International Journal of Technology (2017) 5: 957-967
ISSN 2086-9614 © IJTech 2017

A DEPENDENCY ANNOTATION SCHEME TO EXTRACT SYNTACTIC FEATURES IN INDONESIAN SENTENCES

Budi Irmawati[1,2*], Hiroyuki Shindo[1], Yuji Matsumoto[1]

[1]*Computational Linguistics laboratory, Nara Institute of Science and Technology*
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
[2]*Informatics Department, University of Mataram, Mataram 83125, Indonesia*

(Received: June 2016 / Revised: October 2017 / Accepted: October 2017)

ABSTRACT

In languages with fixed word orders, syntactic information is useful when solving natural language processing (NLP) problems. In languages like Indonesian, however, which has a relatively free word order, the usefulness of syntactic information has yet to be determined. In this study, a dependency annotation scheme for extracting syntactic features from a sentence is proposed. This annotation scheme adapts the Stanford typed dependency (SD) annotation scheme to cope with such phenomena in the Indonesian language as ellipses, clitics, and non-verb clauses. Later, this adapted annotation scheme is extended in response to the inability to avoid certain ambiguities in assigning heads and relations. The accuracy of these two annotation schemes are then compared, and the usefulness of the extended annotation scheme is assessed using the syntactic features extracted from dependency-annotated sentences in a preposition error correction task. The experimental results indicate that the extended annotation scheme improved the accuracy of a dependency parser, and the error correction task demonstrates that training data using syntactic features obtain better correction than training data that do not use such features, thus lending a positive answer to the research question.

Keywords: Dependency annotation; Dependency relation; Error correction; Indonesian language; Syntactic information

1. INTRODUCTION

In many natural language processing (NLP) tasks, along with part-of-speech (PoS) tagging, parsing[1] plays an important role in preprocessing (Buchholz & Marsi, 2006). Kübler et al. (2009) state that dependency grammar is useful in languages with a free or a relatively free word order. However, based on the 2007 Conference on Computational Natural Language Learning (CoNLL) Shared Task on Dependency Parsing's results, fixed word order languages such as English and Catalan score highest in accuracy, while free word order languages such as Arabic and Basque are less accurate (Nivre et al., 2007).

Indonesian is rich in morphology and has a relatively free word order compared to English (Stack, 2005). It has yet to be confirmed, however, whether syntactic information extracted from dependency relations is useful for NLP tasks in the case of Indonesian. Addressing this question in the present study required building our own dependency parser, as previous

* Corresponding author's email: budi-i@unram.ac.id, Tel: +62-37-0636-6126, Fax: +62-37-0636-523
Permalink/DOI: https://doi.org/10.14716/ijtech.v8i5.878

[1] Parsing is a formal analysis to split a sentence to its constituent. It results a parse tree to show the relation between its constituents.

Indonesian dependency parsers (Kamayani & Purwarianti, 2011; Green et al., 2012) and their dependency annotation schemes cannot be accessed freely.

To build the parser, we developed a dependency annotation scheme for Indonesian and trained an automatic parser. To start, we adapted the Stanford typed dependency (SD) manual (de Marneffe & Manning, 2008) to accommodate relations not covered by the original SD annotation scheme (explained in Subsection 2.1). In doing so, we found a universal dependency annotation scheme (Nivre et al., 2016) that covered Indonesian. However, this scheme did not include the morphological features and stem words relevant to the language, nor did it handle date and time notations (explained in item d of Subsection 2.2.2) or apply language-specific relations like in Danish and Finnish. Moreover, unlike its performance in French, the scheme did not split off clitics in the case of Indonesian. As our chosen scheme could not avoid ambiguities in assigning heads and relations, we extended it.

In this experiment, we calculated the accuracy of our annotation scheme before and after extending it. We also evaluated the effectiveness of our proposed annotation scheme in correcting preposition errors made by second language (L2) learners in real learner data (Irmawati et al., 2016a), explained in item c of Subsection 3.1. We also used this annotation scheme to extract syntactic information from Indonesian sentences and generate more sophisticated artificial error data (Irmawati et al., 2016b; Irmawati et al., 2017).

This dependency annotation scheme is explained in Section 2 by first describing the phenomena in Indonesian, then proposed annotation scheme. Section 3 describes the experiment setting for assessing the extension and the effectiveness of the annotation scheme in a real implementation scenario. Section 4 presents the results of the experiment and provides a discussion.

2. THE PROPOSED DEPENDENCY ANNOTATION SCHEME

2.1. Language Phenomena Not Covered by the SD Annotation Scheme

Before describing the built dependency annotation scheme, the language phenomena not covered by the SD annotation scheme must be understood. These are listed below.

a. **Non-verb clause**. Unlike English, which always has a verb or an auxiliary verb as a clause head, non-verb clauses in Indonesian take a noun or an adjective as their clause head.

b. **Adverb.** The prepositions *dengan* (with) and *secara* (by) may be used as markers to introduce adverbs derived from adjectives. Adjectives in Indonesian are translated into English as adjectives with the suffix -ly (Sneddon et al., 2010).

c. **Adjective.** To be preceded by a referent *yang* (that/who is/are). *Yang* is used to introduce certain adjectives and keep the natural flow of a sentence, but it is not translatable to English.

d. **Clitic.** A clitic is a morpheme that behaves syntactically like a word but is reduced in form (Sneddon et al., 2010). Front-clitics like *ku*- [1SG] and *kau*- [2SG] function as subjects, while end-clitics like -*ku* [1SG] and -*nya* [3SG] function as direct objects, indirect objects, or possessive pronouns. -*nya* [3SG] also functions as a determiner, but -*nya* in adverbs is not a clitic.

e. **Ellipses.** Ellipses represent the removal of a word if its presence is dispensable in understanding a sentence (Sneddon et al., 2010). Ellipses may occur for word types like determiners, copulas, or possessive pronouns. While a subject or pronoun cannot be ignored in English, Indonesian may drop them in a sub-clause if the meaning is left clear.

f. One of ellipses is copula, a linking verb in the grammar of other languages. In Indonesian, a copula occurs as an option between the subject and the predicate in non-verbal clauses, though it is required when the subject or the predicate is long (more than three words).

2.2. Adaptations and Extensions of the SD Annotation Scheme

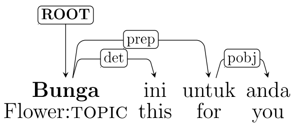
2.2.1. Adaptations of the SD annotation scheme

The SD annotation scheme was adapted to cope with the phenomena explained in Subsection 2.1 that were not covered in the SD annotation scheme. In this adaptation, we made sure that other relations in a sentence did not change—even if a marker existed, such as in Examples (3) and (4), or a word was omitted, as in Example (5)—to keep the natural flow of the sentence.

a. **Adapt-1**: For non-verbal clauses. The head of a noun clause or a quantity clause is a noun; the head of an adjective clause is an adjective. However, in prepositional clauses, the topic is assigned as the head, such as in Example (1).

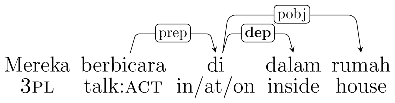
b. **Adapt-2**: For compound prepositions. The SD annotation scheme uses **dep** to label a relation that is not defined by other relations (de Marneffe & Manning, 2008). Thus, we labeled a relation in a compound preposition, such as *di dalam* (inside) as **dep** in Example (2).

(1)



'This flower is for you'

(2)

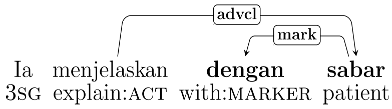

pobj
prep
dep
Mereka berbicara di dalam rumah
3PL talk:ACT in/at/on inside house

They are talking inside the house

c. **Adapt-3**: For adverbs introduced by a marker. Indonesian derives certain adverbs from adjectives (they are preceded by a marker as explained in item b of Subsection 2.1). Because an adverb with a marker is an adverbial clause, we labeled the relation between the adverbial clause and its head as **advcl** and that between the marker and the adverb as **mark**, as seen in Example (3). The relation their head was labeled as **advmod**.

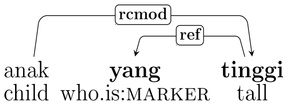
d. **Adapt-4**: For adjectives introduced by a referent (See item c of Subsection c). An adjective introduced by a referent as a relative clause, so we labeled the relation the adjective and its head as **rcmod** and that of the marker to the adjective as **ref**, as seen in Example (4). For regular adjectives, their relation to their head was labeled as **amod**.

(3)


advcl
mark
Ia menjelaskan dengan sabar
3SG explain:ACT with:MARKER patient

'*S/he explains patiently*'

(4)

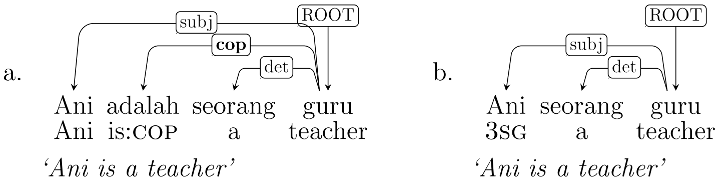


'a tall child'

e. **Adapt-5**: For clitics. We labeled the relation of a front-clitic to its attached word as subj. For an end-clitic, its relation to its attached word was labeled as **dobj** if the clitic followed a transitive verb; as **pobj** if the clitic followed a preposition; and as **iobj** if the clitic followed a ditransitive verb. We also labeled the relation between the clitic and its attached word as **det** if the clitic functioned as a determiner, or **poss** if the clitic functioned as a possessive pronoun.

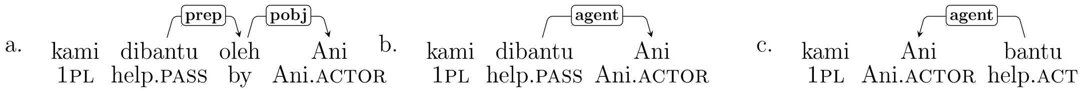
f. **Adapt-6**: For ellipses. To avoid inconsistency in a head assignment, we did not assign any modifiers to words that might be eliminated. Example (5a) shows a copula without a modifier. We did not assign any modifier to such a copula so that other relations within the sentences would not change when the copula was absent, as in Example (5b).

(5)



g. **Adapt-7**: For actors in the passive voice. We labeled the relation of an actor preceded by a preposition in the passive voice as **pobj**, as in Example (6a). However, the SD annotation scheme used **agent** to label the relation between an actor and a passive verb in a collapse dependency. Therefore, we labeled the relation between the actor and the passive verb as **agent** if the preposition was absent, as seen in Example (6b). In the passive type-II (Sneddon et al., 2010), we also labeled the relation between the agent and the verb as **agent**, as seen in Example (6c), because this type did not have a preposition. (All sentences in Example [6] mean "We are helped by Ani".)

(6)



2.2.2. *Extensions of the adapted annotation scheme*

We extended our adapted annotation scheme because the preliminary experiment showed that certain relations assigned by the trained parser were ambiguous.

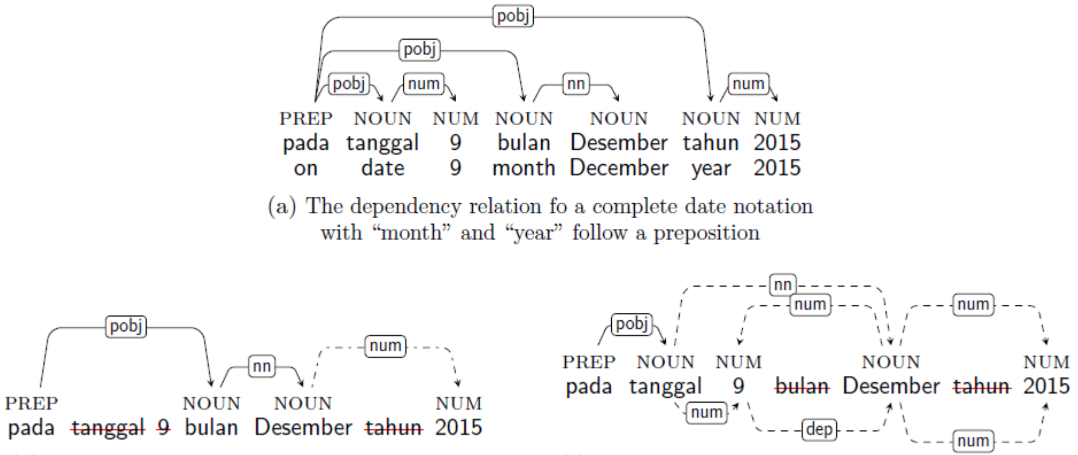
a. **Extend-1**: For compound prepositions. We replaced the **dep** label in Adapt-2 with **pmod**, as in Example (7), to make it easily extractable as a single word.

b. **Extend-2**: For adverbs introduced by a marker. We replaced the **advcl** label in Adapt-3 with **padv** because the parser could mistakenly label the marker as **prep**, thus assigning the head of the adjective as the head of the marker (see the dashed arrow under the sentence in Example (8)). These mistakes were the result of the marker's PoS tag being a preposition.

c. **Chunk-1**: For compound transitional words. The PoS of a compound transitional word can vary—as in *oleh karena itu* (PREP-CONJ-DET) and *oleh sebab itu* (PREP-NOUN-DET) (both meaning "therefore"), and *akan tetapi* (MODAL-CONJ) and *namun demikian* (CONJ-CONJ) (both meaning "however"). To parse them correctly, we referred to a small dictionary of frequently used compound transitional words and replaced their PoS tags with chunks. Then, we assigned a left-to-right relation to those chunks and labeled them as **mwe**.

d. **Chunk-2**: For date and time notations. In addition to transitional words, date and time notations in Indonesian also vary and can have ellipses. Figure 1 shows three different date notations: a complete date notation in Figure 1a and two different omitted date notations in Figures 1b and 1c. Here, we assigned *tanggal*, *bulan*, and *tahun* as objects of a preposition. In Figure 1b, *tanggal* 9 ("date 9") and *tahun* ("year") are omitted from the notation, which changes the dependency relations slightly. The reduced words are struck out with a red line. The dependency relations of these removals thus change and are represented by dotted arrows. However, these ellipses do not affect other relations because the PoS of 2015's head in Figures 1a and 1b remains the same (both PoS's are NOUNS).

Though the meanings of the date notations in Figures 1a and 1c are the same, the dependency relations of the date notation in Figure 1c, because of the ellipses of *bulan* ("month") and *tahun* ("year"), are ambiguous. The question, then, was how to assign a head for *Desember*? We describe two possible dependency relations (above and under the sentence, drawn in dotted arrows): in the dependency relations above the sentence, we assigned *tanggal* as the head of *Desember*, but *Desember* as the head of 9 and 2015 to maintain consistency; in the relations under the sentence, we assigned 9 as the head of *Desember*, even though 9 has a PoS tag different from *bulan*, by considering “9 *Desember* 2015” as a compound word.

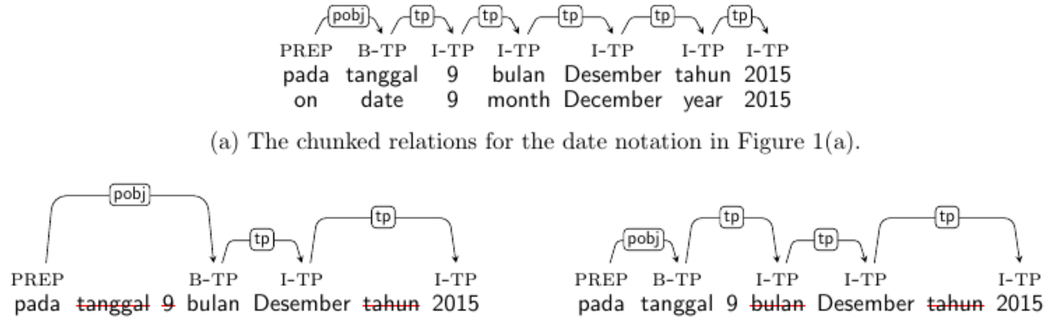

pobj
pobj
pobj
num
nn
num
PREP NOUN NUM NOUN NOUN NOUN NUM
pada tanggal 9 bulan Desember tahun 2015
on date 9 month December year 2015
pobj
nn
num
PREP NOUN NOUN NUM
pada ~~tanggal~~ ~~9~~ bulan Desember ~~tahun~~ 2015
nn
num
num
pobj
PREP NOUN NUM NOUN NUM
pada tanggal 9 ~~bulan~~ Desember ~~tahun~~ 2015
num
dep
num

(a) The dependency relation fo a complete date notation with “month” and “year” follow a preposition

(b) An inherited date notation from *Figure* 1(a), by dropping *tanggal 9* and *tahun*.

(c) Ambiguous relations of an inherited date notation from *Figure* 1(a), by dropping *bulan*.

Figure 1 Head assignments for date notations due to different ellipses


pobj
tp
tp
tp
tp
tp
PREP B-TP I-TP I-TP I-TP I-TP I-TP
pada tanggal 9 bulan Desember tahun 2015
on date 9 month December year 2015
pobj
tp
tp
PREP B-TP I-TP I-TP
pada ~~tanggal~~ ~~9~~ bulan Desember ~~tahun~~ 2015
pobj
tp
tp
tp
PREP B-TP I-TP I-TP I-TP
pada tanggal 9 ~~bulan~~ Desember ~~tahun~~ 2015

(a) The chunked relations for the date notation in Figure 1(a).

(b) The chunked relations for the date notation in *Figure* 1(b).

(c) The chunked relations for the date notation in *Figure* 1(c).

Figure 2 Chunked relations for the date notations in Figure 1

To provide a consistent annotation for cases such as those in Figures 1b and 1c, we chunked the date notations and assigned a left-to-right relation between two consecutive words inside the chunk. This changed the PoS tags of the chunked words to B-TP for the beginning of the chunk and I-TP for the inside of the chunk. Since the words in the date and time notations were more complex than their compound translations, we defined a finite automaton (FA) to decide the beginning and the end of the chunk. Figure 2 represents the modified dependency relations of the date notations in Figure 1 using chunked relations.

3. EXPERIMENTAL METHODS

3.1. Language Tools and Resources

The following language tools and resources were used

a. **Morphind**: an Indonesian morphological analyzer system[2] (Larasati et al., 2011) that

[2] http://larasati.com.morphind/

provides lemmatization information and handles derivations and inflections in a word.

b. **Minimum Spanning Tree (MST) parser** (McDonald et al., 2006): this parser served to evaluate the annotation scheme and to parse native sentences to be used as baseline training data for a real implementation (an error correction model).

c. **Learner data**: error-corrected learner sentence pairs drawn from the lang-8 website[3] (Mizumoto et al., 2011). The data contained 6,488 pairs of learners' sentences and 77,201 tokens. On lang-8, L2 learners write in journals and native speakers correct their sentences. In this study, after we PoS-tagged the learners' sentences and their corrected versions, we automatically aligned learners' sentences and their corrections using a heuristic rule to point out the location of errors and the error types (Irmawati et al., 2016a). We used some of the corrected sentences as training data and others as test data to evaluate the annotation scheme. We then used the learners' sentences as the test data for the error correction model.

d. **Native data**: 1m newspaper sentences were taken from the Indonesian part of the leipzig corpora (Quasthoff et al., 2006) as the baseline training data for the error correction model.

3.2. Pre-processing

To extract the necessary syntactic information, we first tagged all sentences using Morphind. Next, we ran a rule-based script to check whether a word had a clitic; if yes, the clitic was split from its attached word and assigned the clitic with an appropriate PoS tag. The rule-based script also identified adverbs with the suffix *-nya*, which were not to be processed. We then formatted the data in the CoNLL format (Buchholz & Marsi, 2006).

The final step was to annotate manually the 1,032 corrected learners' sentences that contained dependency relations. We then separated 100 sentences to be used as test data. To enable syntactic information extraction, we trained the MST parser on the remaining annotated sentences. The trained parser then automatically assigned dependency relations to the native sentences and learners' sentences. However, to build gold test data annotated with error tags and dependency relations, we had to manually correct the dependency relations of the learners' sentences assigned automatically by the trained parser. We extracted syntactic features like the head and modifiers of a target word, as well as the PoS tag of its head and its modifiers.

3.3. Experiment Setting

The goal of these experiments was to evaluate the effectiveness of our defined annotation schemes. First, we evaluated the accuracy of the adapted and extended schemes. We did not compare our annotation schemes with the universal dependency annotation scheme because we used different data and used different annotation scheme that was not comparable. To compare them, we need some conversions which will be our next research. Second, we evaluated the extent to which our annotation schemes improved an error corrections model.

For this evaluation, we compared the model generated from data using the syntactic features with one that did not use those features and tested it on real learner data.

3.3.1. Evaluation of dependency annotation

To evaluate whether the extensions and chunking were useful to the annotation scheme, we trained and tested an MST parser on the sentences annotated with three annotation schemes as follows:

a. **Adapt**: This scheme followed *Adapt-1* to *Adapt-7* (Subsection 2.2.1). It was the baseline of our evaluation.

[3] http://lang-8.com/

b. **Extend1-2**: This scheme followed the proposed *Extend 1* and *Extend 2*.

c. **Chunk1-2**: This scheme followed *Chunk-1* and *Chunk-2*. In this scheme, we chunked the sentences before assigning heads and labeling the relations.

3.3.2. Evaluation of the error correction task

Before performing an error correction task on a large amount of training data, we conducted a preliminary experiment to evaluate the extended annotation scheme. Two feature sets were compared: (1) a feature set not using syntactic information (*nSynFeat*); and (2) a feature set using syntactic information (*synFeat*). The experiment detected five error types (prepositions, adjectives, adverbs, verbs, and nouns) in 550 learners' manually annotated sentences with error types in five-fold cross-validation.

To obtain a reliable conclusion, 13 preposition errors with error frequencies above five in the learner data were selected for study. We focused on preposition errors because the number of candidate corrections was small compared to other error types; this allowed us to obtain sufficient error samples for each preposition error. From all the learner data, we obtained 382 learners' sentences containing at least one preposition error (Learner) as the test data.

For the training data, we employed the native sentences (Native) and the generated artificial error sentences. We constructed the artificial error sentences by injecting preposition errors randomly to obtain more erroneous sentences easily. We then generated training data that did not use the syntactic features (*nDRndArt*), as well as data that did (*wDRndArt*).

To build the error correction model, we trained a Naïve Bayes classifier. We used the one versus all approach (Rozovskaya & Roth, 2010) to perform a multi class classification. For M target prepositions, we assigned the feature vectors of p prepositions as positive examples and feature vectors of M-1 prepositions as negative examples. Then we used the confident score obtained from each classifier to rank the candidate corrections.

To train our model, we employed context word features in a ±2 window, a bi gram, a tri gram, PoS *n*-grams, head and object of the preposition, and PoS of the head and object of the preposition. For the native training data, we dropped features that contained the target preposition and, for *nDRndArt*, we dropped all dependency features from the training data.

For *Learner*, we trained and tested the model using five-fold cross-validation. We then compared the model trained on *Native*, *nDRndArt*, and *wDRndArt* with the model trained on *Learner*. We checked whether *Learner* performed better as training data as had been claimed in previous works on English prepositions (Cahill et al., 2013; Han et al., 2010).

Then, to assess this task, we followed the evaluation metrics proposed by Dahlmeier and Ng (2011), defined as:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}$$

where precision is the number of correct prepositions proposed by the system divided by the number of corrections given by the system, and recall is the number of correct prepositions proposed by the system divided by the number of preposition errors.

4. RESULTS AND DISCUSSION

4.1. Evaluation of the Dependency Annotation

Table 1 lists the three annotation schemes evaluated in this experiment. It shows that *Extend1-2* and *Chunk1-2* improved both the unlabeled attachment score (UAS) and the labeled attachment score (LAS) compared to the baseline *Adapt*.

Table 1 Dependency parser accuracy of three annotation schemes

Annotations	Accuracy		Complete	
	UAS	LAS	UAS	LAS
Adapt	0.646	0.576	0.217	0.130
Extend1-2	0.788	0.707	0.357	0.214
Chunk1-2	0.812	0.731	0.394	0.214

Complete: % of sentences whose relations are all correct
Unlabeled Attachment Score (UAS): % of tokens with a correct head
Labeled Attachment Score (LAS): % of tokens with a correct dependency relation

The completeness of the sentences improved as well. The improvements occurred because of our proposed *Extend1-2* and *Chunk1-2,* which provided a more consistent annotation scheme than our Adapt annotation scheme. The *Extend1-2* scheme accounted for compound prepositions (in *Extend1*) and they functioned as propositions introducing an adverb (in *Extend2*). The *Chunk1-2* scheme took care of transition words and date and time notations as *mwe*. However, *Chunk1-2* did not assist the LAS's completeness, as some labels in one sentence still were not assigned correctly.

4.2. Evaluation of the Error Correction System

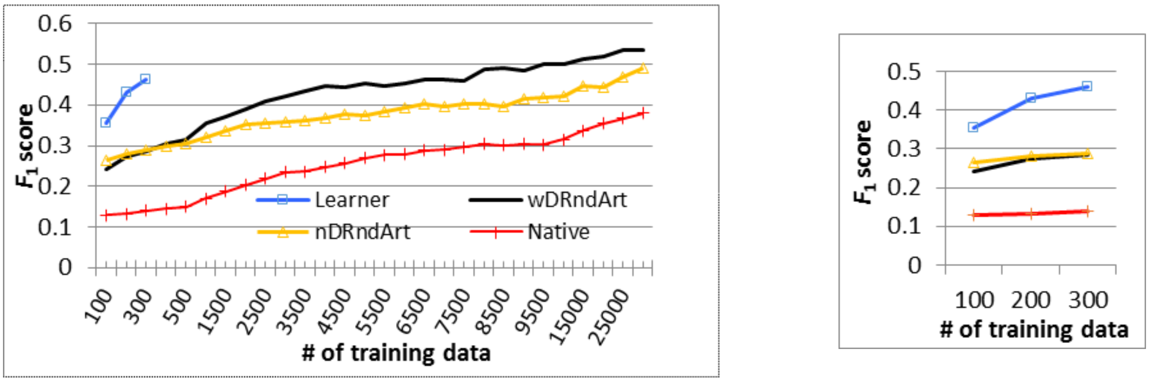
Table 2 presents the preliminary research used to detect five error types. Using a two-tailed test with a confidence interval of 0.95, four error detection models (preposition, adjective, verb, and noun) trained on the data using syntactic information were significantly better than those trained on the data that did not use syntactic information. The results indicate that syntactic information was effective in Indonesian, which has a relatively free word order. This makes our dependency annotation scheme very important for extracting said syntactic information. However, for adverbs, the results were not significantly different because adverb errors were still detectable in the local context (adverbs are usually written not too far from the related verb).

Table 2 Error detection results of five error types

Error Categories	F_1 score	
	nSynFeat	*synFeat*
Preposition	0.135	**0.372**
Adjective	0.292	**0.338**
Adverb	0.351	0.357
Verb	0.471	**0.584**
Noun	0.093	**0.132**

The dependency features significantly improve the F1 score (except for adverbs) by a two-tailed test with a confidence interval of 0.95.

For the preposition error corrections, Figure 3a shows that *Learner* worked best when the training size was less than 300 instances (See Figure 3b for more detail). However, *Learner*'s F_1 score was below 0.5 because of a lack of training data. Figure 3a shows that *Native* received the lowest score. It was unable to outperform *Learner* even when we increased its size to 150K instances. *Native* performed poorly because it did not include information about error types. If we compare *Native* with the two sets of artificial training data, both sets of data worked better because the model could learn what was mistaken in the artificial error sentences and correct those errors in real learner data. Further, Figure 3 shows that the artificial error data using syntactic features (*wDRndArt*) outperformed the other training data. Because extracting syntactic information requires a dependency annotation scheme, the *wDRndArt*'s results indicate that our dependency annotation scheme does indeed improve the preposition error correction task.



(a) Comparison of large training data (b) Comparison of small training data

Figure 3 Comparison of preposition error correction results trained on different sizes of training data

Learner uses learner data as the training data. *Native* uses large amounts of native data as the training data, but cannot outperform *Learner*. *nDRndArt* and *wDRndArt* use artificial training data, with *nDRndArt* using dependency features and *wDRndArt* not. (a) *nDRndArt* outperforms Learner when the data are larger than 15K. The artificial training data using the dependency features of *wDRndArt* performs the best. (b) *Learner* performs the best when its size amounts to only 300 sentences.

5. CONCLUSION

In this study, a dependency annotation scheme was proposed for Indonesian by extending and chunking an adapted version of the SD annotation scheme. By training and testing an MST parser on our adapted and extended annotation schemes, we confirmed that extension and chunking increased the accuracy and completeness of the adapted annotation scheme. The results demonstrated that our annotations were useful in extracting dependency features for correcting preposition errors in real learner data. We further evaluated the annotation scheme to correct preposition errors using larger amounts of training data.

Our experimental results demonstrate that artificial training data using syntactic features extracted from dependency annotated sentences outperform data that do not make use of syntactic features. We plan to continue this work to improve our annotation scheme and extract more features to solve other NLP problems in Indonesian.

6. ACKNOWLEDGEMENT

We would like to thank the unknown reviewers, Erlyn Manguilimotan, and Prof. David Sell for their valuable discussion and comments. This study is supported in part by the DGHE, Republic of Indonesia under BPPLN scholarship Batch 7 fiscal year 2012-2015.

7. REFERENCES

Buchholz, S., Marsi, E., 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. *In*: Proceedings of the 10th Conference on Computational Natural Language Learning. Association for Computational Linguistics, Stroudsburg, USA, pp. 149–164

Cahill, A., Madnani, N., Tetreault, J., Napolitano, D., 2013. Robust Systems for Preposition Error Correction using Wikipedia Revisions. *In*: Proceedings of the Conference of the

North American Conference on Chinese Linguistics: Human Language Technology. Association for Computational Linguistics, Atlanta, Georgia, pp. 507–517

Dahlmeier, D., Ng, H.T., 2011. Grammatical Error Correction with Alternating Structure Optimization. *In*: Proceedings of the 49thAnnual Meeting of the Association for Computational Linguistics: Human Language Technology -Volume 1. Association for Computational Linguistics, Stroudsburg, USA, pp. 915–923

de Marneffe, M., Manning, C.D., 2008. Stanford Typed Dependencies Representation. *In*: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation. Association for Computational Linguistics, Stroudsburg, USA, pp. 1–8

Green, N., Larasati, S.D., Zăbokrtský, Z., 2012. Indonesian Dependency Treebank: Annotation and Parsing. *In*: Proceedings of the 26th Pacific Asia Conference on Language Information and Computation. Faculty of Computer Science, Universitas Indonesia, Bali, Indonesia, pp. 137–145

Han, N., Tetreault, J., Lee, S., Ha, J., 2010. Using an Error-annotated Learner Corpus to Develop an ASL/AFLError Correction System. *In*: Proceedings of the 7th International Conference on Linguistic Resources Evaluation. European Language Resources Association, Valletta, Malta, pp. 763–770

Irmawati, B., Komachi, M., Matsumoto, Y., 2016a. Towards Construction of an Error-corrected Corpus of Indonesian Second Language Learners. *In*: Francisco Alonso Almeida, Ivalla Ortega Barrera, Elena Quintana Toledo, Margarita Sánchez Cuervo (Eds), Input a Word, Analyse the World: Selected Approaches to Corpus Linguistics. Cambridge Scholars Publishing, Newcastle upon Tyne, United Kingdom, pp. 425–443

Irmawati, Budi, Shindo, Hiroyuki, Matsumoto, Yuji, 2016b. Exploiting Syntactic Similarities for Preposition Error Correction on Indonesian. *In*: Proceedings of the 5th Workshop on Spoken Language Technologies for Under Resource Languages. International Research Institute Multimedia, Information, Communication & Applications, Jogjakarta, Indonesia, pp. 214–220

Irmawati, Budi, Shindo, Hiroyuki, Matsumoto, Yuji, 2017. Generating Artificial Error Data for Indonesian Preposition Error Correction. *International Journal of Technology*, Volume 8(3), pp. 549–558

Kamayani, M., Purwarianti, A., 2011. Dependency Parsing for Indonesian. *In*: International Conference on Electrical Engineering and Informatics, pp. 1–5

Kübler, S., McDonald, R., Nivre, J., 2009. *Dependency Parsing*. Morgan & Claypool Publishers

Larasati, S.D., Kuboň, V., Zeman, D., 2011. Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus. *In*: Proceedings of the 2nd International Workshop Systems and Frameworks for Computational Morphology. Zurich, Switzerland, pp. 119–129

McDonald, R., Lerman, K., Pereira, F., 2006. Multilingual Dependency Analysis with a Two-stage Discriminative Parser. *In*: Proceedings of the 10th Conference on Computational Natural Language Learning. Association for Computational Linguistics, Stroudsburg, USA, pp. 216–220

Mizumoto, T., Komachi, M., Nagata, M., Matsumoto, Y., 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. *In*: Proceedings of the 5th International Joint Conference on Natural Language Processing. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pp. 147–155

Nivre, J., de Marnefe, M., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D., 2016. Universal Dependency Volume 1: A Multilingual Treebank Collection. *In*: Proceedings of the 10th International

Conference on Linguistic Resources Evaluation. European Linguistics Resources Association, Portorož, Slovenia, pp. 1659–1666

Nivre, J., Hall, J., Kübler, S., McDonald, R.T., Nilsson, J., Riedel, S., Yuret, D., 2007. The Conference on Computational Natural Language Learning 2007 Shared Task on Dependency Parsing. *In*: Proceedings of the Joint Conference on Empirical Method on Natural Language Processing and the Conference on Computational Natural Language Learning. Association for Computational Linguistics, Prague, Czech Republic, pp. 915–932

Quasthoff, U., Richter, M., Biemann, C., 2006. Corpus Portal for Search in Monolingual Corpora. *In*: Proceedings of the 5th International Conference on Linguistic Resources Evaluation. Genoa, pp. 1799–1802

Rozovskaya, A., Roth, D., 2010. Generating Confusion Sets for Context-sensitive Error Correction. *In*: Proceedings of the 2010 Conference on Empirical Method on Natural Language Processing. Association for Computational Linguistics, Stroudsburg, USA, pp. 961–970

Sneddon, J.N., Adelaar, A., Djenar, D.N., Ewing, M.C., 2010. Indonesian: *A Comprehensive Grammar*. Routledge, London, United Kingdom

Stack, M., 2005. Word Order and Intonation in Indonesian. *In*: Lexical Semantic Ontology Working Papers in Linguistics 5: Proceedings of Workshop in General Linguistics. Milan, Italy, pp. 168–182