The Combination of YAKE and Language Processing for Unsupervised Term Extraction Ontology Learning

Rajif Agung Yunmar*, Andika Setiawan and Hartanto Tantriawan
Informatics Engineering, Institut Teknologi Sumatera, Lampung, Indonesia
* E-mail: rajif@if.itera.ac.id

Abstract. Information that is spread on the internet is available in the form of unstructured texts that can only be understood by humans, but difficult for machines to understand. Ontology learning is a method that can transform information in unstructured forms, into information that can be understood by machines, namely ontology. In ontology learning, the extraction term is one of the stages that must be passed. This stage produces important terms related to a topic before finally being grouped in certain classes. In this study, the term extraction method used is YAKE. The contribution of this research is to investigate the effects of language processing such as stemming and stopword removal when combined with the YAKE method at the term extraction stage. The language processing technique is then applied to the corpus of the test, after that as the input to the YAKE term extraction. Testing is conducted with several scenarios, namely: plain YAKE, stemming+YAKE, stopword removal+YAKE, or a combination three of them. These extraction scenario are evaluated by expert for measure the term correctness. The research shows that the combination of stopword removal+YAKE provide the best accuracy of 48%.

Introduction
The information contained in various sources on the internet is widely presented by prioritizing aspects of presentation, namely how the information is presented interestingly, or how it to be easily understood way by humans. This information is usually written in unstructured text and in the form of natural language. Meanwhile, something that equally important needed, namely how to make the information available on the internet not only be understood by humans but also by machines. Information that understood by machines can be used for various purposes, for example, being a knowledge base for reasoning, sharing knowledge between machines, semantic search, information visualization, and so forth.

Ontology is able to represent information based on semantic concepts that consist of objects, properties, and relationships between objects that occur in certain domains [1]. The use of ontology can produce a knowledge base that can not only be understood by humans, but can also by machines. Thus, the information provided by the system that uses the ontology knowledge base can be more precise and relevant [2]. Ontology learning is a method that can transform information from unstructured form, into a form that can be understood by machines. Information extraction using ontology learning is very necessary, considering that there are currently more than 60 billion web pages [3]. It cannot be imagined how many resources are needed to build ontologies manually. Manually ontology development on a larger scale can make the results not accurate, development process can consume more time, tedious, and costly [4].
One stage that passed by ontology learning is the term extraction. This stage produces important terms related to a topic before finally being grouped in certain classes that make up an ontology. YAKE is one of the methods of unsupervised term extraction which has higher accuracy than other unsupervised methods [5]. In addition, the YAKE is not bound by specific languages or domains, which makes it can be suitable for implementation in any language. To improve the accuracy of the term extraction, this research combines the YAKE term extraction method with language processing methods such as stemming and stopword removal.

Related Works
There are two approaches to do the term extraction, i.e. supervised and unsupervised. The difference between them is the availability of an annotated corpus, where the supervised term extraction method requires it, however unsupervised does not [6]. Because unsupervised term extraction does not need an annotated corpus, that may not always be available on a variety of topics. Furthermore, unsupervised term extraction in ontology learning is divided into two parts, namely using a linguistic and statistical approach [7].

The term extraction with natural language processing has been proposed by several researchers, including those conducted by Hippisley et al. which uses the Hypernym Syntax structural technique [8], i.e. words or phrases whose semantic groups are included in other words. Other studies such as that of Belal and Hala [9], and Fraga et al. [10] uses the seed word to extract terms. Seed Word ensures that each extracted term is of relevant value and semantically close to the given concept. The disadvantage of this method is that it requires additional corpus for specific domains.

Another approach at the term extraction stage is to use a statistical approach. The advantage of a statistical approach is the independence of various language complexity so it can be flexible to use [7]. This approach was proposed by several researchers, including by Guo et al. [11] which uses the Contrastive Analysis technique to filter terms from concepts that are not relevant to the corpus. Another approach was proposed by Drymonas [12], Yang et al. [13], and Chandu et al. [14] uses the C/NC Value method which uses multi-terminology extraction. C/NC value accepts input of various kinds of multi-word terminology and then returns the respective score of the existing words. Meanwhile, the YAKE method proposed by Campos et al. [5] promises better results than other statistical approaches.

In this study, ontology learning will be applied to the Indonesian language tourism domain. The YAKE method as an unsupervised term extraction is applied in consideration of the absence of an annotated corpus in the tourism domain, especially in the Indonesian language. As stated before, YAKE is an unsupervised term extraction method that is language independent. The contribution of this research is to investigate the effect of language processing such as stemming and stopword removal when combined with YAKE. Stopword removal is expected to simplify words and reduce unnecessary words in the corpus test documents, thus the term extraction produces higher quality terms.

Literature Study

1. YAKE Term Extraction
The YAKE method as a term extraction has several important steps that is performed on the test corpus, including text processing, feature extraction, individual term weighting, and candidate keyword generation [5].

1.1. Text Preprocessing
At the stage of text processing, the tokenization process is carried out by breaking up the test corpus into words based on spaces or special characters (brackets, commas, etc.).

1.2. Feature Extraction
At the feature extraction stage, each word is searched for weight based on its characteristics. Each existing word are measured by calculating certain features, including casing, word position, word
frequency, word relatedness to context, and how often a candidate word appears in various sentences in the document.

1.2.A. Casing (WCASE)
At this stage, the WCase feature of a word w will be weighted based on cases in letters, where words with capital or acronym prefixes tend to have higher relevance to the topic. For each word, YAKE calculates the maximum occurrence of words between the two-word types. Equation (1) illustrates the calculation of WCase weight.

\[ W_{\text{Case}} = \frac{\max(TF(U(w)), TF(A(w)))}{\log_2(TF(w))} \]  

where TF(U(w)) is the number of occurrences of the candidate word w which is preceded by a capital letter, and TF(A(w)) is the number of occurrences of the word w in the form of an acronym, while TF(w) is the frequency of occurrence of w.

1.2.B. Word Position (WPosition)
The next feature weight calculation relates to the position of words in a sentence, where words at the beginning of the sentence assumed to have higher relevance, especially for science and news articles. Equation (2) below shows the calculation.

\[ W_{\text{Position}} = \log_2(\log_2(2 + \text{Median}(\text{Sen}_w))) \]  

where Senw shows the position of a collection of sentences where the word w appears. The result of the calculation in Equation (2) is that the word w at the end of the sentence has a high WPosition value, while the word w which often appears at the beginning of the sentence has a low WPosition value. The lower the WPosition value, the more relevant the candidate word w is to the topic. For the record, the value 2 in Equation (2) above aims to guarantee that the WPosition value is always > 0.

1.2.C. Word Frequency (WFreq)
At this stage, the WFreq feature calculates the appearance of the candidate word w in a test corpus document. This calculation is based on the assumption that words that appear more often have higher relevance to the topic of the document. To prevent bias on high TF(w) values on long documents, the TF(w) value is divided by the mean of frequency (MeanTF) plus 1 multiplied by standard deviation (σ). It is intended that the TF(w) value be above the mean value of all terms in the document. Calculation of the weight of the WFreq feature can be seen in Equation (3).

\[ W_{\text{Freq}} = \frac{TF(w)}{\text{MeanTF} + 1 \times \sigma} \]  

1.2.D. Word Relatedness to Context (WRel)
At this stage measured the words that have characteristics such as stopword, namely by calculating the appearance of other words that appear together with the term w candidate, both on the right and to the left of the term w candidate. The more often it appears together with different words (both on the right and left), the more meaningless the word. WRel calculations can be calculated by Equation (4).

\[ W_{\text{Rel}} = \left( 0.5 + \left( WL \cdot \frac{TF(w)}{\text{MaxTF}} + PL \right) \right) + 0.5 + \left( WR \cdot \frac{TF(w)}{\text{MaxTF}} + PR \right) \]  

Equations:

1. \[ \frac{\max(TF(U(w)), TF(A(w)))}{\log_2(TF(w))} \]
2. \[ \log_2(\log_2(2 + \text{Median}(\text{Sen}_w))) \]
3. \[ \frac{TF(w)}{\text{MeanTF} + 1 \times \sigma} \]
4. \[ \left( 0.5 + \left( WL \cdot \frac{TF(w)}{\text{MaxTF}} + PL \right) \right) + 0.5 + \left( WR \cdot \frac{TF(w)}{\text{MaxTF}} + PR \right) \]
where WL and WR are the value of the occurrence ratio of various words that simultaneously appear with the term w candidate (both on the WL level and on the WR level). \( TF(w) \) is the frequency of occurrence of candidate terms w, and \( \text{MaxTF} \) is the maximum occurrence of a word in the document. Whereas PL and PR are the values of the occurrence ratio of various words that appear simultaneously appearing with the candidate term w and the \( \text{MaxTF} \). In practice, the higher the \( W_{\text{Rel}} \) value, the less relevant the word is to the topic of the document. Thus, the term candidate can be known as a stopword-like if he has a high \( W_{\text{Rel}} \) value.

1.2.E. Word DifSentence \( (W_{\text{DiffSentence}}) \)

At this stage, the \( W_{\text{DiffSentence}} \) feature is calculated by seeing how often a candidate word w appears in various sentences in the document. The calculation is done through Equation (5).

\[
W_{\text{DiffSentence}} = \frac{SF(w)}{\#\text{Sentence}}
\]

(5)

where \( SF(w) \) is the number of sentences where the candidate word w appears, and \( \#\text{Sentence} \) is the number of words contained in a document.

1.3. Individual Term Weighting

At this stage, the five features in section 3.1.2 are combined into one calculation that is accommodated in the \( S(w) \) variable. The smaller the \( S(w) \) value, the more important and relevant the candidate word is. The calculation of \( S(w) \) can be seen through Equation (6).

\[
S(w) = \frac{W_{\text{Rel}} + W_{\text{Position}}}{W_{\text{Case}} + \frac{W_{\text{Freq}}}{W_{\text{Rel}}} + \frac{W_{\text{DiffSentence}}}{W_{\text{Rel}}}}
\]

(6)

1.4. Candidate Term Generation

As we know, a candidate term may consist of more than one word (1-gram, 2-gram, or n-gram). Therefore, at the last stage, the final term candidate score will be calculated which consists of 1-gram, 2-gram, and n-gram through Equation (7).

\[
S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) \times (1 + \sum_{w \in kw} S(w))}
\]

(7)

where \( S(kw) \) is the final score of the candidate term kw. Where you are a term candidate who may consist of more than one word. The smaller the \( S(kw) \) value the more relevant the kw term candidate is to the existing documents.

2. Language Processing

In this study, language processing techniques used the combination of term extraction. The combination is expected to increase the term extraction yield. There are two language processing techniques used in this study, including:

- **Stemming.** This stages eliminate the affix that is in a word to get the basic form of the word. This study utilizes an algorithm developed by Nazief and Adriani (1996) as a rule for cutting affixes. According to Agusta (2009), this algorithm provides better stemming results for Indonesian words than other algorithms.

- **Stopword removal.** This stage eliminates the words in Bahasa Indonesia that have no meaning. For example, the prepositions “di”, “yang”, “di”, “ke”, and so forth. This stage is performed by matching every word in the sentence with the stopwords dictionary. If the word tested matches the word in the stopwords dictionary, the word is deleted from the sentence.
Methodology

In this section, the stages that are passed in the research are explained, including the preparation of test corpus, testing, and evaluation. Details of these stages will be explained in Sections 4.1-4.3. Illustration of the stages passed in the study can be seen in Fig. 1.

Corpus Development

The unstructured text that used as the material and source of this research data comes from articles on web pages contained on Indonesian tourism-themed websites, and uses Indonesian language media, namely: www.visitingjogja.com. Typo correction is needed given the YAKE term extraction algorithm is sensitive to character writing errors. The selection of tourism themes is expected to have a positive impact on the Indonesian tourism sector and make this research sustainable. Articles in the "Cultural and Historical Tourism" category were randomly taken as test corpus data. Each article is stored in a plain text file.

Combination Scenario

As stated earlier, this research improves term extraction process using combination of the YAKE term extraction algorithm and language processing techniques such as stopword removal and stemming. In this study, several modifications were made to the stemming algorithm, i.e., first, not making changes to the case character; secondly, only delete characters other than letters, numbers, comma characters, periods, and dash. The scenario of using YAKE term extraction algorithm and language processing in research is as follows:

1. Term extraction using YAKE
2. Stemming+YAKE
3. Stopword removal+YAKE
4. Stemming+stopword removal+YAKE
5. Stopword removal+stemming+YAKE

Keyword Evaluation and Ontology Formation

Each of the five scenarios mentioned above are tested on the corpus text that has been prepared. The test results are evaluated to see each performance and accuracy. The evaluation is carried out by the experts who understand the problem domain, in this case, experts who understand the tourism domain. Each scenario produces 200 extracted keywords. The evaluation is given to the term extracted by giving a value of 1 for items that have the correct, and a value of 0 for the wrong item. The scenario that has the best results is then transformed in the form of an ontology.
The final stage of the term extraction is to transform the correct term items extracted into a machine-readable format, i.e., the ontology format. The terms that come from the scenario having the highest percentage of truth are transformed in the form of ontology. The transformation algorithm explained in Fig. 2.

**Result and Discussion**

The results of the evaluation of the five scenarios proposed in this study can be seen in Table 1. From the five scenarios tested, it appears that the combination of the stopword removal+YAKE scenario has the highest percentage than the other scenarios. This is possible because Stopword Removal discards words that are not too significant in the problem context. On the other hand, various combinations of stemming produce terms with correctness percentage is lower than the combination of stopword removal+YAKE, even compared to original YAKE. This happens because of the characteristic of stemming itself, which makes a word back to its basic words. Bahasa Indonesia in particular, the use of stemming return the terms with incorrect results compared to when stemming is not used. For example the term “bangunan utama candi” (the main building of the temple), the stemming changes the term to “bangun utama candi” (the main wake temple). The stemming term as shown in the previous example cause the changes in meaning, this change makes the resulting term false. However, combination of stemming+stopword removal+YAKE produces a lower percentage than the original YAKE.

The final result of the term extraction and evaluation scenario is a knowledge base in the form of ontology. The combination of the stopword removal+YAKE scenario has the highest percentage compared to the other scenarios, therefore the correct terms originating from that scenario are transformed in the form of ontology. The formation of the ontology uses an algorithm and format as shown in Fig. 2. Meanwhile, Fig. 3 shows the results of the transformation.

**Table 1.** The results of testing and evaluation of the five term extraction scenarios.

| No. | Scenario Detail          | Item Correct | Percentage | Sum of Number | Similarity Average |
|-----|--------------------------|--------------|------------|---------------|--------------------|
| 1   | YAKE                     | 81           | 40.5%      | 1.451670123   | 0.017921853        |
| 2   | Stopword removal+YAKE    | 85           | 42.5%      | 1.571880667   | 0.018492714        |
| 3   | Stemming+YAKE            | 81           | 40.5%      | 1.497001185   | 0.018481496        |
| 4   | Stemming+stopword removal+YAKE | 76   | 38.0%      | 1.417468260   | 0.018650898        |
| 5   | Stopword removal+stemming+YAKE | 83       | 41.5%      | 1.602854725   | 0.019311503        |

*base_url = 'http://semanticweb.org/rajiva/tourism-domain'*

data_property = 'Keyword'

```
str = Create_RDF_Header(base_url)
str+= Create_Data_Property(data_property)

fp = read_file('keyword-list.csv')

while (line = fp.readline()):
    str+= Create_Individual(line.item(), data_property)

write_file('Ontology.owl', str)
```

**Figure 2.** Ontology transformation algorithm.
Conclusion

This research proves that the combination scenario of unsupervised keyword extraction techniques with language processing can increase the percentage of the correctness of the extracted results. The combination of stopword removal + YAKE produces the highest correctness value. However, not all proposed combination scenarios produce better result, the stemming + stopword removal + YAKE combination scenario is not even better than the original YAKE. Ontology learning through the methods and scenarios proposed in this study can accelerate time ontology development. However, ontology learning can be even faster if the keyword evaluation stage uses an automatic evaluation method. In future studies are expected to Evaluate ontology automatically or semi-automatically expected to speed up the learning process ontology.

Acknowledgments

This research was funded by the PNBP Institut Teknologi Sumatera (ITERA) through the HIBAH MANDIRI (Independent Research Grant Program) with contract No. B/318/IT9.C1/PT.01.03/2019.

References

[1] Azhari, Subanar, Wardoyo R and Hartati S 2008 Model Representasi Informasi Dan Pengetahuan Untuk Proyek-Proyek Perusahaan Dengan J. Ilm. Teknol. Inf. 7 85–92
[2] Sun J and Wang L 2014 Research on E-commerce Data Management Based on Semantic Web 14th International Conference on High Performance Computing and Communications p 925
[3] Kunder M de 2019 The size of the World Wide Web (The Internet)
[4] Imam I, Hamouda A and Khalek H A A 2013 An Ontology-based Summarization System for Arabic Documents (OSSAD) Int. J. Comput. Appl. 74 38–43
[5] Campos R, Mangaravite V and Pasquali A 2018 A Text Feature Based Automatic Keyword Extraction Method for Single Documents European Conference on Information Retrieval vol 2 pp 684–91
[6] Beliga S 2014 Keyword extraction: a review of methods and approaches Univ. Rijeka, Dep. Informatics, Rijeka 1–9
[7] Asim M N, Wasim M, Khan M U G, Mahmood W and Abbasi H M 2018 A survey of ontology learning techniques and applications Database 2018 1–24
[8] Hippisley A, Cheng D and Ahmad K 2005 The head-modifier principle and multilingual term extraction Nat. Lang. Eng. 11 129–57
[9] Belal M A E-F and Abdel-Galil H 2015 Ontology Extraction from Text: Related works between Arabic and English languages Int. J. Adv. Res. Comput. Sci. Manag. Stud. 4 120–30
[10] Fraga A L, Vegetti M and Leone H 2017 Semi-Automated Ontology Generation Process from Industrial Product Data Standards 46th Jornadas Argentinas Informática 53–66
[11] Guo R, Qiu J and Zhang G 2015 Web-Based Chinese Term Extraction in the Field of Study 2015 11th International Conference on Semantics, Knowledge and Grids (SKG) (IEEE) pp 133–9
[12] E.G.Drymonas 2009 Ontology learning from text based on multi-word term concepts: the ontogain method (M.Sc. Thesis Technical University of Crete, Greece)
[13] Yang Z, Zhou Y and Nyberg E 2016 Learning to Answer Biomedical Questions: OAQA at BioASQ 4B Proceedings of the Fourth BioASQ workshop (Stroudsburg, PA, USA: Association for Computational Linguistics) pp 23–37
[14] Chandu K, Naik A, Chandrasekar A, Yang Z, Gupta N and Nyberg E 2017 Tackling Biomedical Text Summarization: OAQA at BioASQ 5B BioNLP 2017 (Stroudsburg, PA, USA: Association for Computational Linguistics) pp 58–66
[15] Nazief B and Adriani M 1996 Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia Intern. Publ. Fac. Comput. Sci. Univ. Indones. Depok, Jakarta
[16] Agusta L 2009 Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia Konf. Nas. Sist. dan Inform. 2009 196–201