Building an Indonesian Named Entity Recognizer using Wikipedia and DBPedia

Andry Luthfi, Bayu Distiawan, and Ruli Manurung
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
andry.luthfi@ui.ac.id, b.distiawan@cs.ui.ac.id, maruli@cs.ui.ac.id

***Abstract* - This paper describes the development of an Indonesian NER system using online data such as Wikipedia [1] and DBPedia [2]. The system is based on the Stanford NER system [8] and utilizes training documents constructed automatically from Wikipedia. Each entity, i.e. word or phrase that has a hyperlink, in the Wikipedia documents are tagged according to information that is obtained from DBPedia. In this very first version, we are only interested in three entities, namely: Person, Place, and Organization. The system is evaluated using cross fold validation and also evaluated using a gold standard that was manually annotated. Using cross validation evaluation, our Indonesian NER managed to obtain precision and recall values above 90%, whereas the evaluation using gold standard shows that the Indonesian NER achieves high precision but very low recall.**

Keywords-name entity recognition; stanford ner; wikipedia; dbpedia

I. INTRODUCTION

Named Entity Recognition is an important task in NLP. It serves as a first step in turning unstructured text into structured data, and has broad applications in news aggregation, question answering, and bioNLP [5]. Given an input sentence, an NER tagger identifies words that are part of a named entity, and assigns the entity type and relative position information.

II. BACKGROUND

Development of an Indonesian Named Entity Recognition system has been done before [2]. However, the problem that arises is the lack of training documents. It led the NER system to be not optimal. Typically, to develop NER, researchers must collect labelled documents manually. The collection of the manually labelled documents is a very time and resource-intensive process.

Therefore, in this research, we propose a method to obtain labelled training data automatically by utilizing resources that are already available online.

III. RESOURCES

The model that we build requires two core resources: the Indonesian version of Wikipedia and the DBPedia Indonesia web service [1]. These resources provide important stepping stones to produce the model. Basically the Wikipedia dump will provide the raw text while the DBPedia service will be used to collect information of the "entity" from the raw text. In the next section we will describe our method in this research.

A. *Wikipedia*

The Indonesian version of Wikipedia is our resource to provide rich named entities. It's contents are crowdsourced. Thus, we assume it involves heterogenous contributors, vary in writing style, and vary in describing something in a single article. It matches with real world conditions. We used the archive 'dump' made available by the Wikimedia Foundation. The total number of articles in the Indonesian version of Wikipedia is 564,736 on September, 2013.

Wikipedia provides a lot of articles concerning People, Places, or Organisations. For example, one article describes Indonesia's First President, Soekarno. That article provides information about where was he born, in this example, President Soekarno was born in Surabaya. On Wikipedia's article, the contributor will give a link (we subsequently call this as a *tagged phrase*) to another article that is possibly named Surabaya (for some article will have multiple URI under same content). This link will help us to determine that the words/phrases are a named entity.

Because it is based on human work, not all of the possible phrases are tagged by contributors. Basically there are two reasons: the article is either already mentioned in a preceding sentence or the contributor did not tagged the phrase, whether by choice or by mistake. Typically, an entity phrase is not tagged due to there being no relevant Wikipedia entry to link to or the contributor simply overlooked it. Either way, we propose a way to answer this issue.

As we know, Wikipedia provides a lot of information that has a lot of categories. But, not all of those categories can be decided as a named entity. In this research we are only interested on Person, Place and Organization categories. To make sure that the article which has been linked on word/phrase is on these categories, we will look up the information on DBPedia.

B. *DBPedia Webservice*

DBPedia is a community contribution that extracts structured information from Wikipedia and provides them via the web [3]. DBPedia Indonesia is a web application which provides extracted information from Wikipedia Indonesia [7]. A single page of information has a unique URI defined as a single entity. Single entity is referred to a single article in Wikipedia. For example, let us say there is an article named Soekarno under the URL http://id.wikipedia.org/wiki/Soekarno in Wikipedia, and

[1] http://id.wikipedia.org/
[2] http://id.dbpedia.org/wiki/

there exists one entity named Soekarno on DBPedia page under the URL http://id.dbpedia.org/resource/Soekarno.

On every resource page there is an attribute that tells us the type of the entity. Possible types of entities are listed at http://mappings.dbpedia.org/server/ontology/classes/. As for the entity Soekarno, the type of this entity is Person. It will help us to give a named entity on the candidate phrase which is produced by the Wikipedia dump.

For some entities it is named under a specific type like Populated Place rather than Place. It indicates that there are possible sub-classes under a single type. The ontology list tells us that there are 160 sub-class types under Person, 79 sub-class types under Organisation, and 130 sub-classes under Place. For these sub-classes, we will simply annotate them using the parent class (Person, Place, or Organization).

Another issue is that some Wikipedia articles are just a redirected page for another page. This means that there exists one entity for this redirect page. We need to conduct further processing to determine the entity of this page by searching for the original page.

The Indonesian version of DBpedia describes 140,993 entities. Those entities consist of 19,567 Persons, 57,702 Places, 5,773 Organisations, and 10,711 Works [7]. For comparison, the English version of DBPedia has 4,004,478 entities.

IV. Building Indonesian NER Model

This section will describe the process of creating the automatically tagged data for Indonesian NER training documents. First, we will process the raw text obtained from Wikipedia to produce formatted training data for the Stanford CRF-NER [4]. The processing will be done using WikiParser.

Wikipedia does not only provide raw text for each article but also provides phrases-associated set of link. Of course these phrases happen to exist in this text. It also provides wiki text. This text is more similar to web version. All phrases which had link and some HTML tag is already provided in this version.

After the formatted training data are created, we will use that data to train the Indonesian NER model using Stanford NER tool. Figure 1 shows the entire process from processing Wikipedia documents to generating the Indonesian NER Model.

A. *Text Cleaner*

The raw text provided by Wikipedia still has some unnecessary content such as Infobox, tables, etc. What we are trying to do is to remove those contents from the text. After that, texts will be separated from each paragraph. Basically, the module only tries to remove unnecessary whitespace and maintain its existence whether it helps to classify some text as a paragraph.

It is important, if there exists one paragraph which does not have at least one named entity then it should be removed. It also means that if a single article does not have any named entity then it should be ignored. Even worse, if a single article does not have any link then it also should be ignored. We can use the information of set of links given by the Wikipedia dump to filter the articles.

It should be noted that removing consideration based on named entity will be performed on Simple Tagger and Heuristic Tagger. At this point, the module only tries to preserve the paragraph's structure.

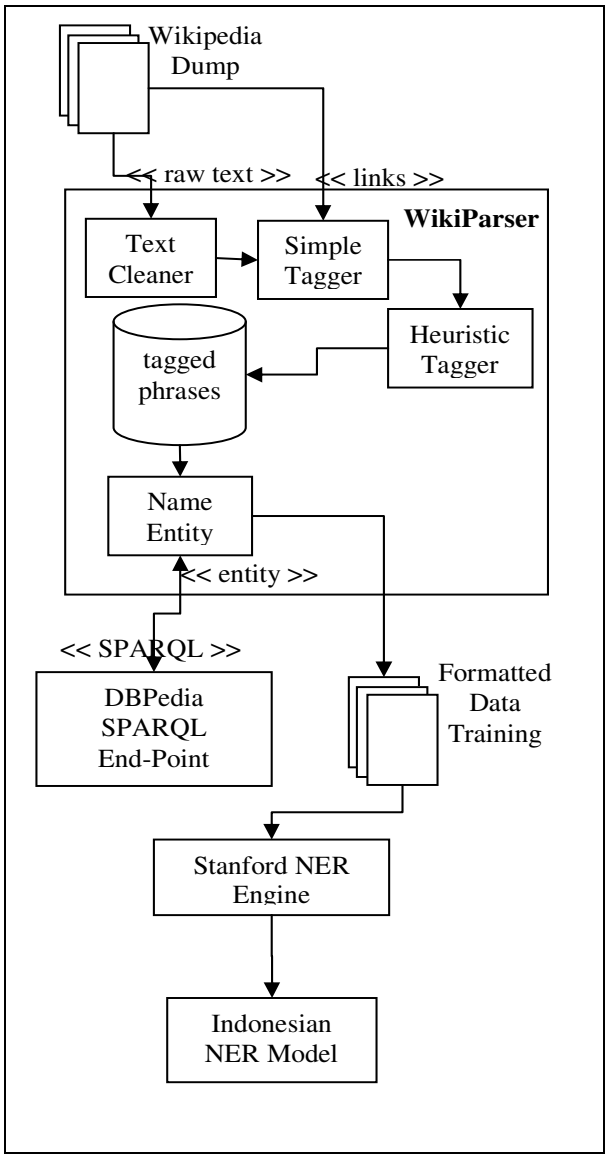

Figure 1. Building Indonesian NER Model Overview

B. *Language Detector*

Language has a major role in content removing consideration. We try to build a corpus based on the Indonesian language. Because of that, we use the Indonesian version of the Wikipedia dump. But, there is a possibility that some articles were written in a language other than Bahasa Indonesia.

In some cases, the article is just copied from the English version of Wikipedia or partially written in another language, which means the other part is written in Indonesian. To make sure that the Indonesian NER models only use the Indonesian articles, we will remove all documents that are just copied, or remove the paragraphs which are not written in Bahasa Indonesia. To identify the language of an articles or paragraph, we use a language detector [6].

C. Simple Tagger and Heuristic Tagger

At this point, the produced text does not provide the phrases which are likely to contain named entities. This module will provide that information by taking a set of links given by the Wikipedia Dump and tagging into phrases on text.

We can use the wiki text to skip this step, but usually wiki text only tags the first occurrence of phrase. The next occurrence of the phrase, which is very likely to be the same, would not have any tag since it is not hyperlinked. For example: the first phrase of Ernest Douwes Dekker in article contains link to http://id.wikipedia.org/wiki/Ernest_Douwes_Dekker. The next occurrence of Ernest Douwes Dekker would not have any hyperlink. So, we take the raw text as the initial resource and try to self-tag all occurrence of phrases that have the same string as the last part of the link. We call this the Simple Tagger.

We found that many phrases which are likely to yield named entities are not hyperlinked in Wikipedia. So we have to make those phrases as candidate named entities. We cannot give all possible phrases in a single article to DBPedia service because it is too impractical. Therefore, we propose a Heuristic Tagger to grab most likely candidates that would yield named entities. The heuristic tagger basically makes all the phrases that are in each starting word using uppercase is likely the best candidate.

The paragraph that does not have any candidate named entity or does not have any resulting tag via Simple and/or Heuristic Tagger will be completely removed by this module. Filtering will make the training data more reliable in order to build a better model.

D. Naming Entity via DBPedia Service

Tagged phrases produced by the Simple and Heuristic taggers will be given to the DBPedia service to find out the ones containing named entities. The service communicates via SPARQL [9], thus we must construct a *question* that is to be answered by DBPedia.

```
select distinct(?ins) ?type where {
   ?ins rdf:type ?type.
   ?ins rdfs:label ?label.
   FILTER regex(str(?label),"^Soekarno$").
}
LIMIT 100
```

Figure 2. SPARQL for type entity for single label

The *question* is, select all instances with its type where the label of its type is the same as the given phrase. The service will then respond with a list of instances and their ontology class or entity type. After obtaining the entity type (named entity) the module will lookup the ontology classes to check whether the entity type is a sub-class of Place, Person or Organisation. Otherwise, it will be classfied as O (other entity). If the response is an empty list then it definitely does not exist a matching label in DBPedia, thus it is considered as O.

For some cases the O is not the end of the result. As described before, there are some redirect pages for some articles. For examle, Ernest Douwes Dekker will provide Person as a result. Yet, unfortunately Dr. Douwes Dekker will provide Thing as a result, which is O. This phrase is actually an entity of a redirected page for Ernest Douwes Dekker. So we can result this phrase the same as Ernest Douwes Dekker by getting the originating URI or label via wikiPageRedirects.

E. Formatter

Other untagged phrases will be given O as its named entity. After getting all named entities for all phrases, the Formatter will print these out in files that complywith the default format of the Stanford CRF-NER features map, i.e. word and named entity per line.

This formatter will ensure that each line only consists of two words separated by a space. The Stanford CRF-NER system will assume that the second word of each line is the named entity. Therefore phrases consisting of multiple words will be separated into multiple lines.

F. Stanford NER

Having been able to automatically tag the Wikipedia articles, we do the next process, i.e. to build the Indonesian NER model. This model was developed by Stanford NER[3] tools. The tool uses a general implementation of Conditional Random Field (CRF) sequence model. In building the Indonesia NER Model, we just use the default settings of this tool. We have not yet experimented on adjusting any parameters on this tool. We also did not add any other information such as a gazetteer for Indonesian documents.

V. Experiments and Evaluation

In this experiment, we have collected 10.000 articles (8MB) in Indonesian that have been automatically tagged. In automatic tagging, we only include three entities, namely Person, Place, and Organization. From the collected data, we get 1.723 unique entities; 657 for Person entity, 788 for Place entity, and 278 for Organization entity.

From this data, we conducted several experiments to determine the effect of the amount of training data used. We build four different models of NER, i.e. the model with 1MB, 2MB, 4MB, and 8MB training data.

To measure the accuracy of this system, we perform 10-fold cross validation. The result of this evaluation is quite good. Figure 3 shows that there is an increase in precision and recall on the experiment when we add the training data. Using all the training data we had (8MB), Indonesian NER managed to obtain precision and recall above 90%.

In addition to using fold cross validation, we also made a gold standard to test how well the results of the automatic tagging are. We created a gold standard by taking 68 random Wikipedia documents and manually tagging each entity in the article. From this experiment, the system is only able to identify 65 out of 879 Person entities and 75 out of 656 Place entities that appear in the testing documents. From 65 Person entities and 75 Place entities, all of them are correctly identified as their respective entity types. Thus we achieve 100% precision but very low recall.

After further investigation, we discovered that for a lot of phrases that should be single entities within the

[3] http://nlp.stanford.edu/software/stanford-ner-2014-01-04.zip

Wikipedia article, only some of the words of the phrase have a hyperlink. For example, the entity "J.F. Kennedy" in some texts in Wikipedia articles, only has the word "Kennedy" that is hyperlinked. This causes the automatic tagging process to lack recall power. Figure 3 shows an overview of the obtained results of the evaluation.

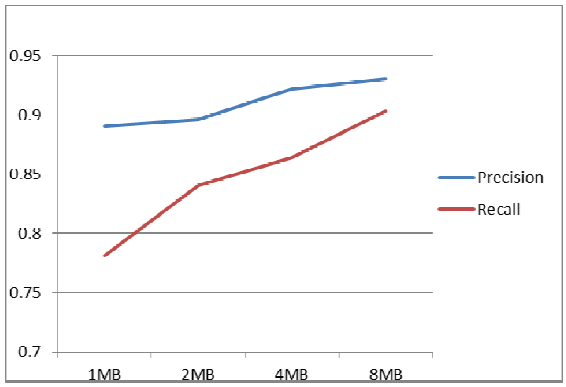

Figure 3. Evaluation Result Using 10-Fold Validation

VI. CONCLUSION AND FUTURE WORK

The results using cross fold validation show that the recall and precision produced by Indonesian NER is very high, but when the system is evaluated using the gold standard, Indonesian NER has high precision, but has very low recall.

In the future, we will improve the automatic tagging process in order to overcome the phrase link problems. As mentioned above, there are a lot of articles on Wikipedia that give a hyperlink only to some words in an entity phrase. One idea that can be explored is to utilize a rule-based POS tagger to split words/phrases in one sentence, so if there is one word in a phrase that has a link, all the words in the phrase will also be tagged as single entity. Another idea to overcome this problem is to maintain every title in Wikipedia articles than match it to every word and phrase in the article's content.

REFERENCES

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann. "DBpedia-A crystallization point for the Web of Data". 2009. Web Semantics: Science, Services and Agents on the World Wide Web, 7(3), 154-165.

[2] I. Budi, S. Bressan, G. Wahyudi, Z.A. Hasibuan, B.A. A. Nazief. "Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach". Discovery Science 2005: 57-69.

[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P.v. Kleef, S. Auer, C. Bizer. "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". 2014

[4] 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. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf

[5] M. Wang, W. Che, C.D. Manning. "Effective Bilingual Constraints for Semi-Supervised Learning of Named Entity Recognizers". Twenty-Seventh AAAI Conference on Artificial Intelligence 2013: 2

[6] N. Shuyo. "Language Detection Library for Java". 2010. http://code.google.com/p/language-detection/

[7] R.A. Prasetya. "Pengembangan DBPEDIA Indonesia". 2013. 56-58

[8] S. Dingare, J. Finkel, M. Nissim, C. Manning, and C. Grover. "A System For Identifying Named Entities in Biomedical Text: How Results From Two Evaluations Reflect on Both the System and the Evaluations". In The 2004 BioLink meeting: Linking Literature, Information and Knowledge for Biology at ISMB 2004.

[9] S. Sizov. SPARQL. Slide Semantic Website,Web Science&Technologies, University of Koblenz, Landau, Germany. 2012