Web-based E-Commerce Products Grouping

M Elveny1*, S M Hardi1, I Jaya1 and P Gundari1

1 Faculty of Computer Science and Information Technology, Universitas Sumatera Utara

*marischaelveny@usu.ac.id

Abstract. E-Commerce is a group of technologies, applications, and businesses that connect companies or individuals as consumers to conduct electronic transactions, exchange of goods or information through the internet or television, www, or other computer networks. The rapid growth of E-Commerce is caused by the many conveniences for buyers and sellers. This can be seen by the increasing number of E-Commerce sites such as Bukalapak.com, Tokopedia.com, and so on. The buyers want to get products with the lowest prices, the best quality, and from trusted sellers. This encourages them to search for and compare the same products from one store to another, even between different sites. Variables that they compare vary. Some are based on price, number of reviews, and so on. But it is troublesome when they have to look for the same product from another site. This study used the product name as a reference in grouping the same product from 5 different sites. The algorithm used to group online shop products was Jaro-Winkler Distance. The study was to create a system that can group products from 5 different sites with the highest level of similarity product as the output. From the 20 times of the search tests, the accuracy of the online shop product grouping with Jaro-Winkler Distance was 81.27%.

1. Introduction

E-Commerce is an activity of buying and selling products or services electronically. E-Commerce makes it easy for sellers to market their products without having to have a physical store. E-Commerce also makes it easy for buyers to find a product because they don't have to go to each store manually to find the item they want. In addition, E-Commerce also provides flexibility for consumers to choose products.

This current rapid growth of E-Commerce is due to some of these conveniences that makes E-Commerce becomes a fertile business land. This can be seen by the emergence of E-Commerce sites such as Bukalapak.com, Tokopedia.com, and so on. These sites display additional information on the products they sell such as the number of people who buy the product, the number of reviews, and so on. These sites generally display the same product but with different names due to different sentences, wording, typos, and so on. This troubles the buyer when looking for the same product from another site.

The buyers want to get products with the lowest prices, the best quality, and from trusted sellers. This encourages them to search for and compare the same products from one store to another, even between different sites. Variables that they compare vary. Some are based on price, number of reviews, and so on. Searching and comparing the same product on the same site may not be too complex. However, what if the buyer tries to compare the same product from 5 different sites?

Previously, research had been carried out about the implementation of the Levenshtein Distance algorithm and empirical methods in building a spelling checking system. The empirical method was used to determine the existence of words written without spaces so that the suggestions given can be
more accurate [1]. There were 3 stages in Levenshtein Distance, namely: Insertion, Deletion, and Substitution. Each character can only go through one stage [2].

In this study, we proposed the Jaro-Winkler Distance method. Jaro-Winkler Distance is an accurate approximate string matching algorithm that has never been used in the case of matching item names. Jaro-Winkler Distance is a metric on a string to calculate edit distance between 2 strings. Distance between 2 strings is number transposition of a character to change from one word to a new word. Jaro-Winkler Distance uses prefix scale to increase distance accuracy if there is the same character at the beginning of the string [3].

2. Method
The method proposed in this study consists of several stages. All stages can be seen in Figure 1.

![General Architecture](image)

Figure 1. General Architecture.

The first stage starts from the search input by the user to the system, then the input will be sent to the Tokopedia site to obtain a web page that contains products that match the inputted search. After that the web page will be scraped to keep the source code of the page temporarily.

Web scraping is the process of gathering information from certain sites with human as the end user [4]. When doing web scraping on a website, the first thing we have to understand is the policy of the person/group that owns the website. Between one website and another, there are different policies for using data from their site. When doing web scraping, it's better to give a time lag of more than 1 second between requests. This is done so that server performance is not excessive and the users who are actually
aiming to access the website are not disturbed because of the slow access. One technique for web scraping is HTML Parsing.

HTML parsing is a syntax analysis in the HTML. Analysis of the HTML syntax series will produce parsing trees that show their syntactic relationships with each other and may also contain semantic information. Beautiful Soup is one of the libraries in the python programming environment that has HTML Parsing capabilities. Beautiful Soup was made by Leonard Richardson in 2004 and is still being developed today. Some examples of modules from Beautiful Soup are as follows:

1. Looking for the first element with the tag “div”, the attribute “class” and its value starts with “detail__name” then retrieving content from that element.

**HTML**

```
<div class="detail__name js-ellipsis ng-binding" data-js-ellipsislimit="42">Mifi Modem Wifi 4G XL Go Movimax MV003 Free 60Gb 60Hari...</div>
```

**Python**

```
x = find("div", {"class": "detail__name"}).string
```

Output “x” adalah “Mifi Modem Wifi 4G XL Go Movimax MV003 Free 60Gb 60Hari…”

2. Looking for all elements with the tag “div”, the attribute “class” and its value of “c-product-card c-product-list__item c-product-card_view_grid”

**HTML**

```
1. <div class="c-product-card c-product-list__item c-product-card_view_grid">a b c d</div>
2. <div class="c-product-card c-product-list__item">a b c</div>
```

**Python**

```
a. select("div.c-product-card.c-product-list__item")
b. find_all("div", {"class": "c-product-card c-product-card_view_grid"})
```

In example a, all elements with the value class starts with “c-product-card.c-product-list__item” will be searched so that the elements taken are elements 1 and 2. While in example b, the element to be searched for must be exactly the same value sought so if the specified element is not found then nothing is done so that the element taken is the 2nd element.

3. Taking the value of the element with the tag "img" and the attribute "src"

**HTML**

```
<img itemprop="image" src="https://ecs12.tokopedia.net/newimg/cache/200-square/product-1/2015/2/1/315373/315373_6a52b5cc-aa2b-11e4-9d6b-3e934908a8c2.jpg">
```

**Python**

```
x = produk.find("img").get('src')
```

The output “x” is “https://ecs12.tokopedia.net/newimg/cache/200-square/product-1/2015/2/1/315373/315373_6a52b5cc-aa2b-11e4-9d6b-3e934908a8c2.jpg”.

Then, HTML parsing will be done on the source code to detect product names, product prices, number of reviews, and the first to fourth product URL. After that, the price will be ranged with the selection of products that are 50 percent larger or smaller. After that, accuracy improvement is done, which is a step to increase the edit distance value of a product such as non-alphanumeric removal, stop words removal (deletion of words that do not really affect the semantics of a sentence), porter stemming (the process of finding the basic form of a word by removing the prefix or suffix word), and word position. In the Preprocessing section, the porter stemming stage precedes the stop words removal stage because if for example the word “terlaris” is found. The word is not in the dictionary stop words, but the word “laris” is in the stop words dictionary so that porter stemming is needed to delete stop words first. The Stop Words Removal step precedes the word position stage because if for example in the
comparison of the first string “Epson printer xyz” with the second string “printer epson xyz terlaris”. The two strings will not be detected as 2 strings that are exactly the same but differ in their word order because the word “terlaris” blocks that.

After that, an approximate product matching will be carried out, which is the stage to assess the similarity of a product based on the similarity of the product name using the Jaro Winkler Distance algorithm. The Jaro-Winkler Distance algorithm is an algorithm for calculating the similarity of two strings (edit distance) and is generally used in the problem of detecting data duplication [5]. The algorithm is the development of the Jaro algorithm Distance with the addition of a prefix scale to increase distance accuracy if there are 1 to 4 characters in common at the beginning of the string [3].

The Jaro-Winkler Distance algorithm:

\[
d_j = \begin{cases} 
0 & \text{if } m=0 \\
\frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{|m|} \right) & \text{else} 
\end{cases}
\]  

\( m \) = the same number of characters  
\( s_1 \) = first string length  
\( s_2 \) = second string length  
\( t \) = the number of transpositions is divided by 2

The characters between \( s_1 \) and \( s_2 \) are considered the same if in distance:

\[
\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1
\] (2)

Transposition is when the same character is found in the distance above but not in one column. In the Jaro-Winkler Distance algorithm, a bonus prefix is added to increase the accuracy value based on the same number of characters in the prefix of a string. Edit distance Jaro-Winkler Distance algorithm:

\[
d_w = \begin{cases} 
0 & \text{if } d_j < bt \\
d_j + (l \cdot p \cdot (1 - d_j)) & \text{else} 
\end{cases}
\] (3)

\( l \) = the same number of characters (maximum of 4)  
\( p \) = prefix scale (minimum of 0.1 and maximum of 0.25)  
\( bt \) (boost threshold) = the commencement of adding a prefix bonus

After these steps are carried out, the product with the highest similarity value will be chosen from other sites.

3. Results and Discussions

We carry out the test by conducting the search 20 times with the following conditions:
1. Minimal price : Rp. 1
2. Maximum price : Rp 100,000,000
3. Sites : Tokopedia, Lazada, Shopee, Bukalapak, and Blibli
4. Accuracy Improvement : Price Range and Preprocessing
5. Prefix Scale : 0,1
6. Boost Threshold : 0,7

The test results are shown in Figure 2 and Figure 3:
In Figure 2 and Figure 3, it is shown that there are 48 products proven to be the same and 8 products proven to be different. Based on the results of tests that have been carried out on the online shop product grouping system using the Jaro-Winkler Distance algorithm can be obtained the value of accuracy in grouping the same online shop product.

\[
\text{accuracy} = \frac{\text{the total of same products}}{\text{total product}} \times 100\%
\]

\[
\text{accuracy} = \frac{191}{235} \times 100\% = 81.27\%
\]

From the above calculations it can be seen that the accuracy of the Jaro-Winkler Distance method in grouping the same online shop product is 81.27%.

4. Conclusions and Suggestions
The conclusions based on the results of the online shop product grouping system using the Jaro-Winkler Distance algorithm is that the Jaro-Winkler Distance algorithm is able to group products well with an accuracy of 81.27%. Our suggestions for the development of this research in the future are to avoid
taking the products next to them as the same products on the Tokopedia website, to use methods that can minimize resources used when doing web scraping and Porter stemming, to perform image processing in images obtained from the Tokopedia website to ensure that the products taken are not the same, to compare the accuracy obtained from other approximate string matching algorithms and to add the dictionary of stop words both in Indonesian and English.

5. Acknowledgments
The authors gratefully acknowledge that the present research is supported by Fund Dissemination IPTEKS Research Results for Lecturers / Researchers Universitas Sumatera Utara.

References
[1] Adriyani N M M, Santiyasa I W, Muliantara A 2012 Implementasi Algoritma Levenshtein Distance dan Metode Empiris Untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia J. Elektronik Ilmu Komputer Universitas Udayana 1 1.
[2] Kaur S, Kiranjyoti 2015 Mining Text using Levenshtein Distance in Hierarchical Clustering Int. J. of Computer Techniques 2 1 pp 93–97.
[3] Grannis S J, Overhage J M and McDonald C 2004 Real World Performance of Approximate String Comparators for use in Patient Matching (Amsterdam: IOS Press)
[4] Watson M 2009 Scripting Intelligence Web 3.0 Information Gathering and Processing (New York: Apress)