DBpedia Entities Expansion in Automatically Building Dataset for Indonesian NER

Ika Alfina, Ruli Manurung, and Mohamad Ivan Fanany
Machine Learning and Computer Vision Laboratory
Faculty of Computer Science Universitas Indonesia
Depok, Indonesia
{ika.alfina, maruli, ivan}@cs.ui.ac.id

***Abstract*—Named Entity Recognition (NER) plays a significant role in Information Extraction (IE). In English, the NER systems have achieved excellent performance, but for the Indonesian language, the systems still need a lot of improvement. To create a reliable NER system using machine learning approach, a massive dataset to train the classifier is a must. Several studies have proposed methods in automatically building dataset for Indonesian NER using Indonesian Wikipedia articles as the source of the dataset and DBpedia as the reference in determining entity types automatically. The objective of our research is to improve the quality of the automatically tagged dataset. We proposed a new method in using DBpedia as the referenced named entities. We have created some rules in expanding DBpedia entities corpus for category person, place, and organization. The resulting training dataset is trained using Stanford NER tool to build an Indonesian NER classifier. The evaluation shows that our method improves recall significantly but has lower precision compared to the previous research.**

Keywords—NER; automatic tagging; DBpedia

I. INTRODUCTION

Named Entity Recognition (NER) is a subtask in Information Extraction (IE) that detecting the occurrence of named entities (NEs) in text and classifying each NE into a particular category. NER is a core component in IE that will be used by many other advanced components, such as semantic annotation, question answering, ontology population and opinion mining. Since introduced in 1996, at the 6th MUC conference, NER research has been developed [1]. In general, NER researchers use one of the two approaches: knowledge engineering and machine learning methods. Knowledge engineering approaches utilized expert knowledge to recognize named entity in the text. This method is usually rule-based. In machine learning techniques, we train a computer program to identify named entities [2].

Several studies have been conducted in machine learning based NER. In [3], a NER system based on Hidden Markov Models (HMM) was developed. The system can recognize and classify names, dates, times, and numerical quantities. A NER system that used a semi-supervised learning algorithm using conditional random fields (CRFs) was presented by [4]. In [5], a NER system based on CRF called Stanford NER CRF[1] was created. They made the library available online so that other researchers can utilize the tool.

NER for the Indonesian language has been developed by several studies [2, 6, 7, 8, 9]. In [6], an Indonesian NER that used the machine learning approach was presented. They use association rules method as the algorithm, and as dataset they use 55 manually labeled news articles [6]. Reference [2] presented a new rule-based method that use a set of rules to retrieve the contextual, morphological, and part of speech information from Indonesian texts to detect and classify named entities. As the dataset, they also manually labeled 802 sentences from Indonesian newspaper article. Later, in [7] another Indonesian NER system using association rules mining and co-reference resolution was proposed. As the corpus, they used 100 papers from online newspaper and the data were also tagged manually. From these three previous studies in Indonesian NER [2, 6, 7], they use corpus that manually marked, and the size of the corpus is very small.

We need a large training dataset to build a robust Indonesian NER system when using the machine learning approach. To address this problem, Luthfi et al. [8] and Leonandya et al. [9] proposed methods to automatically build training dataset for Indonesian NER. They use Wikipedia[2] articles in the Indonesian language as the source corpus and DBpedia Indonesia[3] as the named entities reference. Both used Stanford NER Classifier tool to build Indonesian NER model. Both studies reported that the quality of the resulting tagged dataset is still unsatisfactory since the recall is very low [8] or the F1-score still needs to be improved [9]. We will discuss about their studies in Section II.

The objective of our research is to improve the quality of the automatically tagged dataset for Indonesian NER. We propose a new method in using DBpedia instance types corpus as the reference for named entity types. We have identified the characteristics of named entities that recorded in DBpedia for each type (person, place, and organization) and developed rules to expand the corpus to improve recall.

The contributions of this study are:

- A new method in utilizing DBpedia as named entities type reference. A set of rules in expanding names for

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

[2] https://id.wikipedia.org/

[3] http://id.dbpedia.org/wiki/

 978-1-5090-4629-4/16/$31.00

person, place, and organization types had been defined that can be applied to any language.

- A better quality automatically tagged dataset in term of recall for future Indonesian NER research.

This paper is organized as follows. Section II describes the latest research in building dataset automatically for Indonesian NER system. Our proposed method is explained in Section III. Section IV presents the experiment results and evaluation. Finally, we discuss about the conclusion and the future work in Section V.

II. Related Works

In this section, we discuss more detail about two previous studies in automatically building tagged dataset for Indonesian NER. Luthfi et al. [8] proposed a method to build automatically tagged training dataset using Wikipedia articles as the source corpus and DBpedia as the reference in determining the type of the entities. They defined 4 categories for named entities: person, place, organization and other. More than 300,000 Indonesian Wikipedia articles were processed to produce the dataset that automatically labeled. A gold standard dataset to evaluate how well the quality of the automatically tagged dataset was also created. The gold standard was built by randomly choose 68 Indonesian Wikipedia articles and manually tagged each word in the articles. Experiments show that the NER model that built using this automatically tagged training dataset has a very good precision but with very low recall when evaluated with the gold standard testing dataset. For person category, the recall is only 65 of 879 (7.39%) and for place category, is only 75 of 656 (11.43%). There is no information about the recall for organization category [8].

The methods used by [8] in tagging the article are as follow. First, the candidates of named entities in the article are extracted. A phrase becomes a candidate if it has hyperlink or if it is composed of words started with a capital letter. After that, DBpedia is inquired about the type of the candidate. DBpedia records entities and their corresponding types. The type of the candidate is determined based on exact match with entry in DBpedia. Luthfi et al. observed that the very low recall is caused by "phrase problems". The phrase that only a subset of the corresponding phrase in DBpedia is being incorrectly classified [8].

Leonandya et al. [9] built Indonesian NER system using similar approach with [8]. There are three main differences between them. First, [8] used supervised learning approach, while [9] used semi-supervised learning. Second, [8] accessed DBpedia corpus online through DBpedia web service and communicate via SPARQL, while [9] constructed an offline corpus of only person, place, and organization named entities from DBpedia Indonesia corpus. Entities of the other types are discarded. Third, [8] utilized the hyperlinks and the phrases composed of words started with uppercase as the candidates of named entities, while [9] only use the later. The method used by [9] is tested using another gold standard that is different with [8]. The only performance measure reported is the overall F1-score for all types with the best performance of 31.96%. They stated that the bad quality of automatically tagged dataset is one of the causes of the poor performance [9].

III. Proposed Method

In this research, we propose a new method in matching the phrases in Wikipedia article and the corresponding phrases in DBpedia corpus in order to improve the quality of the automatically tagged dataset for Indonesian NER system. Our research is similar with [8, 9] that we use Wikipedia articles in the Indonesian language as the data source, DBpedia as the named entity reference and Stanford NER CRF library to build Indonesian NER model. We use supervised learning approach as [8] did, but in selecting the candidate named entities we follow [9] approach that use only the phrases composed of words begin with uppercase. We also use offline DBpedia corpus like [9] did.

A. System Design

Fig. 1 shows the main component of our NER system. The input to the system is a dump of Indonesian Wikipedia articles. This corpus is preprocessed to produce a set of sentences in Indonesian. After that, these sentences are automatically tagged to construct the training dataset. This dataset is further processed by the Stanford NER Classifier to produce Indonesian NER model.

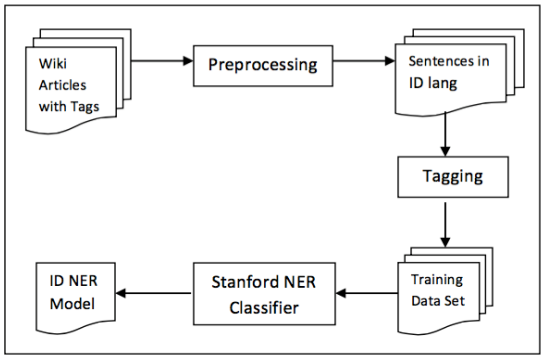

Fig. 1. NER system components

Fig. 2 shows the detail processes in the preprocessing phase. We follow methods used by [9] in preparing the dataset.

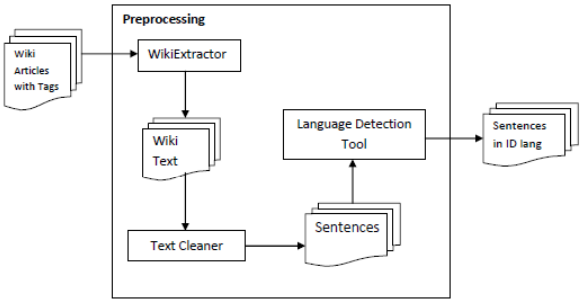

Fig. 2. Preprocessing Components

 978-1-5090-4629-4/16/$31.00

The method consists of 3 steps. First, using WikiExtractor[4] tool we extracted only text part of Wikipedia articles that originally have XML tags, tables or figures. Second, Text Cleaner removes the remaining tags from each article and transforms the paragraphs into sentences using NLTK[5] library [10]. Third, as there is a possibility that the Indonesian Wikipedia articles still contain text in another language, we filter such text using language detection tool[6]. The output of this phase is a set of sentences in the Indonesian language.

Fig. 3 below shows components of the tagging phase. First, we use Stanford NER Tokenizer library to transform file of sentences into file of tokens, the format used by Stanford NER Classifier. After that, the Entity Tagger automatically tagged each token in the dataset using DBpedia as the reference. We created the expanded version of DBpedia entities in order to enrich the corpus. The construction of the Expanded DBpedia is the main contribution of our research.

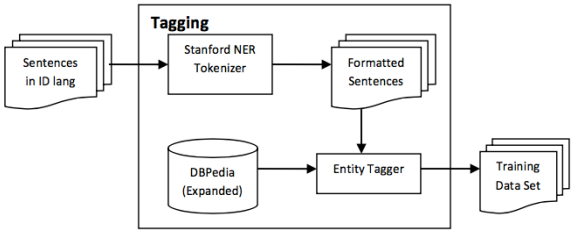

Fig. 3. Tagging Components

B. DBpedia Entities Expansion

We have examined DBpedia Instance Types corpus that contains entity names and their corresponding entity types. The corpus is a RDF[7] (Resource Description Framework) file. For each entity type (person, place, and organization), we create a separate file that contains only entity names of that type by parsing the RDF file. After that, for each type we analyzed the characteristics of the entity names.

We identified 11 categories for person names, 5 categories for place names, and 5 categories for organization names, as follows:

1. Categories of person:
   a. Standard name, composed of words started with uppercase, e.g.:
      Muhammad Hatta
      Ignasius Jonan
   b. Name with roman number, e.g.:
      Hamengkubuwana IX
      Pakubuwana VI
   c. Name that has single capital letter, e.g.:
      I Made Mangku Pastika
      Kenny G
   d. Name that has single capital letter followed by period, e.g.:
      M. S. Kaban
      R. E. Martadinata
   e. Name that contains word "bin" or "binti", e.g.:
      Ali bin Abi Thalib
      Fatimah binti Muhammad
   f. Name that contains words that relate the person with a place, such as "from", "dari", "van", "von", e.g.:
      Otto von Bismarck
      Elizabeth II dari Britania Raya
   g. Name that followed by title, usually using words like "der", "de", or "dos", e.g.:
      Noli de Castro
      Juan Silveira dos Santos
   h. Name with comma, followed by description, e.g.:
      Martin Luther King, Jr.
      Catherine, Duchess of Cambridge
   i. Name followed by description in parenthesis and is a valid person, e.g.:
      Abdul Rahman Saleh (jaksa)
      Indro (Warkop)
   j. Name followed by description in parenthesis and the description confirmed that it is not a valid person, e.g.:
      Jamrud (grup musik)
      Wings (grup musik Malaysia)
   k. Name that incorrectly label as person but with no description, e.g.:
      Sheila on 7
      Project Pop

2. Categories of place name:
   a. Standard name, e.g.:
      Bali
      Jawa Timur
   b. Name followed by description in parenthesis, e.g.:
      Liverpool (kota)
      Rijswijk (Holland Selatan)
   c. Name contains words often excluded when mentioned in the sentences, like "kabupaten", "kota", e.g.:
      Kabupaten Purbalingga
      Kota Depok
   d. Name with commas, contains other places that is the greater region of the place e.g.:
      Kota Kuala Simpang, Aceh Tamiang
      Marunda, Cilincing, Jakarta Utara
   e. Name contains common words, such as "panjang", "penularan", e.g.:
      Panjang, Bandar Lampung
      Penularan, Laweyan, Surakarta

3. Categories of organization name:
   a. Standard name, e.g.:
      Universitas Indonesia
      Google
   b. Name with periods, e.g.:
      Universitas Persada Indonesia Y.A.I
      A.C. Milan
   c. Name with commas, e.g.:
      Universitas California, Riverside
      MIS Al Hidayah, Jl. Jati Bening

[4] https://github.com/attardi/wikiextractor
[5] http://www.nltk.org/
[6] https://github.com/shuyo/language-detection
[7] https://www.w3.org/RDF/

 978-1-5090-4629-4/16/$31.00

d. Name contains word "di" (at), e.g.:
Kedutaan Besar Amerika Serikat di Jakarta
Sekolah Dasar Islam Terpadu di Sumatera Barat
e. Name followed by description in parenthesis, e.g.:
Bundesliga_(sepak_bola_Austria)
3 (telekomunikasi)

After exploring the contents of DBpedia entities corpus, we observed that there is no standard in recording entity names, but majority comply with standard name type that named entity is a phrase composed of words begin with uppercase. We also found that generally an entity only represented once in DBpedia. For example, in DBpedia Indonesia, Depok is recorded as "Kota Depok", Jakarta as "Daerah Khusus Ibukota Jakarta", and Pasar Minggu as "Pasar Minggu, Jakarta Selatan". There are no entries for "Depok", "Jakarta", or "Pasar Minggu" only. This will cause any occurrences of "Depok", "Jakarta" or "Pasar Minggu" in a sentence will not be tagged as a "Place", but as "Other" type. The limitation of [8] and [9] methods is: they do not handle this situation when a candidate phrase is a subset of the corresponding phrase in DBpedia. We decided to enrich DBpedia corpus by expanding the entities.

Fig. 4 shows our approach in expanding DBpedia entities that consists of 5 steps. First, DBpedia Instance Types corpus is parsed to construct 3 named entities files, one for each entity type, we named them as DBpedia Person, DBpedia Place, and DBpedia ORG. Second, Name Cleansing component removed invalid entries from the corpus. Third, the valid names are normalized before being expanded. The normalization rules for each entity types are different. Fourth, we expanded the names using specific rule for each type. Finally, the new names are validated. Every new name that consists of one word and meets certain criteria is removed from the final corpus. We use KBBI (Kamus Besar Bahasa Indonesia) and NLTK English corpus as the references in validating the names.

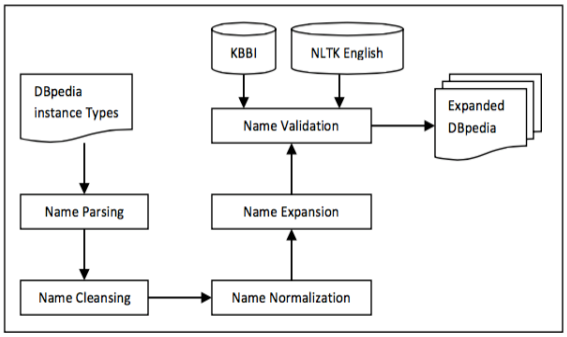

Fig. 4. DBpedia Entities Expansion

Further, we will explain the detailed approach in creating Expanded DBpedia for each named entity type. In explaining the method, we will refer to name categories defined in the previous paragraphs with their numbering. For example, we refer to the standard name for person as category 1-a or 3-c for organization names with commas.

1. Expansion of Person Names

The process in expanding DBpedia Person is as follows :

a. Name Cleansing
Names of category 1-j are removed since it is incorrectly labeled as a person.
Examples:
- *Jamrud (grup musik)* → *is removed because invalid*

b. Name Normalization
Names of category 1-e are normalized by splitting the names with delimiter "bin" or "binti". As the result, we got more than one names for each name.
Names of category 1-f, 1-g, 1-h, and 1-i are normalized by removing the description from the names.
Examples:
- Ali bin Abi Thalib → "Ali" and "Abi Thalib"
- *Elizabeth II dari Britania Raya* → *Elizabeth II*
- *Juan Silveira dos Santos* → *Juan Silveira*
- *Martin Luther King, Jr.* → *Martin Luther King*
- *Abdul Rahman Saleh (jaksa)* → *Abdul Rahman Saleh*

c. Name Expansion
First, we split the name with delimiter white space and then expand it using n-gram method. If a phrase consists of 3 words, we will split it using 1-gram, 2-gram until 3-gram.
Examples:
"Susilo Bambang Yudhoyono", is expanded to six entries: "Susilo Bambang Yudhoyono", "Susilo Bambang", "Bambang Yudhoyono", "Susilo", "Bambang", and "Yudhoyono".

d. Name Validation
For every name in expanded DBpedia Person that composed of only one word, we remove it if:
- The word is in KBBI or NLTK English corpus
- The words is a roman number
- The word contains periods
- The word has length of 1
- All the letters in the word are uppercase

2. Expansion of Place Names

The process in expanding DBpedia Place is as follows :

a. Name Cleansing
Since all names in DBpedia Place are valid locations, this step can be skipped.

b. Name Normalization
Names of category 2-b and 2-c are normalized by removing the description from the main name.
Examples:
- *Kota Depok* → Depok
- *Liverpool (kota)* → *Liverpool*

 978-1-5090-4629-4/16/$31.00

c. Name Expansion
Instead of splitting names with whitespace delimiter like for DBpedia Person, for DBpedia Place we use a comma delimiter. After that, we use n-gram to expand the names.
For example:

"Pasar Minggu, Jakarta Selatan" is expanded to three entries, "Pasar Minggu, Jakarta Selatan", "Pasar Minggu" and "Jakarta Selatan".

d. Name Validation
We employ the same rules with DBpedia Person in eliminating names that composed of one word only.

3. Expansion of Organization Names

The process in expanding DBpedia Place is as follows :

a. Name Cleansing
Since there is no name that incorrectly label as organization in DBpedia ORG, this step can be skipped.

b. Name Normalization
Names of category 3-e are normalized by removing the description from the names.
Examples:

- *Bundesliga (sepak bola Austria)* → *Bundesliga*
- *Universitas Teknologi Sulawesi (UTS) Makasar* → *Universitas Teknologi Sulawesi Makasar*

c. Name Expansion
Name of category 3-c and 3-d is expanded, by creating two new phrases: one without the comma or word "di"/"at" and another one by removing remaining words, starting from the occurrence of a comma or word "di"/"at".
Examples:

- "*Universitas California, Riverside"* → "*Universitas California Riverside"and* "*Universitas California"*

d. Name Validation
We employ the same rules with DBpedia Person in eliminating names that composed of one word only, except that we do not remove word that all of its letters are uppercase. An organization often has an acronym that composes of uppercases.

Table 1 shows the numbers of entities in the original DBpedia instance types corpus along with the number of named entities that we produced in Name Normalization and Name Expansion phases. The place entities are expanded almost 2.5 times, person entities are about 2 times and the organization type increased only 1.58%. This data also shows that we had removed a lot of person names after normalization phase. Those names mostly are from category 1-j. Unfortunately, for category 1-k we have not found good rules to remove them yet. For now, we just eliminate names with digit like "Dewa 19" or "Sheila on 7". The failure to eliminate these invalid names before Name Expansion phase will introduce a lot of invalid entries in final Expanded DBpedia.

TABLE I. NUMBER OF NAMED ENTITIES IN DBPEDIA CORPUS

Entity Type	*Original*	*Normalized*	*Expanded*
Person	17,749	17,352	36,514
Place	57,193	57,193	137,710
Organization	5,633	5,633	5,722
Total	80,575	80,176	179,946

C. Tagging Process

Entity Tagger on Fig. 3 reads the file of tokens and determines the type of each token using Expanded DBpedia as the reference. Entity Tagger scans tokens from the first row. If it finds token started with an uppercase letter, it will check whether the next token is also started with an uppercase letter. This process stops until it found a token that started with lowercase. This step will produce a phrase of the candidate named entity, e.g. "Presiden Joko Widodo".

Furthermore, the candidate phrase is expanded using n-gram method. For the "Presiden Joko Widodo", 6 phrases are created: "Presiden Joko Widodo", "Presiden Joko", "Joko Widodo", "Presiden", "Joko" and "Widodo". Types inquiry to Expanded DBpedia corpus is started from the longest phrase. If there are entries that exist in more than one corpus, e.g. "Setiabudi" exists in DBpedia Person and Place, their types will be set as "Other".

IV. EXPERIMENT AND EVALUATION

We evaluate the quality of our automatically tagged dataset by using it as the input for Stanford NER Classifier and test the performance of the model resulted. The Indonesian NER model was tested in three scenarios. First, we use 5-fold cross validation. Second, we use gold-standard testing dataset used by [8] and finally we compare the performance of models built using a different kind of DBpedia as mention in Table 1 to see the effect of DBpedia entities expansion.

A. Data Preparation

Indonesian Wikipedia dump[8] of 360MB are downloaded and preprocessed. Only articles that the title matches with one entry in DBpedia Person or Place or ORG are selected. Furthermore, the articles that have less than 5 paragraph are discarded. From the remaining articles, we choose first 3 paragraphs that have at least 50 words. After paragraphs are transformed into sentences, we select only the sentences that have at least 15 words, and no less than 5 of those words started with uppercase. These criteria are applied to have dataset that rich of named entities. At the end of the tagging phase, the automatically tagged dataset of 20,000 sentences are created.

B. The result of 5-fold cross validation.

Table 2 shows the results. It can be seen our methods work best for "Place", and has the worst performance for "Person".

[8] https://dumps.wikimedia.org/idwiki/latest/

 978-1-5090-4629-4/16/$31.00©

TABLE II. 5-FOLD CROSS VALIDATION

Entity Type	*Precision*	*Recall*	*F1-score*
Person	71.44%	47.11%	56.74%
Place	88.83%	81.49%	85.00%
Organization	80.40%	60.79%	69.13%
Total	85.06%	71.82%	77.86%

C. *The result using gold-standard*

We used the 20,000 tagged sentences dataset as training data and 68 articles gold standard used by [8] as the testing data. Table 3 shows the results. If we compare with the result of [8], the recall of our method for "Person" increased 4.8 times, and for "Place" was 4.77 times higher than [8]. Unfortunately, the rising of recall was paid by the declining of precision, 23% for "Person" and 19.88% for "Place". Our F1-score also exceeded the performance of [9] with the margin of 16%.

TABLE III. USING GOLD STANDARD

Entity Type	*Precision*	*Recall*	*F1-score*
Person	77.10%	35.50%	48.62%
Place	80.12%	54.51%	64.88%
Organization	89.66%	7.34%	13.58%
Total	79.31%	35.29%	48.84%

D. *Performance comparison using different version of DBpedia*

The 20,000 sentences are tagged three times to construct 3 dataset using the different version of DBpedia as named entity reference: the original one, the normalized and the expanded one. Fig. 5 shows the overall performance using the three dataset. Name normalization cause a slight increase in the recall (2.58%) and F1-score (3.37%), while the name expansion improves precision 3.41%, recall increased 2.8 times, and F1-score rose 2.17 times, compare to original DBpedia. This confirmed that our approach had produced a better dataset.

V. CONCLUSION AND THE FUTURE WORK

We have proposed a new method in automatically building tagged dataset to improve method use by [8] and [9], that we call DBpedia entities expansion. Some rules for entities expansion had been created for type person, place, and organization. Performance evaluation shows that our approach improves recall significantly, but the precision decreased compared to [8]. Our method also outperforms F1-score of [9].

To improve the precision, the next research should pay more attention on how to remove invalid entries from Expanded DBpedia, especially for the person type. In this research, we have not found good rules to expand the organization named entities. Adding a list of acronyms of organizations to DBpedia ORG could be tried in the future research.

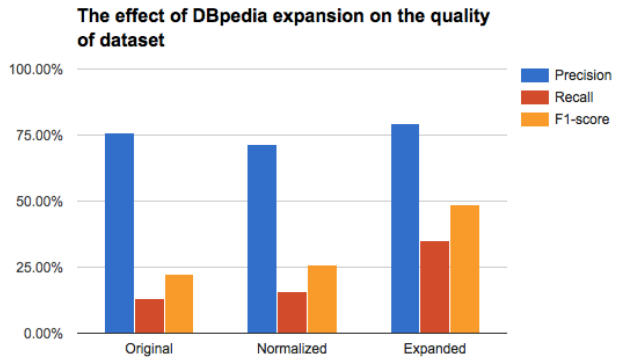

Fig. 5. The effect of DBpedia expansion

ACKNOWLEDGEMENT

This work is supported by Student Paper in Indexed Publication Grant funded by Directorate of Research and Public Services, Universitas Indonesia, Contract Number: 1853/UN2.R12/HKP.05.00/2016.

REFERENCES

[1] M. Marrero, et al., "Named Entity Recognition: Fallacies, challenges and opportunities," *Computer Standard and Interface*, vol.35, pp. 482–489, Jan. 2013.

[2] I. Budi and S. Bressan, "Application of association rules mining to Named Entity Recognition and co-reference resolution for the Indonesian language," *International Journal of Business Intelligence and Data Mining*, vol. 2 no 4, pp 426-446, Des. 2007.

[3] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel.," An algorithm that learns what's in a name," Machine Learning, 34(1-3):211–231, 1999.

[4] W. Liao and S. Veeramachaneni, "A Simple Semi-supervised Algorithm For Named Entity Recognition" presented at NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing pp 58-65, Boulder, Colorado, June 2009.

[5] J. R. Finkel, T. Grenager, C. Manning. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". 2005. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[6] I. Budi and S. Bressan, "Association Rules Mining for Name Entity Recognition," presented at the Fourth International Conference on Web Information Systems Engineering (WISE'03), 2003.

[7] I. Budi, et al, "Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach," presented at the 8th international conference on Discovery Science, 2005, pp 57-69.

[8] A. Luthfi, B. Distiawan and R. Manurung, "Building an Indonesian named entity recognizer using Wikipedia and DBPedia," presented at the Asian Language Processing (IALP) 2014 International Conference, Oct. 20-22, 2014.

[9] R. A. Leonandya, B. Distiawan and N. H. Praptono, "A Semi Supervised Algorithm for Indonesian Named Entity Recognition", presented at 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015), 2015

[10] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

 978-1-5090-4629-4/16/$31.00