Entity Annotation WordPress Plugin using TAGME Technology

William Aprilius[1], Seng Hansun[2], Dennis Gunawan[3]
[1,2,3] Universitas Multimedia Nusantara, Jl. Boulevard Gading Serpong, Scientia Garden, Tangerang
*Corresponding author, e-mail: william.aprilius@yahoo.com[1], hansun@umn.ac.id[2],
dennis.gunawan@umn.ac.id[3]

Abstract

The development of internet technology makes more information can be accessed. It makes information need to be organized in order to be easily managed. One solution can be used is by using the entity annotation approach which generates tags to represent that document. In this study, TAGME technology is implemented on a WordPress plugin, which is used to manage a blog. Moreover, information on Wikipedia 'Bahasa Indonesia' is processed to generate an anchor dictionary which is required by the technology that is implemented. This plugin performs entity annotation by giving tag suggestion for posts in a blog. Testing is carried out by measuring the precision, recall, and F_1 of tag suggestions given by the plugin. The result shows that the plugin can give tag suggestions with precision 0.7638, recall 0.5508, and F_1 0.59.

***Keywords**: entity annotation, TAGME, wikipedia, wordpress*

1. Introduction

The development of internet technology makes more information can be accessed. The most important part of the data remains unstructured documents [1]. This makes internet users need to manage large amounts of information [2]. Therefore, a method to organize that information is needed in order to be easily managed.

One of the solutions used to overcome the problem of organizing the information in the document is indexing or "tagging". Here, each document in a collection is indexed with a set of key phrases ("tags") that reflect its principal topics. However, assigning tag or key phrase manually is so time-consuming and impractical [3]. Therefore, a technique that can generate tags for a document automatically is required.

Key phrase indexing is an approach that maps a word or phrase in the documents to a related term in a controlled vocabulary. It is an intermediate approach between key phrase extraction and term assignment that combines the advantages of both and avoids their shortcomings [4]. Wikipedia can be used effectively to build a controlled vocabulary, which is then used to perform key phrase indexing. This approach is also called as topic indexing [5]. Thus, the resulting tag or key phrase is a term in a controlled vocabulary built from Wikipedia.

Mihalcea & Csomai [6] define "text wikification" (shortly "wikification") as the task of automatically extracting the most important words and phrases in the document, and identifying for each such words and phrases the appropriate link to a Wikipedia article. Milne and Witten [7] also had done a research to resolve "wikification" by using machine learning algorithm. Thus, the problem of topic indexing is closely related to "wikification" when Wikipedia is used to build a controlled vocabulary.

TAGME [8, 9] is the first software system that, on-the-fly and with high precision or recall, perform entity annotation [10] for short text. An example of a short text is the post of a blog. Ferragina and Scaiella [8] also explains that there are new challenges in performing entity annotation on short text, which (1) could occur on-the-fly and thus cannot be pre-processed and (2) should be designed properly, because the input texts are too short that it is difficult to mine significant statistics that are rather available when texts are long. Thus, organizing the posts of a blog can be done with entity annotation approach, which uses tags to represent that blog posts. Challenges in entity annotation process can be solved with TAGME's algorithmic technology (shortly TAGME technology). Therefore, in this study, technology (algorithm), which

Received August 5, 2016; Revised October 12, 2016; Accepted November 18, 2016

is used in TAGME, is implemented to perform entity annotation on a post in WordPress CMS. WordPress is the most used CMS (47.09%) over the world [11]. Furthermore, it's also the most used CMS for blog management (97.78%) [11]. Linawati et al. [12] even proposed synchronization interfaces to migrate teachers' or lecturers' learning materials from WordPress into Moodle, so that it can improve the Moodle utilization as an e-learning system.

2. TAGME Technology

TAGME is a “topic annotator” that is able to identify meaningful sequences of words in a short text and link them to a pertinent Wikipedia [13]. TAGME uses the sequences of terms composing the anchor texts which occur in the Wikipedia pages to identify meaningful term sequences (spot) in the input text, then uses the pages (possibly many) pointed in Wikipedia by that spot or anchor as possible senses for each spot. Sense selection for the spot which has more than one sense is done by using functions, which fast to be computed and accurate to find a collective agreement among all spot to Wikipedia page (sense) mapping [8].

There are three steps, which is done by TAGME to perform annotation. They are as described by the following subsection [8].

2.1. Anchor Parsing

TAGME receives a short text as input, tokenizes it, and then detects the anchors by querying the anchor dictionary. If there are two anchors (a_1 and a_2) and a_1 is a (word-based) substring of a_2, TAGME drop a_1 only if $lp(a_1) < lp(a_2)$.

2.2. Anchor Disambiguation

In this step, TAGME tries to disambiguate each anchor, i.e. choose one sense for each anchor. This is done by calculating a value for each anchor's possible sense which is wanted to be disambiguated. This value is obtained by using a voting scheme, which calculates the vote from each other anchors to annotation for that anchor.

Ferragina and Scaiella [8] explains that the basic calculation of vote as described by Equation (1), which is proposed by Milne and Witten [14] to measure relatedness between two Wikipedia pages (a and b).

$$rel(a,b) = 1 - \frac{log\left(max(|in(a),in(b)|)\right) - log(|in(a) \cap in(b)|)}{log(W) - log\left(min(|in(a),in(b)|)\right)} \quad (1)$$

In (1), W is the number of page in Wikipedia, $in(a)$ and $in(b)$ are the set of Wikipedia pages pointing to page a and b.

After obtaining total vote for each anchor, TAGME uses an approach which is called Disambiguation by Threshold (shortly DT), i.e. get 30% sense which has highest total vote, then annotates anchor with sense which has the highest commonness.

2.3. Anchor Pruning

The disambiguation phase produces a set of candidate annotations, one per anchor detected in the input text. This set has to be pruned in order to possibly discard the meaningless annotations. These "bad annotations" are detected by using a simple scoring function that takes into account only two features: the value of link probability and coherence. Pruning score is obtained by calculating the average of these two values. If the obtained pruning score for an annotation is smaller than a threshold, ρNA, that annotation is discarded from the final resulting set of annotation.

3. Entity Annotation

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that make the reader understand easily [2], [5]. The discussion can be made in several sub-chapters.

Entity annotation is an approach to overcome the limitation of classic approaches which are based on "bag-of-words" paradigm in providing semantic representation for a text document. The key idea of this approach is to identify, in the input text, short-and-meaningful sequences of terms (also called mentions) and annotate them with unambiguous identifiers (also called entities) drawn from a catalog, such as Wikipedia [10].

The process of entity annotation involves three main steps [10]:

1. parsing of the input text, which is the task to detect candidate entity mentions and link each of them to all possible entities they could mention;
2. disambiguation of mentions, which is the task of selecting the most pertinent Wikipedia page (i.e. entity) that best describes each mention;
3. Pruning of a mention, which discards a detected mention and its annotated entity if they are considered not interesting or pertinent to the semantic interpretation of the input text.

The problem of entity annotation can be casted in two main classes: (1) the identification of (possibly scored) annotations, and thus the identification of mention-entity pairs; and (2) finding tags (possibly scored or ranked), and thus accounting only for the entities [10].

4. Methodology and Design

In this study, two applications were developed, i.e. preprocessing application and plugin application. Preprocessing application is used to process information in Wikipedia (i.e. Wikipedia articles which are exported by using 'Special: Export' facility) to create an anchor dictionary which is required to implement TAGME technology. Plugin application is a plugin for WordPress CMS which is used to perform entity annotation. The entity annotation process is performed by using anchor dictionary, which is created by preprocessing application. The relationship between these two applications is shown in Figure 1.

In this study, Wikipedia Bahasa Indonesia articles under Computer Science (Ilmu Komputer in Bahasa Indonesia) category structure were downloaded. We retrieved only articles that are in the range of two level subcategories from the root category. These Wikipedia articles were downloaded on 22 February 2015. This also explains the steps being taken to get the Wikipedia Bahasa Indonesia articles in Computer Science category which is shown in Figure 1.

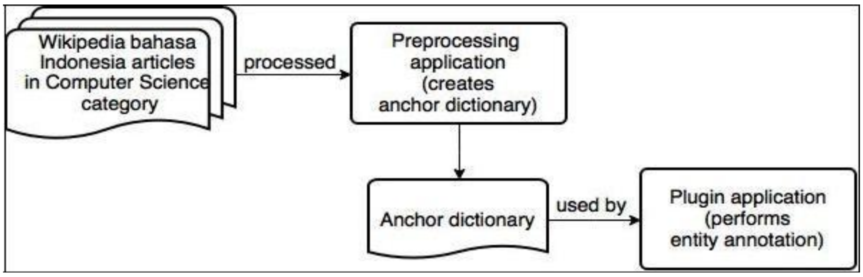

Wikipedia bahasa Indonesia articles in Computer Science category
processed
Preprocessing application (creates anchor dictionary)
Anchor dictionary
used by
Plugin application (performs entity annotation)

Figure 1. Relationship between Preprocessing Application and Plugin Application

4.1. Preprocessing Application

Preprocessing application is an application which is used to process information in Wikipedia. This information is in the form of XML file of the exported Wikipedia pages (henceforth referred to as Wikipedia XML dump). This application creates an anchor dictionary in the form of XML file.

Figure 2 shows a flowchart of the main process in preprocessing application. Parsing and iteration process of Wikipedia XML dump is done by using WikiXMLJ. In Figure 2, in the process of getting the set of mapping data, a mapping data is a wikilink or internal link. A ‘wikilink’ links a page to another page within Wikipedia. This process aims to get the set of mapping data in the Wikipedia page which is being processed in the iteration. This also means that mapping data consists of an anchor text (i.e. a text that appears in Wikipedia pages as a link) and a sense (i.e. Wikipedia page which is targeted by that link). In addition, this process ensures the anchor text in the mapping data is composed by more than one character and not just numbers.

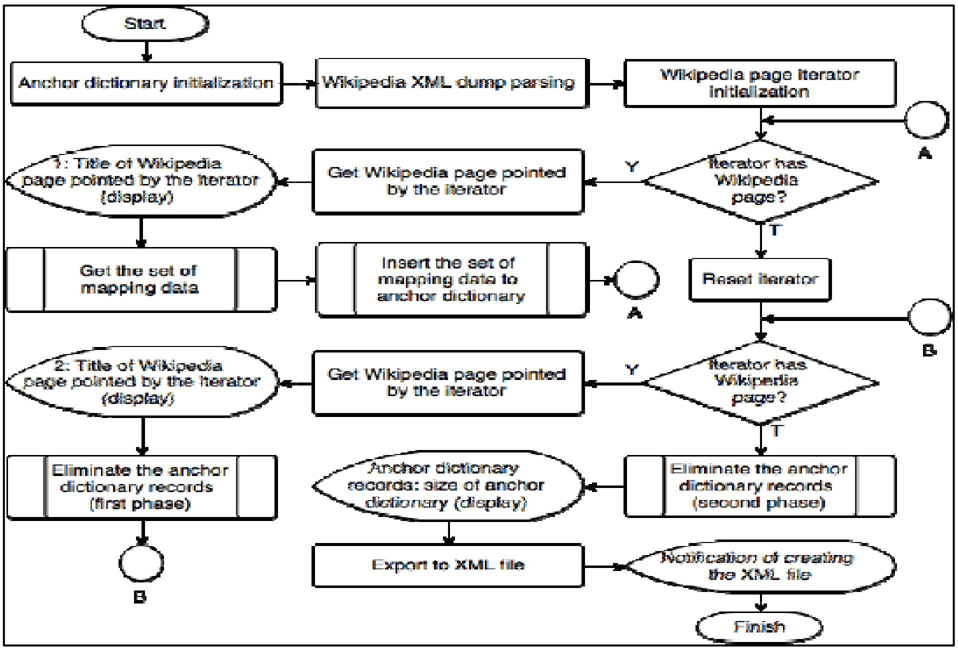
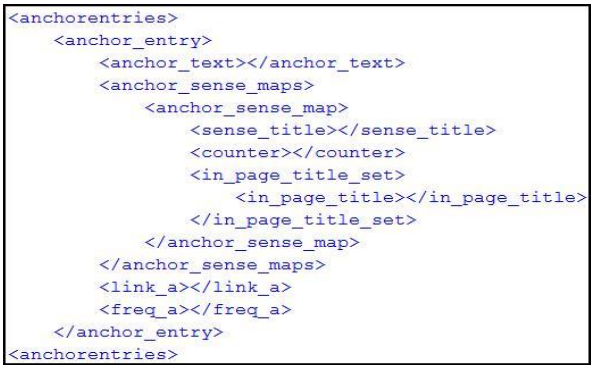

Start
Anchor dictionary initialization
Wikipedia XML dump parsing
Wikipedia page iterator initialization
A
1: Title of Wikipedia page pointed by the iterator (display)
Get Wikipedia page pointed by the iterator
Y
Iterator has Wikipedia page?
T
Get the set of mapping data
Insert the set of mapping data to anchor dictionary
A
Reset iterator
B
2: Title of Wikipedia page pointed by the iterator (display)
Get Wikipedia page pointed by the iterator
Y
Iterator has Wikipedia page?
T
Eliminate the anchor dictionary records (first phase)
Anchor dictionary records: size of anchor dictionary (display)
Eliminate the anchor dictionary records (second phase)
B
Export to XML file
Notification of creating the XML file
Finish

Figure 2. Flowchart of the Main Process in Preprocessing Application

The next process is to insert the set of mapping data which is found to anchor dictionary. Moreover, $link(a)$, the number of mapping data (counter), and "in page" set which corresponds to a mapping data is also specified to anchor dictionary. The "in page" set is a set of Wikipedia page title which the corresponding page contains that mapping data.

In Figure 2, the process of elimination of the anchor dictionary records (first phase) aims to discard anchor dictionary records with $link(a) < 2$. Moreover, this process counts the frequency of anchor text occurrence in Wikipedia page which is being processed in the iteration, then accumulates that frequency value to $freq(a)$ in anchor dictionary record which corresponds to that anchor text. The process of elimination of the anchor dictionary records (second phase) aims to discard anchor dictionary records with $lp(a) < 0.1\%$. Figure 3 shows the XML element structure of an anchored dictionary record.



```
<anchorentries>
    <anchor_entry>
        <anchor_text></anchor_text>
        <anchor_sense_maps>
            <anchor_sense_map>
                <sense_title></sense_title>
                <counter></counter>
                <in_page_title_set>
                    <in_page_title></in_page_title>
                </in_page_title_set>
            </anchor_sense_map>
        </anchor_sense_maps>
        <link_a></link_a>
        <freq_a></freq_a>
    </anchor_entry>
<anchorentries>
```

Figure 3. XML Element Structure of an Anchor Dictionary Record

4.2. Plugin Application

Plugin application performs entity annotation by using anchor dictionary which is created by preprocessing application. The entity annotation process is performed by giving tag suggestion(s) to user based on the written input text and creating hyperlink on word or phrase in the input text to Wikipedia page based on the selected tag suggestion(s) which are wanted to be the tags of the post. This plugin is named Wiki CS Annotation.

The plugin's user interface is displayed in the page of editing a post or creating a new post, under the text editor in WordPress CMS. This plugin allows user to request tag suggestion(s), choose tag suggestion(s) given to become tag(s) of a post, delete the chosen tag(s), and change the ρNA threshold value.

TAGME technology which had been implemented on this plugin works when user requests tag suggestion. Figure 4 shows the process when user request tag suggestion. Moreover, Figure 4 shows that there are three main processes which are performed to handle tag suggestion request. These processes are also denoted as the main step in TAGME, i.e. anchor parsing, anchor disambiguation, and anchor pruning.

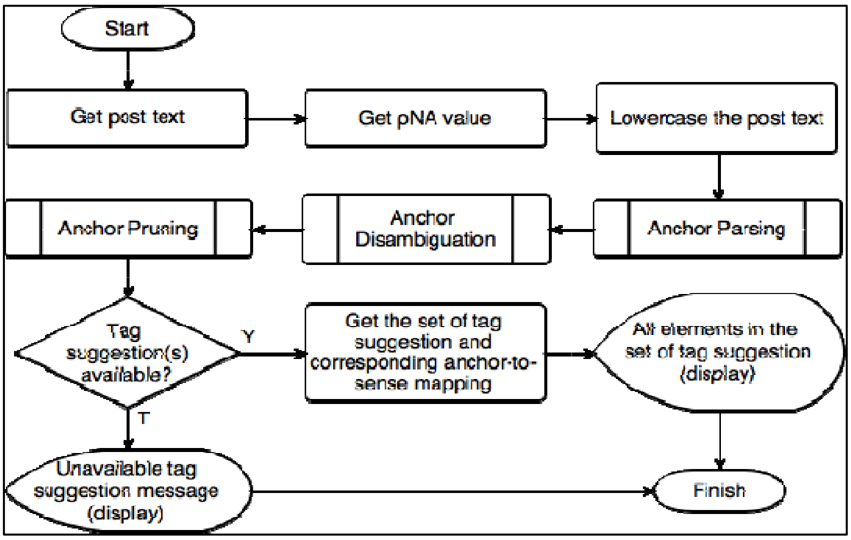

Start
Get post text
Get pNA value
Lowercase the post text
Anchor Parsing
Anchor Disambiguation
Anchor Pruning
Tag suggestion(s) available?
Y
T
Get the set of tag suggestion and corresponding anchor-to-sense mapping
All elements in the set of tag suggestion (display)
Unavailable tag suggestion message (display)
Finish

Figure 4. Flowchart of the Main Process when User Requests Tag Suggestion

Anchor parsing process queries the anchor dictionary to obtain anchor text records and searches the occurrence of those anchor text records in the post. Moreover, this process also perform selection process if more than one anchor text is found, which one of those anchor text is substring (word-based) of other anchors text, follows what is done by TAGME.

The next process is anchor disambiguation. If an anchor text has more than one sense, disambiguation is a process of choosing one of those senses. Thus, this process generates a set of anchor text that is mapped to exactly one sense.

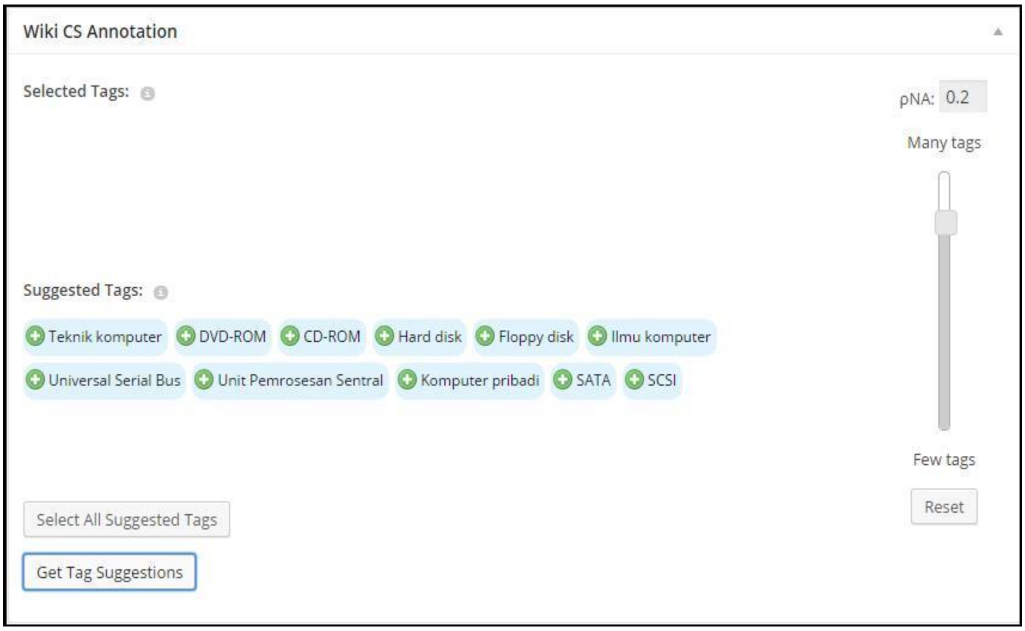
The next process is anchor pruning which aims to discard meaningless annotations. This is done by calculating a pruning score, then by comparing it with a threshold ρNA as done by TAGME. This process results a set of final annotations for a post.

5. Implementation

The plugin application which was developed is implemented on WordPress 4.1.5. This plugin can be used after the user installs the plugin by using Add Plugins page on WordPress. Then, user needs to activate the plugin which can be done through Plugins page on WordPress. The installation can be done by user who has privilege as an administrator.

Anchor dictionary which is created by preprocessing application contains 1,259 records. This anchor dictionary needs to be imported to WordPress, so it can be used by the plugin. This importing process can be done by using anchor dictionary importing facility which is provided by the plugin.

The request of tag suggestion can be done from the page of editing a post or creating a new post. Figure 5 shows tag suggestions which given by the plugin with input text about computer hardware.



pNA: 0.2

Figure 5. Tag Suggestion Given by the Plugin for Input Text about Computer Hardware

A tag suggestion which is chosen by the user will become a post's tag. When the user saves (as a draft) or publishes a post, before that process is executed by the WordPress, plugin will create hyperlink on word or phrase in the input text to Wikipedia page based on the selected tag suggestions. For example, in Figure 5, the "Perangkat keras" tag ("perangkat keras" means "hardware" in English) is chosen, then user decides to save or publish that post (an input text about computer hardware). Before WordPress saves or publishes that post, plugin will create hyperlink on all "perangkat keras" phrase in the input text to Wikipedia page titled “Perangkat keras”. This is because "perangkat keras" phrase is the anchor text of "Perangkat keras" sense.

6. Testing and Results

Testing is performed to measure precision, recall, and F_1 of the tag suggestion(s) given by the plugin.

6.1. Testing Data

Data set for testing is built by selecting randomly 100 Wikipedia articles under Computer Science category structure. Then, texts in those articles are edited by discarding citation (if available), i.e. number between square brackets. The result of this process is 100 plain texts which becomes the data set for testing.

In this study, we limit the text's length in the testing data set to around 200 words. This is done as follows: if the length of a Wikipedia article is less than 200 words, we take all text in that article as the testing data. Otherwise, we take only the first (around) 200 words with sentence as the window of selection.

6.2. Results

In this study, precision and recall are measured with focus on which sense got linked, i.e. tag suggestion which is given by the plugin. Let $G(T)$ is the set of sense in text T in ground truth, i.e. Wikipedia articles which are used to create text T and $P(T)$ is the set of sense in text

T which is given by the plugin in the form of tag suggestion set. Then, $G(T)$ is added with the title of Wikipedia article which is used to create text, T. This addition is performed because that title of Wikipedia article can become a tag of the text T, but not contained in the set of sense in the text T in ground truth. Precision and recall is calculated based on $G(T)$ and $P(T)$. Moreover, let $p \epsilon P(T)$ and $g \epsilon G(T)$, there is probability that $p \neq g$, but p is still relevant to g (p is considered equal to g). This can be occured if in the Wikipedia, p is a redirect of g, or vice versa.

For each text in testing data set, precision and recall is calculated with five variations of ρNA values, i.e. 0, 0.1, 0.2, 0.3, and 0.4. Table 1 shows the average of precision and recall from 100 testing data, and F_1 for that average of precision and recall.

Table 1. Precision, Recall, and F1 for each Variation of ρNA Values

ρNA	Precision	Recall	F_1
0	0.5185	0.5508	0.5342
0.1	0.6674	0.5287	0.5900
0.2	0.7638	0.3923	0.5184
0.3	0.7043	0.2430	0.3613
0.4	0.5968	0.1339	0.2187

Table 1 shows that the highest average of precision (i.e. 0.7638), is obtained when $\rho NA = 0.2$ and the highest average of recall (i.e. 0.5508) is obtained when $\rho NA = 0$. Moreover, the highest F_1 (i.e. 0.59) is obtained when $\rho NA = 0.1$.

Based on the testing process, it was found that Wikipedia Bahasa Indonesia is not good enough if it is used as ground truth for testing or evaluation, particularly the articles which are under the Computer Science category structure. This is because we still found sense that is not appropriate for a particular anchor text. For example, sense of anchor text "eksekusi" (means "execute" in English) in Ada (programming language) is “Hukuman mati” (means "death penalty" in English), which is not proper. Anchor text "eksekusi" in that article should be interpreted as execution of program code. Thus, it is necessary to use a better set of testing data, i.e. each anchor text in that data set has an appropriate sense.

7. Conclusion

Information in Wikipedia (i.e. the exported Wikipedia articles, particularly which are under Computer Science category structure) is processed to create an anchor dictionary in the form of XML file. This anchor dictionary contains 1,259 records. Moreover, TAGME technology is implemented as a WordPress plugin to perform entity annotation. The entity annotation is performed by giving tag suggestion for a post and creating hyperlink to Wikipedia page on word or phrase in the post based on the selected tag suggestion which is wanted to be the tag of the post.

Based on the testing result, the highest precision, recall, and F_1, that can be achieved by the plugin is 0.7638, 0.5508, and 0.59 respectively. In a subsequent study, testing should be carried out by using a better data set. Moreover, to increase precision, recall, and F_1, a study can be conducted to find a novel or modified relatedness function which can give better result when use a smaller anchor dictionary (subset of Wikipedia), as done in this study.

In the future, we intent to develop the plugin to perform text categorization, which uses the entity annotation approach. In our hypothesis, this can be done by including the Wikipedia page's category structure and information to anchor dictionary which is conducted by the preprocessing application. Moreover, we also need to modify the plugin so that it can facilitate the text categorization functionality.

References

[1] Abdoulahi Boubacar, Zhendong Niu. Conceptual Search Based on Semantic Relatedness. *Indonesian Journal of Electrical Engineering and Computer Science*. 2014; 12(8): 6380-6385.

[2] Dan Roth, Heng Ji, Ming-Wei Chang, Taylor Cassidy. *Wikification and Beyond: The Challenges of Entity and Concept Grounding*. Proc. 52nd Annual Meeting of the Association for Computational Linguistics: Tutorials (ACL2014), USA. 2014: 7.

[3] Olena Medelyan, Ian H Witten. Domain Independent Automatic Keyphrase Indexing with Small Training Sets. *Journal of the American Society for Information Science and Technology*. 2008; 59(7): 1026-1040.

[4] Olena Medelyan, Ian H. Witten. *Thesaurus Based Automatic Keyphrase Indexing*. Proc. Joint Conference on Digital Libraries (JCDL), USA. 2006.

[5] Olena Medelyan, Ian H. Witten, David Milne. *Topic Indexing with Wikipedia*. Proc. Wikipedia and AI Workshop at the AAAI-2008 Conference, Chicago. 2008: 19-24.

[6] Rada Mihalcea, Andras Csomai. *Wikify! Linking Documents to Encyclopedic Knowledge*. Proc. 16th ACM Conference on Information and Knowledge Management (CIKM ’07), New York. 2007: 233-242.

[7] David Milne, Ian H. Witten. *Learning to Link with Wikipedia*. Proc. Conference on Information and Knowledge Management (CIKM '08), New York. 2008: 509-518.

[8] Paolo Ferragina, Ugo Scaiella. Fast and Accurate Annotation of Short Texts with Wikipedia Pages, *IEEE Software*. 2011; 29(1): 70-75.

[9] Paolo Ferragina, Ugo Scaiella. *TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities)*. Proc. Conference on Information and Knowledge Management (CIKM '10), Canada, October 2010.

[10] Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita. *A Framework for Benchmarking Entity-Annotation System*. Proc. 22nd International Conference on World Wide Web (WWW '13), Switzerland. 2013: 249-260.

[11] BuiltWith, CMS Usage Statistics: Statistics for Websites using CMS Technologies, [online]. Available on http://trends.builtwith.com/cms. accessed on 6 June 2015.

[12] Linawati, Gede Sukadarmika, GM Arya Sasmita. Synchronization Interfaces for Improving Moodle Utilization. *TELKOMNIKA*. 2012; 10(1): 179-188.

[13] Paolo Ferragina. TAGME Technology. Available on http://acube.di.unipi.it/tagme/. accessed on 23 February 2015.

[14] David Milne, Ian H Witten. *An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links*. Proc. AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, Chicago. 2008: 25-30.