each consisting of 250 sentences. For each iteration when a given fold was used as testing data, the remaining four folds were used as training data: they were parsed using Joice's grammar and lexicon in PC-PATR, and their parse trees analysed by our PCFG learning program. From the learnt language model, the lexical rules were discarded, and the remaining syntactic rules were concatenated with the lexical rules previously trained on the entire corpus.

This PCFG was then used to parse each sentence in the training fold using the probabilistic CKY algorithm. If successful, exactly one parse tree would be returned. We then compared this parse tree with the set of possible parse trees returned by PC-PATR on the training fold, and computed the occurrence of the five possible cases, i.e.:

- **Case 1:** The PCFG parse tree correctly appears in the set of PC-PATR parse trees.
- **Case 2:** The PCFG parse tree does not appear in the set of PC-PATR parse trees.
- **Case 3:** Both the probabilistic parser and PC-PATR fail to return valid parses.
- **Case 4:** The probabilistic parser fails to return a parse tree, but PC-PATR does.
- **Case 5:** The probabilistic parser return a valid parse tree, but PC-PATR fails to do so.

5. Results and discussion

The results are shown in Table 1. In the majority of instances (58.96%), the most probable parse tree based on the PCFG model is not present in the set of candidate parse trees returned by the constraint-based parser. The PCFG agrees with the more sophisticated constraint-based parser only 6.56% of the time. This seems to indicate that the statistical information implicitly encoded in the PCFG is somehow failing to capture the subcategorization rules explicitly encoded in the PATR-II feature structure constraints.

However, to further investigate what is going on, we present some sample sentences and parse trees returned by both parsers.

One example sentence of Case 1 is as follows:

[1] available at `http://www.cog.brown.edu/∼mj/Software.htm`

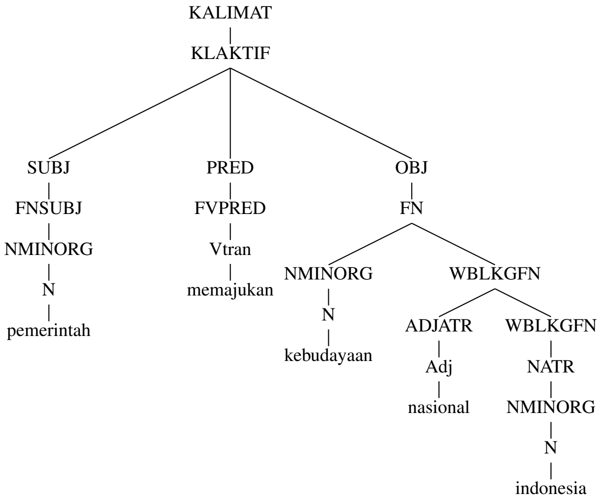

Figure 2. Probabilistic and constraint-based parse for Case 1 example sentence

(2)

pemerintah	*memajukan*	*kebudayaan*	*nasional*	*indonesia*
government	advances	culture	national	indonesia

‘The government helps advance the national culture of Indonesia.’

for which both the PCFG model and the constraint-based parser return the same parse tree, shown in Figure 2. This is a fairly straightforward sentence, which both parsers have little difficulty in analysing. However, note that the global sentential structure is analysed as *SUBJ PRED OBJ*. This approach seems to mix pure syntax with some semantic functions, and also means that the syntactic rules for noun phrases are unnecessarily duplicated for subject and object position. Moreover, it does not capture the regularity of verb subcategorization in the way that verb phrase constituents do.

As for the most frequent occurrence of Case 2, one example instance is the following sentence:

(3)

komnas	*ham*	*berasaskan*	*pancasila*
national commission	human rights	has-foundation	pancasila

‘The national commission for human rights has Pancasila as its foundation.’

Given this sentence, the probabilistic model returns the parse tree in Figure 3(a), whereas the constraint-based symbolic model constructs the parse tree in Figure 3(b). The main difference is that the probabilistic model analyses the sentence as transitive sentence, as evidenced by the *SUB PRED OBJ* structure, yet tags the verb *berasaskan* as an intransitive verb, resulting in an inconsistent parse. This is indeed a limitation of the PCFG formalism – it inherits the CFG independence assumption that non-terminal rewrites can be applied anywhere. As a result, there is no way of conditioning the sentence structure on the type of verb. The constraint-based parser, on the other hand, analyses the sentence as an intransitive sentence, with a complementary modifier *PEL = pelengkap*). However, it seems that even this analysis is still awkward: the role of *pancasila* in this sentence plays more of a predicative role rather than a complementary one.

This example highlights several points:

1. The correct way to assess the parse results would actually be to compare the parse trees selected by the PCFG and the constraint-based symbolic parser against a **gold standard**, i.e. human judgments of syntactic structure. Unfortunately, we do not yet have such annotations in place. This is a direction for future work.

2. Our evaluation method takes a discrete approach in comparing whether the PCFG-selected parse appears in the constraint-based set of alternatives or not. However, there will invariably be instances, as above, where the PCFG-selected parse tree is very similar, but not identical, to a constraint-based parse. This suggests that more sophisticated models of **tree similarity** could be applied, e.g. [12].

Finally, we present an example sentence which both models are unable to parse, i.e. Case 3, as follows:

(4)

setiap	*orang*	*bebas*	*untuk*	*bertempat*	*tinggal*	*di*	*wilayah*	*negara*	*,*	*meninggalkannya*	*,*	*dan*	*berhak*	*untuk*	*kembali*
each	person	free	to	domicile		at	area	country	,	leave	,	and	right	to	return

‘All citizens are free to take residence within the country, leave, and subsequently has the right to return.’

This is indeed a slightly awkward construction, particularly the clause *meninggalkannya*, which has some anaphoric reference to elements in the previous clause. We believe that a correct analysis of such a sentence may require some measure of linguistic sophistication, such as the use of feature structures to co-index implicitly appearing constituents with their text, as in the *gap-threading* technique [10].

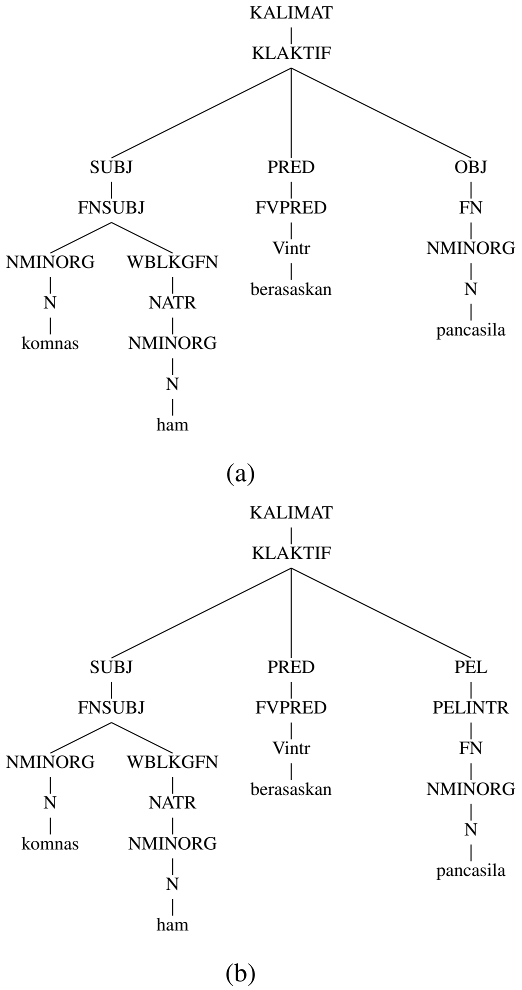

Figure 3. (a) Probabilistic and (b) constraint-based parse for Case 2 example sentence

6. Summary

We have presented an initial experiment in constructing a probabilistic parser for Indonesian, by training a PCFG model on the output of an existing constraint-based symbolic parser. Due to limitations of the PCFG formalism, the model is unable to capture verb subcategorization information. However, it also illuminates certain linguistic infelicities in the constraint-based grammar.

We believe there is much benefit to be obtained from the fine-grained modelling of probabilistic grammars, such as the ability to disambiguate, and intend to pursue more thorough evaluation, i.e. against a gold standard, and using verious tree similarity algorithms. Further down the road, more sophisticated **lexicalized** formalisms should be investigated, among others lexicalized tree adjoining grammar (LTAG) [5].

References

[1] H. Alwi, S. Dardjowidjojo, H. Lapoliwa, and A. Moeliono. *Tata Bahasa Baku Bahasa Indonesia*. Balai Pustaka, Jakarta, Indonesia, third edition, 1998.

[2] I. Hendrawan. Pengurai sintaks kalimat untuuk bahasa indonesia dengan metode linguistic string analysis. Undergraduate Thesis, Faculty of Computer Science, University of Indonesia, 1999. Call number: SK-0388.

[3] M. Johnson. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, December 1998.

[4] Joice. Pengembangan lanjut pengurai struktur kalimat bahasa indonesia yang menggunakan constraint-based formalism. Undergraduate Thesis, Faculty of Computer Science, University of Indonesia, 2002. Call number: SK-0487.

[5] A. K. Joshi and Y. Schabes. Tree-adjoining grammars and lexicalized grammars. Technical Report IRCS-91-04, Institute for Research in Cognitive Science, University of Pennsylvania, 1991.

[6] D. S. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, 2000.

[7] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, USA, May 1999.

[8] M. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[9] S. McConnel. *PC-PATR Reference Manual*. Summer Institute for Linguistics, October 1995. http://www.sil.org/pcpatr/manual/pcpatr.html.

[10] F. Pereira and S. Shieber. Prolog and natural-language analysis. CSLI Lecture Notes 10, Center for the Study of Language and Information, Stanford, USA, 1987.

[11] S. Salvitri. Pengembangan lanjut pengurai sintaks bahasa indonesia dengan metode linguistic string analysis. Undergraduate Thesis, Faculty of Computer Science, University of Indonesia, 1999. Call number: SK-0417.

[12] G. Sampson. A proposal for improving the measurement of parse accuracy. *International Journal of Corpus Linguistics*, 5(1):53–68, 2000.

[13] S. Sari. Prototipe pemeriksa tata bahasa baku bahasa indonesia: sebuah program yang dikembangkan dengan alat bantu lex dan yacc pada sistem operasi unix. Undergraduate Thesis, Faculty of Computer Science, University of Indonesia, 1995. Call number: SK-0287.

[14] S. Shieber. An introduction to unification-based approaches to grammar. CSLI Lecture Notes 4, Center for the Study of Language and Information, Stanford, USA, 1986.