Building Cendana: a Treebank for Informal Indonesian

David Moeljadi
Department of Asian Studies, Faculty of Arts
Palacký University Olomouc
Czechia
davidmoeljadi@gmail.com

Aditya Kurniawan, Debaditya Goswami
NLP, Vision, Speech (NVS)
Traveloka Services Pte. Ltd.
Singapore
{akurniawan,
debaditya.goswami}@traveloka.com

Abstract

This paper introduces Cendana, a treebank for informal Indonesian. The corpus is from a subset of online chat data between customer service staff and customers at Traveloka (traveloka.com), an online travel agency (OTA) from Indonesia that provides airline ticketing and hotel booking services. Lines of conversation text are parsed using the Indonesian Resource Grammar (INDRA) (Moeljadi et al., 2015), a computational grammar for Indonesian in the Head-Driven Phrase Structure Grammar (HPSG) framework (Pollard and Sag, 1994; Sag et al., 2003) and Minimal Recursion Semantics (MRS) (Copestake et al., 2005). The annotation was done using Full Forest TreeBanker (FFTB) (Packard, 2015). Our purpose is to create a treebank, as well as to develop INDRA for informal Indonesian. Testing on 2,000 lexically dense sentences, the coverage is 64.1% and 715 items or 35.8% was treebanked, with correct syntactic parses and semantics. INDRA has been developed by adding 6,741 new lexical items and 22 new rules, especially the ones for informal Indonesian. The treebank data was employed to build a Feature Forest-based Maximum Entropy Model Trainer. Testing against the annotated data, the precision was around 90%. Moreover, we leveraged the treebank data to develop a POS tagger and present benchmark results evaluating the same.

1 Introduction

This work is an attempt to build a new open resource for colloquial/informal Indonesian annotated corpus or a treebank, i.e. a linguistically annotated corpus/text data that includes some grammatical analyses, such as parts-of-speech, phrases, relations between entities, and meaning representations. The existing treebanks for Indonesian are mainly for formal Indonesian, e.g. manually tagged Indonesian corpus (Dinakaramani et al., 2014) and JATI (Moeljadi, 2017). Thus, building a treebank for informal Indonesian can be considered as a pioneer. This treebank is named Cendana, the Indonesian word for "sandalwood", built using tools developed in the Deep Linguistic Processing with HPSG (DELPH-IN) community.[1] This paper describes the construction of this new language resource and gives new analyses and implementations on phenomena in informal Indonesian morphology and syntax.

2 Sociolinguistic situation in Indonesia

Indonesian (ISO 639-3: ind), called bahasa Indonesia (lit. "the language of Indonesia") by its speakers, is spoken mainly in the Republic of Indonesia by around 43 million people as their first language and by more than 156 million people as their second language (2010 census data). The lexical similarity is over 80% with Standard Malay (Lewis, 2009). It is written in Latin script. Morphologically, Indonesian is a mildly agglutinative language. It has a rich affixation system, including a variety of prefixes, suffixes, circumfixes, and reduplications. The basic word order is SVO (Sneddon et al., 2010).

The diglossic nature of the Indonesian language exists from the very beginning of the historical

[1] http://www.delph-in.net

33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), pages 156-164, Hakodate, Japan, September 13-15, 2019

record when it is called Old Malay around the 7th century to the present day (Paauw, 2009). While much attention has been paid to the development and cultivation of the standard/formal “High” (H) variety of Indonesian, little attention has been particularly paid to describing and standardizing the informal “Low” (L) variety. Sneddon (2006) calls this variety “Colloquial Jakartan Indonesian” and states that it is the prestige variety of colloquial Indonesian in Jakarta, the capital city of Indonesia, and is becoming the standard informal style. In addition to this L variety, more than 500 regional languages spoken in Indonesia, such as Javanese, Balinese, and various local Malay languages, add to the complexity of the sociolinguistic situation in Indonesia.

The H variety is used in the context of education, religion, mass media, and government activities. The L variety is used for everyday communication. The regional vernaculars or *bahasa daerah* are used for communication at home with family and friends in the community. In this paper, the term ‘informal Indonesian’ or L variety refers to Colloquial Jakartan Indonesian mentioned above.

3 Traveloka Conversational Corpus

We use more than 10 millions of lines of conversation or chat data between Traveloka users and customer service agents. We have more varieties in terms of language registers in the chat data, compared with other commonly used text for corpus such as newspaper and Wikipedia articles. The customer service agents usually write in H variety, while the users or customers usually write in L variety. Many informal features which can be found in online written text such as in tweets (Le et al., 2016), also appear in the chat data. They are informal words, abbreviations, typos, discourse particles, interjections, foreign words, emojis, emoticons, and unusual word orders, as shown in Table 1.

The raw data is mainly in H and L varieties of Indonesian or Indonesian with some English words related to flights, hotels, bookings, and payments such as “booking”, “check-in”, “form”, “payment” and sometimes they appear together with Indonesian affixes, e.g. *formnya* “the form”. Very few chat lines are written entirely in foreign languages, such as English, Malay, Javanese,[2] Vietnamese, Tagalog, and German. Traveloka is expanding to countries in Southeast Asia and Australia and thus, we got chat data in various languages. In addition to the informal features, the raw data has been processed to mask sensitive information such as email addresses, phone numbers, and booking codes/numbers. The data preprocessing is described in Section 5.1. It includes text normalization, sentence segmentation (chunking the chat data into sentences), and word tokenization (chunking a sentence into words).

4 Related work

There are few open-source treebanks for Indonesian, annotated with both syntactic and semantic information. Most previous work on Indonesian treebanks focuses on the H variety and on syntactic annotation, rather than semantic annotation, e.g. the Indonesian Dependency Treebank developed by Charles University in Prague (Green et al., 2012), with manually annotated dependency structures for Indonesian; the Indonesian treebank developed by the University of Indonesia (UI) (Dinakaramani et al., 2014) which uses a part-of-speech (POS) tagged corpus as a starting point and adopts Penn Treebank bracketing guidelines; and the Indonesian treebank in the Asian Language Treebank (ALT) which was built by the Agency for the Assessment and Application of Technology (BPPT) (Riza et al., 2016), comprises about 20,000 sentences originally sampled from the English Wikinews in 2014, and uses tools such as POS tagger, syntax tree generator, shallow parser, and word alignment. The Indonesian Treebank in the ParGram Parallel Treebank (ParGramBank) (Sulger et al., 2013) is based on Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Dalrymple, 2001) and publicly available via the INESS treebanking environment but contains only 79 sentences and 433 words.

Similar to the Indonesian Treebank in ParGramBank, another treebank called JATI (Moeljadi, 2017) was built based on a computational grammar for Indonesian called the Indonesian Resource Grammar (INDRA) (Moeljadi et al., 2015).[3] The raw cor-

[2] Regional languages such as Javanese are treated as foreign languages in this paper.

[3] `http://moin.delph-in.net/IndraTop`

Feature	Example
Informal word	*gak* (*tidak* “NEG”), *mulu* (*melulu* “only, just”), *uda* (*sudah* “PERF”), . . .
Abbreviation	*sy* (*saya* “1SG”), *cm* (*cuma* “only”), *yg* (*yang* “REL”), *jg* (*juga* “too”), . . .
Typo	*tikey* (*tiket* “ticket”), *abntu* (*bantu* “help”), *sata* (*saya* “1SG”), *fi* (*di* “at”), . . .
Discourse particle	*koq*, *lho*, *nich*, *yach*, *sich*, *donk*, *deh*, *kek*, *mah*, *nah*, *tuh*, *yuk* . . .
Interjection	*hahaha* (*haha* “ha-ha”), *wkwkwk* (*haha* “ha-ha”), *hehehe*, *hihi*, *wowww*, . . .
Foreign word	within (English), *semakan* (Malay), *ngono* (Javanese), *trong* (Vietnamese), *maawain* (Tagalog), . . .
Emoji/emoticon	:), :(, :-|, ^_^

Table 1: Informal features in Traveloka chat data

pus data are dictionary definition sentences related to food and beverages, extracted from the official Indonesian dictionary (KBBI) fifth edition. INDRA is open-source and it is developed within the framework of HPSG and MRS, using tools and resources developed by the DELPH-IN research consortium. The creation of Cendana is similar to the one of JATI but deals with both H and L varieties. Cendana uses INDRA to parse the data. During the treebank development, INDRA was developed with informal lexicon, morphology, and syntax rules (see Section 5.4). Similar to JATI, Cendana uses an approach called “parse and select by hand”, in which lines of corpus data are parsed and the annotator selects the best parse from the full analyses derived by the grammar.

5 Treebank development

Treebanking is a part of grammar development process (Bender et al., 2011), as shown in Figure 1. The motivation is to develop a broad-coverage grammar together with the treebank, which allows the grammar developer to immediately identify problems in the grammar and the treebanker to improve the quality of the treebank (Oepen et al., 2004). The process starts from preparing the corpus data or test-suite.

Section 5.1 describes the data preprocessing part before creating the test-suite, which is mentioned in Section 5.2. Afterwards, the lexical acquisition, linguistic type classification, linguistic phenomena analysis, and implementation, are described in Section 5.3 and Section 5.4. Lastly, the annotation/treebanking part is written in Section 5.5.

5.1 Data preprocessing

Data preprocessing includes text normalization, sentence segmentation, and word tokenization, as illustrated in Figure 2.

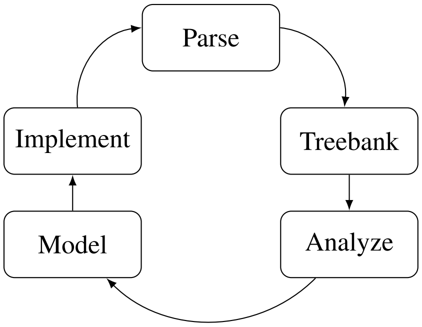


Figure 1: Grammar engineering spiral

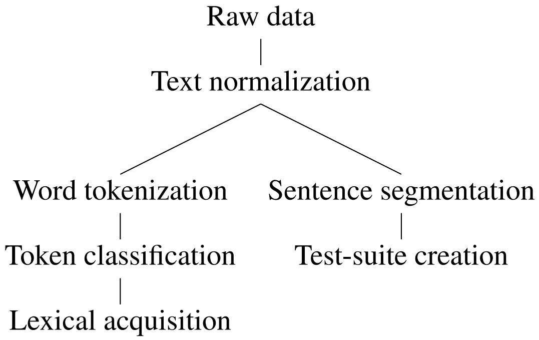


Figure 2: Data preprocessing and lexical acquisition

Text normalization: In order to ensure privacy of any user data within the linguistic corpus outlined in Section 3, we encoded email addresses into a token EMAIL, phone numbers into a token PHONE_NUMBER, website addresses into a token SITE, URIs into a token URI, image into a token image, @ sign into a token AT, and booking numbers into a token NUMBER. We normalized repetitive punctuations, removed spaces in abbreviations and within a single token, added spaces between numerals and nouns, removed excessive characters, converted Unicode into ASCII characters, and removed non-printable characters like emoticons.

Sentence segmentation and word tokenization: We used Python 3 and Natural Language Toolkit (NLTK) (Bird et al., 2009) for sentence segmentation and word tokenization. After that, we counted the number of sentences and tokens. We had 13,372,929 sentences and 111,175,597 tokens. There are duplicates of sentences and tokens, thus we counted the number of unique sentences and tokens, too. There are 8,527,072 unique sentences (63.8% of total number of sentences) and 693,718 unique tokens (0.6% of total number of tokens).

5.2 Test-suite creation

For the purpose of building Cendana, only a representative subset of the chat data having the most lexically dense tokens, is extracted. We extracted a sample of data consisting of two thousand sentences having at least ten tokens in a sentence. The lexical density is measured by dividing the number of lexical word tokens (tokens written in alphabet other than stop words and foreign words) by the number of all tokens. We used NLTK for stopwords and added more stopwords from spaCy.[4] Since the available sources for stopwords are for formal Indonesian, we added more stopwords for informal Indonesian.

We made a test-suite, i.e. a sample of text, selected and formatted for treebanking. The format is explained in the DELPH-IN page.[5] Each line in the test-suite consists of an ID number, a sentence, the number of tokens in that sentence, an optional comment, and information on author and date.

5.3 Linguistic type classification and lexical acquisition

After word tokenization with NLTK, we extracted 63,294 tokens (0.09% of the total number of unique tokens) which have at least two characters and have frequency more than ten. Before lexical acquisition from the chat data, INDRA had 16,751 lexical items. Out of 63,294 unique tokens extracted, 3,059 tokens were already in INDRA's lexicon. Thus, there is a potential to add more lexical items, especially the informal ones, into INDRA. We did lexical acquisition firstly for tokens having a circumfix *pe-...-an*, *ke-...-an*, enclitic *-nya*, *-ku*, and those with reduplication (marked with a hyphen). These tokens are usually nouns. Afterwards, we added tokens having a prefix *me-*, *di-*, *nge-*, and a suffix *-kan* and *-in*. These tokens are usually verbs. This lexical acquisition process was not done at once, instead it was done throughout the treebanking project, before and during treebanking. During lexical acquisition, we grouped the tokens based on lexical types in INDRA, e.g. inanimate noun, temporal noun, intransitive verb, ditransitive verb, and transitive verb with an optional or obligatory complement.

We keep in mind that the same semantic predicate is applied to the lexical items having the same concept, regardless their varieties (H or L). For example, the negation word with non-nominal predicates is *tidak* "NEG". Sneddon (2006) notes this as a word which mostly appears in the H variety. He lists six counterparts of it in the L variety: *enggak*, *nggak*, *ngga*, *gak*, *kagak*, and *ndak*. We found 32 more variants in the data, including abbreviations and typos: *nda*, *nd*, *dk*, *nfk*, *ndk*, *tda*, *tijdvak*, *tidaj*, *tidar*, *tidk*, *tida*, *tdak*, *tdk*, *tidsk*, *ngaak*, *ngaa*, *nggaj*, *nggah*, *nggal*, *nggk*, *ngg*, *ngak*, *ngal*, *nga*, *ngk*, *ngx*, *ngakk*, *kgk*, *gag*, *ga*, *gk*, and *g*. All these 39 lexical items, although they are orthographically different, have the same concept semantically and thus, they are given the same MRS semantic predicate. After lexical acquisition, INDRA has 7,181 more lexical items, thus the total number of lexical items in INDRA became 23,932.

5.4 Linguistic phenomena analysis and implementation in INDRA

Linguistic phenomena in the test-suite are identified and analyzed based on reference grammars and other linguistic literature. The analyses are modeled in HPSG and implemented in INDRA.

Text normalization in INDRA: Beside text normalization mentioned in Section 5.1, we did more detailed text normalization using INDRA, dealing with typos, morphology, and token boundaries (see Table 3). In addition, we added more regular expression patterns to detect dates and currencies.

Active voice prefixes: Formal Indonesian has transitive verbs in active voice which take prefix

[4] `https://github.com/explosion/spaCy/blob/master/spacy/lang/id/stop_words.py`

[5] `http://moin.delph-in.net/ItsdbReference`

Action	Example	
	Before	After
Normalize repetitive punctuation	,,,+++!!!	,+!
Remove spaces	␣e␣-tiket␣␣a␣.␣n␣␣Mr␣.␣John	e-tiket a.n Mr.John
	Rp␣800000␣,-␣01␣/␣09	Rp800000,- 01/09
Add spaces	30Juni 2org 1anak 1kmr 2bed 1hr	30 Juni 2 org 1 anak 1 kmr 2 bed 1 hr
Remove excessive characters	ruangannnya hahahaha	ruangannya haha
Encode emails etc	abc@abc.com, http://abc, www.ab.co,	EMAIL, SITE, URI,
into respective tokens	+62-1234-5678, image.jpg, @	PHONE_NUMBER, IMAGE, AT

Table 2: Text normalization

Action	Example	
	Before	After
Fix words	*bantuaanya*, *danannya*, *kodebya*	*bantuannya*, *dananya*, *kodenya*
-nya as a separate token	*hotel nya*, *uang ny*, *tiket nyq*, *namax*	*hotel -nya*, *uang -nya*, *tiket -nya*, *nama -nya*
Fix token boundary	*kal omau di gantii tiketsaya 7an CGKPDG*	*kalo mau diganti tiket saya tujuan CGK PDG*

Table 3: Text normalization in INDRA

meN-, where N symbolizes a nasal which assimilates to the first sound of the verb stem. Moeljadi et al. (2015) show how this is dealt with in INDRA, in terms of morphological rule and inflectional rule. In informal Indonesian, the situation is more complex, Sneddon (2006) notes there are four possibilities:

- without any prefix
- with prefix *meN-*, as in formal Indonesian
- prefix *N-*, just drop the *me*, except for stems started with *c* and *per*
- prefix *nge-*, which occurs before all initial consonants except *p*, *t*, *s*, *c*, *k* if the stems have more than one syllable. The initial *h* is often lost. Prefix *nge-* occurs before *p*, *t*, *s*, *c*, *k* when the stems are one-syllable or the stems are borrowings, either assimilated or unassimilated borrowings.

In addition to these four possibilities, we found another one in our chat text data:

- prefix *m(N)-*

Table 4 shows these five possibilities with examples. We analyzed the patterns and implemented the rules. INDRA's lexicon lists down only the stems or the forms without prefixes.

Using the morphological and inflectional rules, INDRA can parse and generate all surface forms both with and without prefixes. All surface forms having different surface forms but derived from the same stem, have the same MRS semantic predicate. For example, *proses*, *memproses*, *mproses*, *mroses*, *ngeproses* have the same semantic predicate `_proses_v_rel`.

Because of this, given a formal sentence as input, INDRA can generate all informal sentences. For example, given an input: *Traveloka memproses pesanan saya* "Traveloka processes my booking", INDRA can generate the outputs: *Traveloka proses pesanan saya*, *Traveloka ngeproses pesanan saya*, *Traveloka mproses pesanan saya*, *Traveloka mroses pesanan saya*, ...

Compound rules for proper names Two rules for proper name (PROPN) compound were made. The first was given an underspecified semantics predicate because this type of compound can have a different meaning in different context, similar to a noun-noun compounds which are often highly ambiguous and thus, it seems necessary to have a large degree of 'world knowledge' to understand them (Ó Séaghdha, 2007).

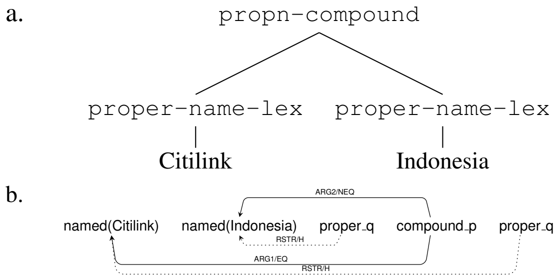
It may have a semantic relation IN, e.g. *CGK JKT* and *PLM Palembang* as in *rute CGK JKT ke PLM Palembang* "the route (from) CGK (airport) (IN) JKT (Jakarta) to PLM (airport) (IN) Palembang"; it may also have a semantic relation SPECIFICALLY, e.g. *Surabaya Juanda* as in *menuju Surabaya Juanda* "towards Surabaya SPECIFICALLY Juanda (airport)"; another possibility is a semantic relation BELONG, e.g. *Citilink Indonesia* as in *maskapai penerbangan Citilink Indonesia* "Citilink airline (which BELONGS TO) Indonesia"; and the last one is a relation which connects name parts e.g. *F Budi Warsito*. Similar to noun-noun compound analysis and implementation in INDRA (Moeljadi, 2018), the underspecified semantics is represented by `compound_p_rel`, which

stem	**without prefix**	***meN-***	***m(N)-***	***N-***	***nge-***
p-initial (also m-initial)	*panggil* “call”	*memanggil*	*mmanggil*	*manggil*	(none)
b-initial	*bantu* “help”	*membantu*	*mbantu*	*mbantu*	*ngebantu*
t-initial (also n-initial)	*tunggu* “wait”	*menunggu*	*mnunggu*	*nunggu*	(none)
d-initial (also j-initial)	*dapat* “get”	*mendapat*	*mdapat*	*ndapat*	*ngedapat*
c-initial	*cuci* “wash”	*mencuci*	*mcuci*	*nyuci*	(none)
s-initial (also ny-initial)	*sewa* “rent”	*menyewa*	*mnyewa*	*nyewa*	(none)
k-initial (also ng-initial)	*kirim* “send”	*mengirim*	*mngirim*	*ngirim*	(none)
g-initial	*ganti* “replace”	*mengganti*	*mganti*	*ngganti*	*ngeganti*
h-initial	*hitung* “count”	*menghitung*	*mhitung*	*ngitung*	*ngehitung*
l-initial (also r-initial)	*lempar* “throw”	*melempar*	*mlempar*	*nglempar*	*ngelempar*
vowel initial	*ambil* “take”	*mengambil*	*mngambil*	*ngambil*	(none)
borrowing	*proses* “process”	*memproses*	*mproses*	*mroses*	*ngeproses*
one syllable	*cek* “check”	*mengecek*	*mngecek*	(none)	*ngecek*

Table 4: Morphology process of active voice prefixes

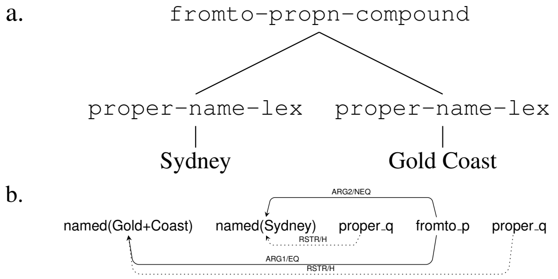
takes two proper names as its arguments, as shown in (1).

(1)



The second one has a special semantics predicate for directions (FROM one place TO another place) and appears a lot in the data, e.g. *pesawat JOG CGK* “plane FROM JOG (airport) TO CGK (airport)”, also *jur sydney gold coast* “direction FROM Sydney TO Gold Coast”, as illustrated in (2).

(2)



In addition to the morphology of active voice prefixes and compound rules for proper names mentioned above, new syntactic rules, e.g. imperatives and a `head-subject` rule for informal Indonesian, as well as discourse particles, were added.

5.5 Annotation

The treebanking process was done semi-automatically using an approach called “parse and select by hand” or “discriminant-based treebanking”. It is a grammar-based corpus annotation, using INDRA to parse and select or reject discriminants or possible readings until one (best) parse remains. The discriminant-based treebanking produces all syntactic and semantic parses which are grammatical and consistent, it gives feedback to INDRA, and if there’s some changes or updates in the grammar, it is easy to update the treebank. However, its coverage is restricted by the computational grammar (INDRA). A treebanking tool called Full Forest TreeBanker (FFTB) (Packard, 2015) was used to select the best tree with correct syntactic and semantic parse from the ‘forest’ of possible trees proposed by INDRA for each sentence, and store it into a database that can be used for statistical ranking of candidate parses.

The test-suite is parsed using INDRA and then the first author as the only annotator selects the correct analysis (or rejects all analyses) using FFTB. The system selects features that distinguish between different parsers and the annotator selects or rejects the features until only one parse is left. The choices made by the annotators are saved and thus, it is possible to update the treebank when the grammar changes (Oepen et al., 2004). If a sentence is ungrammatical or if INDRA cannot parse the sentence, no discriminants will be found. However, if a sentence is grammatical and no correct tree is found, all the possible trees should be rejected and the grammar has to be modified or debugged. Sentences for which no analysis had been implemented in the grammar or which fail to parse are left unannotated.

Using FFTB, we can note some interesting findings or linguistic analyses item by item. During the treebanking process, new words, especially informal words, and new rules were added into INDRA, so that INDRA can parse informal Indonesian sentences. Some phenomena in colloquial Indonesian were analyzed (see Section 5.4).

6 Result and evaluation

Cendana can be evaluated by measuring the number of coverage, i.e. how many sentences or how many percent of total sentences INDRA can parse and how many of them are good (having correct parse trees and semantics).

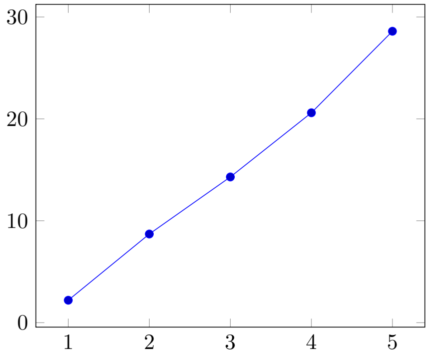


Figure 3: Evolution of coverage for the first 100,000 items (x axis = stage, y axis = coverage)

Figure 3 shows the increase in coverage of the first 100,000 items in the chat data from stage one to stage five. Out of 100,000 first items, 73.3% or 73,280 items are unique. At the **first stage**, we did coverage test before data preprocessing. The result is only 1,614 items or 2.2% could be parsed by INDRA. We got an increase to 8.7% (**second stage**) after lexical acquisition for tokens with affixes *pe-...-an*, *ke-...-an*, *-nya*, *meN-*, *di-* with frequency above 10. We got 14.3% coverage at the **third stage** after lexical acquisition for tokens with *-kan*, *N-*, *-in*, adding morphological rules for active voice prefixes and text normalization in INDRA. At the **fourth stage**, after adding compound rules for proper names and lexical acquisition for other tokens having frequency more than 1000, the coverage increased to 20.6%. At the **fifth stage**, we added more tokens which appear more than 100, including typos, as well as regular expressions for dates and time, discourse particles, and got a coverage of 28.6%. At this point, we began to make a test-suite for 2,000 representative items (see Section 5.2).

Testing INDRA on this full set of 2,000 items at the **initial stage** gave a coverage of 12.9%, as illustrated in **Figure 4**. We added rules for imperatives and added more words, and got 16.8% coverage at the **second stage**. The first big increase in coverage to 36% (**third stage**) was from lexical acquisition. At this point, we started treebanking 20 items or sentences. We kept doing lexical acquisition when treebanking and got 42% coverage with 37 items treebanked at the **fourth stage**. At the present stage (**seventh stage**), testing INDRA on the set of 2,000 items from Cendana test-suite gave a coverage of 64.1% with 715 items treebanked. INDRA has been developed too. The number of lexical items has increased from 16,751 before lexical acquisition in October 2018 to 23,932 after 715 items were treebanked in March 2019. Similarly, the number of types has raised from 2,057 to 2,130; the number of lexical rules from 12 to 24; and the number of grammar rules from 63 to 85.

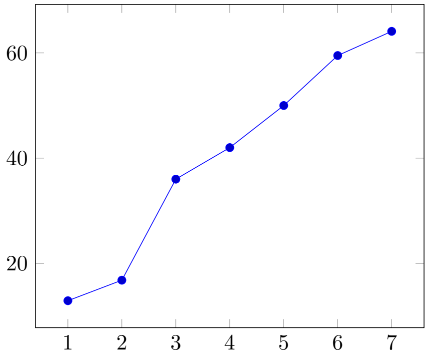


Figure 4: Evolution of coverage for 2,000 items (x axis = stage, y axis = coverage)

The treebanking result is stored in a directory consisting of several text files. The result file contains a derivation tree/phrase structure tree, node labels or POS tags from the phrase structure tree, and a MRS semantics representation for each annotated item. They can be easily edited to accommodate the changes made in INDRA. We made some of the treebank data (552 sentences/items) publicly available, licensed under the GNU General Public License, version 2 for researchers to develop Indonesian NLP.[6] We documented the treebanking process.

We run Feature Forest-based Maximum Entropy Model Trainer, using a tool developed in DELPH-IN[7] based on Miyao and Tsujii (2002). We used the model to treebank 1,000 sentences (number 9000 to 9999) automatically. The result was promising: 428 sentences could be treebanked automatically. We checked the model against the 1,000 data which contain manually annotated items. The result was 612 sentences could be treebanked automatically and the precision was around 90%.

The initial effort to leverage the treebank is by testing it for POS Tagging task. Due to the small amount of data in recent treebank, we leverage Wikipedia and manually tagged Indonesian corpus from UI (Dinakaramani et al., 2014) as our training set and use the treebank as our golden test data. The Wikipedia that we use comes from universal dependency (UD) project (Nivre et al., 2016).

[6] https://github.com/davidmoeljadi/INDRA/tree/master/tsdb/gold/Cendana

[7] http://moin.delph-in.net/FeatureForestTrainer

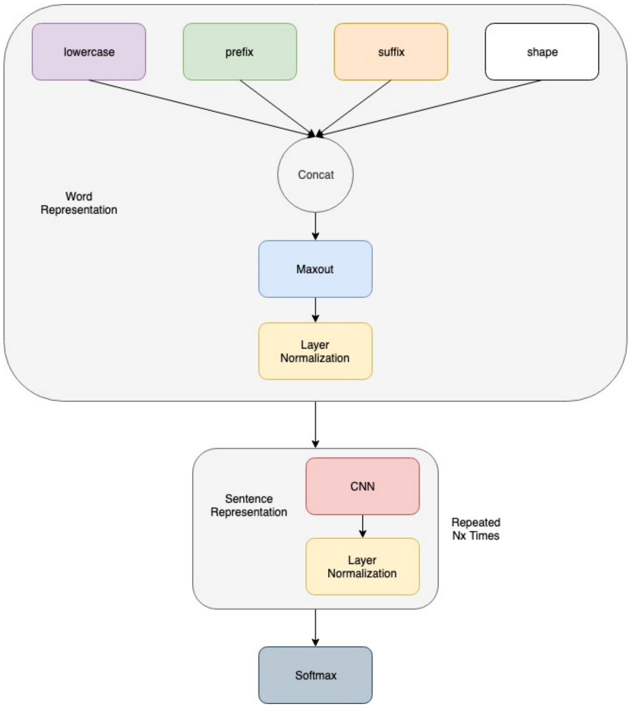


Figure 5: Machine learning model

Train data	Test data	f1-score	OOV in test
UD	UD	92.939	34.33%
UI	UI	97.630	21.68%
UD	Cendana	53.313	68.68%
UI	Cendana	52.168	71.66%

Table 5: POS Tagger experiment

We split the experiment in two folds: we set up the baseline using only UD and UI for all train, validation and test data to see the performance of the model in the same domain, and later we used the training data from UD and UI in order to make prediction on the treebank.

The machine learning models we use to do the training and inference are off-the-shelf model from spaCy. In brief, the algorithm used by spaCy is neural network based with the architecture depicted in Figure 5. The architecture consists of combining multiple features from the word such as lower case, prefix, suffix and shape embedding. Shape embedding is a transformation process by replacing numbers with token `d` and capital words with `w`. The embedding was later concatenated and used as an input to maxout layer. The result was then normalized with layer normalization. After normalization, the output was forwarded to CNN before getting the probability of the tags in softmax layer.

Type of errors	Example
Names	ulfah, mega, subagyo, hadi, heryanto, setiawan, rahayu
Typos	passanger, pkanbaru, rescedule, pembayarsn, soekarna, trransfer, tikcet
Unprocessed numerics	11-12, 20.20, 6.20, 11.10, 29-11-20, 2017, 12.25
Cases (uppercase/ lowercase)	Pemesan, Airliner, Booking, TRINUSA, CGK, DENGAN, Simpati
Abbreviations	tlpn, jog, cgk, jogja, kenapa, kmrn, sya
Tokens	EMAIL, DATE, URL, NUMBER, PHONE, SITE

Table 6: OOV Examples

The number of out-of-vocabulary (OOV) affects the performance of the model quite significantly, as shown in Table 5. We classify the OOV into six different types (see Table 6). We assume that the performance of the model could be improved by adding more data from Cendana.

7 Summary

This paper has described the construction of Cendana treebank, created from a subset of Traveloka chat data, parsed using INDRA, and annotated using FFTB. The construction of Cendana improved the development of INDRA with lexical items and rules for informal Indonesian. At the present stage, the coverage is 64.1% and 35.8% was treebanked, with correct syntactic parses and semantics (715 out of 2,000 items). The treebank was employed to build a Feature Forest-based Maximum Entropy Model Trainer and to develop a POS tagger. The results were promising. Adding more treebank data could improve the performance of the model. Cendana is available on GitHub, under the GNU General Public License.

Acknowledgments

We gratefully acknowledge the support of Traveloka (PT Trinusa Travelindo and Traveloka Services Pte. Ltd.) and the NLP, Vision and Speech department where this research was conducted, and for giving us the opportunity to work with the chat data. The first author also gratefully acknowledges the support of the European Regional Development Fund-Project “Sinophone Borderlands – Interaction at the Edges” CZ.02.1.01/0.0/0.0/16_019/0000791.

References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2011. Grammar Engineering and Linguistic Hypothesis Testing: Computational Support for Complexity in Syntactic Analysis. In *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 5–29. CSLI Publications, Stanford.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(4):281–332.

Mary Dalrymple. 2001. Lexical-Functional Grammar. *Syntax and Semantics*, 34. Academic Press.

Arawinda Dinakaramani, Fam Rashel, Andry Luthfi, and Ruli Manurung. 2014. Designing an Indonesian part of speech Tagset and Manually Tagged Indonesian Corpus. In *2014 International Conference on Asian Language Processing (IALP)*, pages 66–69. IEEE.

Nathan Green, Septina Dian Larasati, and Zdeněk Žabokrtský. 2012. Indonesian Dependency Treebank: Annotation and Parsing. In *26th Pacific Asia Conference on Language, Information and Computation*, pages 137–145.

Ronald Kaplan and Joan Bresnan. 1982. Lexical Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, pages 173–281. the MIT Press, Cambridge.

Tuan Anh Le, David Moeljadi, Yasuhide Miura, and Tomoko Ohkuma. 2016. Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In *Proceedings of the 12th Workshop on Asian Language Resources*, pages 123–131.

M. Paul Lewis. 2009. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, 16 edition.

David Moeljadi, Francis Bond, and Sanghoun Song. 2015. Building an HPSG-based Indonesian Resource Grammar (INDRA). In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) Workshop, 53rd Annual Meeting of the ACL and 7th IJCNLP*, pages 9–16.

David Moeljadi. 2017. Building JATI: A Treebank for Indonesian. In *Proceedings of The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4)*, pages 1–9, Jakarta.

David Moeljadi. 2018. *An Indonesian resource grammar (INDRA): and its application to a treebank (JATI)*. Ph.D. thesis, Nanyang Technological University.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *LREC*.

Diarmuid Ó Séaghdha. 2007. Annotating and learning compound noun semantics. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*, pages 73–78. Association for Computational Linguistics.

Stephan Oepen, Dan Flickinger, and Francis Bond. 2004. Towards holistic grammar engineering and testing–grafting treebank maintenance into the grammar revision cycle. In *Beyond Shallow Analyses–Formalisms and Statistical Modelling for Deep Analysis (Workshop at IJCNLP-2004)*, Hainan Island.

Scott H. Paauw. 2009. *The Malay contact varieties of Eastern Indonesia: A typological comparison*. PhD dissertation, State University of New York at Buffalo.

Woodley Packard. 2015. Full forest treebanking. Master's thesis, University of Washington.

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Hammam Riza, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thái, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, et al. 2016. Introduction of the Asian Language Treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (O-COCOSDA)*, pages 1–6. IEEE.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, 2 edition.

James Neil Sneddon, Alexander Adelaar, Dwi Noverini Djenar, and Michael C. Ewing. 2010. *Indonesian Reference Grammar*. Allen & Unwin, New South Wales, 2 edition.

James Neil Sneddon. 2006. *Colloquial Jakartan Indonesian*. Pacific Linguistics, Canberra.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, et al. 2013. Pargrambank: The pargram parallel treebank. In *ACL (1)*, pages 550–560.

Miyao Yusuke and Tsujii Jun'ichi. 2002. Maximum entropy estimation for feature forests. In *Proceedings of the second international conference on Human Language Technology Research*, pages 292–297. Morgan Kaufmann Publishers Inc.