

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

AUTOMATING PRICE MATCHING ON E-COMMERCE WEBSITES USING NATURAL LANGUAGE PROCESSING

A postgraduate project dissertation presented
in partial fulfilment of the requirements

for the degree of

Masters in Information Technology

at Massey University,
Auckland, New Zealand

Jiansheng Xie

2016

Abstract

With the development of internet, shopping online has become an important part in our daily life. Global B2C e-commerce turnover grew by 24.0% to reach 1,943 billion dollars in 2014. Not only customers need to face a great amount of information while shopping online, the companies also need to catch the information from their competitors. There is a case which a company wanted to realize was a simple way for them to monitor the prices of equivalent products on competitor's website. Base on the development of the E-commerce plat form, after analyzing the requirement of companies and customer, we propose a frame of E-commerce website data extraction, data storage and production matching. We build up a customized web crawler to crawl the production on E-commerce website and extract the production detail for matching. Finally we got average 87.18% matching rate after applying enhance TF/IDF algorithm with weight adjustment.

Contents

1	Introduction	1
1.1	Case Introduction	1
1.2	Research Problem	2
2	Literature Review	3
2.1	Review on E-commerce Website	3
2.2	Review on Website Crawler	3
2.2.1	Brief Review of Crawler	3
2.2.2	Category of Crawler	4
2.2.3	Strategy of Crawler	5
2.2.4	Recently Research about Focused Crawler	5
2.3	Review on Website Related Technology	6
2.3.1	HTML	6
2.3.2	XML	6
2.3.3	DOM	7
2.3.4	Regular Expressions	8
2.4	Web Extraction	8
2.4.1	Brief Review of Web Extraction	8
2.4.2	The Categories of Web Extraction	10
2.5	Review on production matching	11
2.5.1	Natural Language Processing	11
2.5.2	Tokenization	11
2.5.3	Syntactic Analysis	12
2.5.4	Product Search by Semantic Web Technology	13
2.5.5	Synthesized similarity method (SSM)	15
3	Methodology	17
3.1	Gather Data	17
3.1.1	The Structure of the Web Crawler	17
3.1.2	The Technique of the Web Crawler	17
3.1.3	Setting the rule for filtering URL and extract product detail. . . .	17
3.1.4	Information Storage	21
3.2	Production Matching	23
3.2.1	Clean the Data	23
3.2.2	Tokenization the document	23
3.2.3	Syntactic Analyses	26

3.2.4	Vector Space Model	26
3.2.5	Term Weight	27
3.2.6	Measurement	27
3.2.7	Calculating Vector Similarity	28
3.2.8	Find Similar Product	28
3.2.9	Adjusting Parameters	28
3.3	Price Comparison	30
3.4	Price Guide Test	30
4	Result	33
4.1	The Test Environment	33
4.2	The Result of Web Crawler	33
4.3	The Parameter Adjustment Product Matching	33
4.4	The Result of Product Matching	35
4.5	The Result of Price Guidance Test	36
5	Discussion	39
5.1	Discussion about the Production Matching	39
5.2	Discussion about the price guidance test	40
	Bibliography	43

List of Figures

2.1	International Standard of HTML file	7
2.2	HTML DOM Tree	7
2.3	Meta Characters	9
2.4	Product Name Identification	14
2.5	Category Mapping Algorithm	15
3.1	Screenshot of Farmers Website	19
3.2	Activity Diagram of Information Storage	21
3.3	Clean Data from Farmers	24
3.4	Clean Data from the Warehouse	24
3.5	Tokenization Example	25
3.6	Recommended TF-IDF weighting schemas	27
3.7	Distances between two websites	29
3.8	Screenshot of Website A	31
3.9	Screenshot of Website B	31
4.1	Plot of Matching Scores	35
4.2	Sales of Aptamil	37
4.3	Sales of Anchor	37

List of Tables

2.1	Regular Expression Matches	8
2.2	Example of key-value pare	16
3.1	Tools of Web Crawler	18
3.2	Product Detail of Farmers Website	19
3.3	Product description of Farmers and Warehouse	23
3.4	Vectors of Three Documents	26
3.5	Description of Two Products	28
3.6	Adjusted Parameters	30
3.7	Basic Information of Four Website	30
4.1	Hardware Enviroment	33
4.2	Software Environment	34
4.3	Result 1 of Web Crawler	34
4.4	Result 2 of Web Crawler	34
4.5	Scores of Product Matching	35
4.6	Distance Results of Product Matching	36
4.7	Accuracy of Matching Product	36

Chapter 1

Introduction

1.1 Case Introduction

With the quick development of internet, people can retrieve information from the internet much more conveniently and easier than before. But facing a vast amount of information and unsorted resources, finding an efficient way to get the useful information has become internet users "number one" issue. Though search engine has provided a convenient way for users by just type some key words simply. But after reviewing the searching result we can find out that there will be thousands of records which may match the key words. Which result is the answer for a persons specific requirement is a challenge for us to handle.

On the other side, online business has become one of the biggest part of networking event. Till the end of 2013 the output value of Business to Business (B2B) E-commerce has come to fifteen trillion dollars. At the same time the Business to Consumer (B2C) E-commerce grew more quickly and it output value has exceeded 1.2 trillion dollars. With the much more frequently E-commerce activities, there are kinds of business data having been generated. Not only consumer but also business mans are looking forward to take advantage of these useful data.

There is a big retailer store, the Warehouse, want to acquire as much as information and use them. The Warehouse is a large retail brand founded in New Zealand by Stephen Tindall in 1982. As a local retail brand, it soon became a popular institution and changed Kiwi shoppers landscape. In 2014, The Warehouse Group was set to grow the business, with a clear strategy leading to brands of Noel Leeming, Torpedo 7 in retail market.

A large amount of retailers occupies a high proportion of the market. Therefore, price will be an impact in attracting customers. If The Warehouse can acquire the real-time price, it will be possible for them to show that their prices are more competitive. Furthermore, it can put those competitors prices on its own websites can convince the customers that it has the cheaper prices. Therefore, the warehouse wanted to realize was a simple way for them to monitor the prices of equivalent products on competitor websites.

But we can find out that the general search engines which are based on general web crawler may unsatisfied with some special requirement by some specific user like the Warehouse. There are still many unsatisfactory features of general search engine.

- General search engine may not understand the purpose of users so it will return a

great amount of result that the users don't care about.

- Because of the purpose of general search engine, it will try its best to retrieve as much web pages as it can, the efficiency of retrieving specific information will be low.
- General search engine only records data at a certain point, it can't reflect a history of an activity, such as the price change history of a product.

To solve the weakness of the general searching engine, we propose a frame of crawling the E-commerce websites, extracting the information and classifying them to store in database for future analyzing. According to provide organized data, we hope to help the company be more competitive.

1.2 Research Problem

From the case, these two following questions need to be handled.

First, each company has its unique Enterprise Resource Planning(ERP) to manage their business activities, including product planning, manufacturing or service delivery, marketing and sales, inventory management, shipping and payment[33]. Product management as a major part of business process, each store uses different way to record and describe a product. For example. Both the Warehouse and Farmers sell Lego Classic Bricks. Not only name, the description and SKU is also different. In the warehouse, the product name is LEGO Classic Creative Bricks 10692 and the SKU is 5702015355704. In Farmers, its name becomes Lego Classic Bricks and its SKU is 5995019. Their descriptions also are different. In this situation, matching them directly between sites is very hard. How to match products efficiently?

After getting the competitors' data. The warehouse would like to be able to show that they are cheaper than specific competitors. What happens if we put others' retail store price on a website? How does it impact the two websites?

Chapter 2

Literature Review

2.1 Review on E-commerce Website

In an E-commerce website, the seller will try to enrich the related information, including, price, shipment, description, advertisement, to attract customers. In order to promote sales item and improve sales, the web page of E-commerce generally have the following features. The product is shown with the clear tree structure. A typical Ecommerce web site may look like this, the search form and the category list show in an obvious position and can be easily found out. The search form can help consumers retrieve what they want to buy when they have the information of the product. The consumer can also reach the target product page conveniently without key word by using category list. A clear category structure can help consumer arrive at the target product detail page quickly. The most of information of the product show on the product detail page. The product detail page is the main channel for seller to advertising their products which will impact the behavior of purchasing directly. All the information about this product will show on this page. It will be the main target of the web crawler. There will be a lot of none product page and link on the website. These pages may be imported for the E-commerce platform website, such as contact page, policy page, but for web crawler, they are useless. It is quite important to filter out this kind of pages.

By analyzing the character of E-commerce website, we can dig out the rule of filter which is embedded on the website. Then, we can build a more effective web crawler.

2.2 Review on Website Crawler

2.2.1 Brief Review of Crawler

Web crawler also can be call as web robot or spyder.it is a program which can analyze the website ,fetching the destination URL from the site HTML file and download websites automatically and iteratively[38]. A whole process of a human being want to get information or resource from internet start with typing a correct website address and then the user looking for the particular information receive it or click the link to jump to another site base on the guidance. Web crawler emulates what human do to receive information on the internet. Generally, a crawler contains the following parts: URL storage part, it is used to store the list of URLs which is going to be analyzed. Downloading part, it is used

to download the website by using HTTP protocol. Website analysis part: it is used to get the link from the html file. URL judgement part, it is used to determine whether the link URL should be crawled in the next step[36], a complete process of crawling website normally starts from a seed website. Based on breadth-first principle, crawler will extract every link from the website. URL judgement part will determine the URL hasnt been crawled before, the URL will be put them into a list and store by URL storage part. The crawler will execute the process iteratively till the result achieves some goals which have been setting at the beginning. Web crawler mainly uses list as its data structure. The list is full of the URL which havent been visited. The designer of the crawler can set the priority level of the URL that make the crawler achieve the goal faster.

2.2.2 Category of Crawler

Base on the rule and the algorithm, web crawler can be classified as the following part.

scalable web crawler This kind of web crawler mainly helps search engine to collect data. It doesnt just focus on some specific area, it will crawl the whole internet to extract not only web page, but also picture, audio, video or other kinds of data. This kind of crawler pursuit the biggest number of website to visit, so generally it will be setup on supercomputers to satisfy its crawl speed and parallel work requirement[18]. There are some weaknesses of scalable web crawler, such as there will be a lot of irrelevant information in the result and the requirement of equipment and network speed is too high. But this kind of crawler try it best to provide as much content for user, it is the cornerstone of the search engine. Google Crawler is one of the most famous typical scalable web crawler.

focused crawler Focused crawler is a kind of crawler which filter out the irrelevant URL base on the algorithm of web site analyses, and follow the specific rule to select the goal URL in the list for crawling[3]. In this kind of crawler, there will be a part of analyzer which is responsible for analyzing the existing URL to guide the next step of crawling. Unlike the scalable web crawler ,the focused crawler will provide a more specific result , it will not try to visit the whole URL in the URL list . So the hardware requirement will be much lower. From the characteristic of the focused crawler, we can find out that the focused crawler needs to handle the following problems.

The problem of how to define and describe the topic. During the process of focused crawler crawling, there must be a topic of reference for comparison. If the similarity reaches to a point, the crawler will put it into the URL list. So, it is important to describe and define the topic.

The sequence of URL for crawling. Generally, the crawler will calculate the similarity base on some algorithms and then arrange in order of the similarity. It will help to increase efficiency and the constraint of the topic by proposing an appropriate algorithm.

The problem of cover rate. In a real-life website, two related pages may not be linked directly, there may be some irrelevant pages between them. So, how

to find the related pages which are hide behind the irrelevant pages is also a problem.

The problem of the balance between accurate and efficiency. On one side, the more complex algorithm can help to increase the accuracy but on the other side, it must decrease the efficiency of the crawler when it face a tremendous amount of internet resource. It is also significant to find a point to keep the balance between the accuracy and efficiency.

Deep web crawler Though, through scalable web crawler we can get a great number of websites, there are still a number of sites that the crawler cant achieve. Deep website mainly contains four parts[23]:

1. the dynamic pages which need to query the database by fill the form.
2. the pages havent been index by search engine.
3. the pages contain some content need to get permission.
4. some accessible non-website file, such as pictures, PDF files or document files.

2.2.3 Strategy of Crawler

The searching strategy of a website crawler:

Breadth first strategy(BFS) It is a kind of strategy consider the whole link of the websites as a tree. Each link can be seen as a node of the tree. BFS gives the higher priority to the node which is longer distance to the crawling node. This strategy makes the crawler try to crawl as much as websites. Though it is a simple strategy, but with the number of the visited websites grow up, it will require more hardware resource to judge whether the next page has been visited before.

Depth first strategy(DFS) This strategy will make crawler start with seed node and give the child node higher priority to let crawler not come back until it arrives the end of each path. And then the crawler will go back to the recent branch to achieve another end of the others path. While the website is nested structured, this strategy can easily get the structure of the whole website for further analyzing. But when the different pages link each others without following the structure, the crawler may be trapped and cannot come back through the correct path.

Customize setting Following this strategy, the crawler will only visit the website base on the requirement of the user. The crawler will calculate the degree of correlation between the site on the list and the target site.

2.2.4 Recently Research about Focused Crawler

In 1998, Robert C. starts to make research on site-specific web crawler and propose a typical think of how to crawl more accurate information[35]. Soumen Chakrabarti proposed the concept of focused crawler[3], he tried to a method of classification and distillation to realize the process of crawling and design a frame of focused crawler which is filter out irrelevant pages by calculating the similarity. Diligenti M. et al tried to build context graphs

of the pages to measure the similarity[11]. PageRank is a kind of algorithm which uses quality of the link on the page to calculate the quality of the page itself. If a lot of useful page link to a specific page, this page also is useful[19]. This algorithm has been applied in Google search engine and has been optimize and improved. Soumen Chakrabarti found out that the DOM features can help to accelerate crawling[4]. YounngSig Choi applied Support Vector Machine (SVM)[7], Zhumin Chen applied Hierarchical Taxonomy[6], Piotr Dziwinski applied ant colony algorithm[13], Hongyu Liu applied Hidden Markov Model (HMM)[22] into focused crawler in order to prevent the theme-drift problems. Ontology is good for representing a crawling topic, so the crawler based on ontology has become new research area. The features of ontology can satisfy the requirement of a topic of reference for comparison. Ehrig M. Proposed a frame of ontology based focused crawler[24]. Maedche A. combined the documents and relational metadata with Ontology crawler to calculate the similarity[37]. At the beginning of crawling, the reference of a topic may be not so apropos. With the crawling process, the reference can be modified and optimized by analyzing and astatizing the crawled data. The ontology can also be adjusted. Dong, H. et al. propose a framework for a novel semi-supervised ontology-learning-based focused (SOF) crawler which has capability to automatically enrich the content of ontologies by applying SVM model[12]. Chang Su et al. present an ontology embedded algorithm to evaluate the pages relevance to the topic. Hai-Tao Zheng et al. proposed a learnable focused crawling framework based on ontology[41]. An ANN (artificial neural network) was constructed using a domain-specific ontology to classify web pages. Learning from link structure is also a direction in the research. But unlike focused crawling, it is not necessary to provide representative topical examples, since the crawler can learn its way into the appropriate topic Review on website extraction[34].

2.3 Review on Website Related Technology

2.3.1 HTML

Hyper Text Markup Language (HTML) is a kind of markup language use for creating web page. HTML can making information structured by using some tags, such as title, paragraph, and list. To a certain extent, it can also help to describe the outside of the web page. Initially, HTML is defined by Tim Berners Lee, and then developed by Internet Engineering Task Force (IETF). Finally, it became an international standard and maintained by World Wide Web Consortium (W3C). Below is International standard an example of HTML file.

As a markup language, HTML uses different tags to describe different document content. As a result of a different browser may support different version of HTML, Cascading Style Sheets (CSS) was generated to provide a new way to edit websites content and appearance separately. But the information of website appearance is not the goal to crawler; we should filter them out while crawling.

2.3.2 XML

Like HTML, Extensible Markup Language (XML) is also a kind of Markup language that defines a set of rules for encoding documents in a format that is both human-readable

```

<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>

```

Figure 2.1: International Standard of HTML file

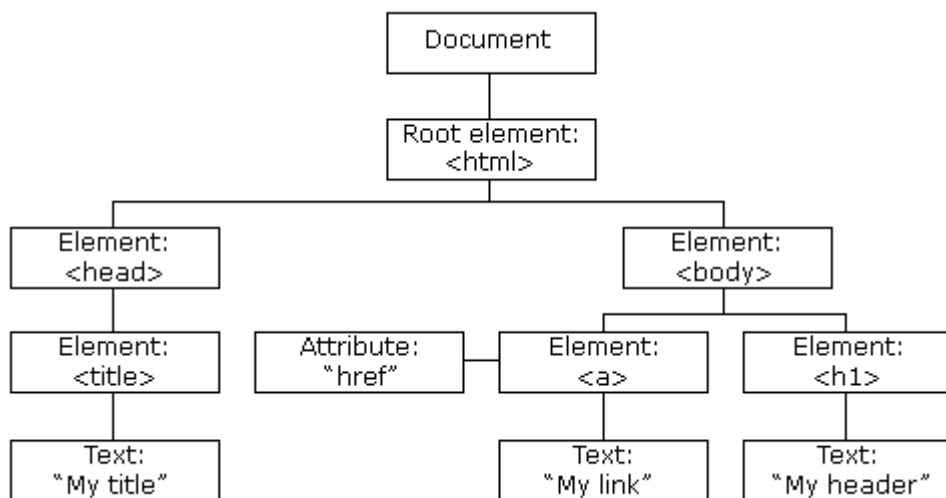


Figure 2.2: HTML DOM Tree

and machine-readable (<https://en.wikipedia.org/wiki/XML>). XML is designed to pass and carry data. It focuses on describing what kind of data does it carry and transferring the information of data including rich documents, metadata and configuration files. HTML specifies that all the tags are already pre-define but XML only provide a serious of rules to define the tags. Users can generate their own tags to identify different content. That is why XML can describe far more complex information.

2.3.3 DOM

The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. In a DOM tree, every elements are nodes including HTML attributes. Here is The HTML DOM Tree of Objects: Base on the figure above, we can see each node except document node has it parent node. When two nodes share a same

Regular expression	$(^{([0]2\ 4)})(6,7\$)\ $
Matched	(021)1234567 — (0411)123456 — (000)000000
unmatched	(123)1234567 — 025123456 — 0252345678

Table 2.1: Regular Expression Matches

parent node, we can say they are brother nodes. Different nodes which have different parents node can also be brother nodes if their parent node in the same level. DOM have following features:

1. whole model will be stored in computer memory while editing.
2. It is easy to manipulate, such as, it supports delete, insert, modify and some other manipulation. These feature make DOM suit for operation of manipulating the XML or HTML file frequently.

2.3.4 Regular Expressions

Regular expression is a pattern which is built by a sequence of character to matching some specific strings. Regular expression is made by some general characters and metacharacters. The metacharacters contain some special meaning in a regular expression. Regular expression is mainly used to complete the follow function.

1. It tests whether the string is matching some pattern, such as test a series number is a format correct phone number or ID number. Table 2.1 is an example of a regular expression of matching phone number. In this example, means the number must start with number 0 and the second number is 2 or 4, the first series number must in a bracket, means the second series number only can have six or seven numbers.
2. Find and then delete or replace the specific string. Figure 2.3 shows the meta-characters and their description.

2.4 Web Extraction

2.4.1 Brief Review of Web Extraction

Information extraction (IE) is a task of organizing the unstructured information or semi-structured information and making them change to a kind of structured information. The input of IE is unstructured information or semi-structured information, the unstructured information is a kind of data which haven't been pre-define or organized, includes audio, video, metadata, document, web page. Semi-structured data is a form of structured data not in conformity with the formal structure of the data model which associate with a relational database or other forms of a data table, but it contains labels, other tags semantic elements or separation of records within the hierarchy of data. Extensible Markup Language (XML), JavaScript Object Notation (JSON) and other markup languages are kinds of typical semi-structured data. The output of IE is structured information which each field has a specific meaning. The structured information is easier for the further analysis. IE system will not try to understand the whole information of the input. It only analyses

Meta-character(s)	Description
.	Normally matches any character except a newline. Within square brackets the dot is literal.
()	Groups a series of pattern elements to a single element.
+	Matches the preceding pattern element one or more times.
?	Matches the preceding pattern element zero or one time.
*	Modifies the *, +, ? or {M,N}'d regex that comes before to match as few times as possible.
*	Matches the preceding pattern element zero or more times.
{M,N}	Denotes the minimum M and the maximum N match count.
[...]	Denotes a set of possible character matches.
	Separates alternate possibilities.
\b	Matches a zero-width boundary between a word-class character (see next) and either a non-word class character or an edge
\w	Matches an alphanumeric character, including "_";
\W	Matches a non-alphanumeric character, excluding "_";
\s	Matches a whitespace character,
\S	Matches anything BUT a whitespace.
\d	Matches a digit;
\D	Matches a non-digit;
^	Matches the beginning of a line or string.
\$	Matches the end of a line or string.
\A	Matches the beginning of a string (but not an internal line).
\Z	Matches the end of a string (but not an internal line).
[^...]	Matches every character except the ones inside brackets.

Figure 2.3: Meta Characters

the related part of the information and the relationship has been set at the beginning of the system designing.

IE is quite useful for extracting specific data from a lot of documents. The internet is a virtual document library. Sometimes, related information may be put on different pages in the different form. It will be very helpful if we can reorganize the data which are in unstructured form and store them as structured information.

2.4.2 The Categories of Web Extraction

There are several ways to classify IE[21]. Based on the degree of automation, IE can be classified as manual method, semi-automatic method and automatic method. Based on different theory, IE system can be classified as Nature language processing (NLP) based method, wrapper based method, ontology based method, and html structure based method. Each of above method has its advantage and weakness. Actually, an IE system may use two or more method at the same time to get a better result.

Based on NLP method Based on NLP method can be applied in the situation of the source document which contains a great amount of unstructured text, especially grammatical text. This method use a NLP technic to build extraction rule. Till now there are RAPIER[27], SRV[15], WHISK[32] and so on IE system use this method. We will take WHISK for example to explain this kind of method.

To unstructured information, WHISK will separate the source document into some semantically related text blocks according to the separator, such as punctuation. Then the system will show a text block to the user each time. User will mark the blocks which contain the content they interests. Thus, the system can analyze the grammatical components and set the extraction rule based on the user marked blocks.

This kind of NLP based method actually treats the web information as traditional text information. Besides to obtain an efficient extraction rule, this method requires massive training data.

Wrapper-based method Wrapper-based method applies machine learning technic to create a delimiter based extraction rule according to the data sample which already marked by users. A delimiter is a sequence of one or more characters used to specify the boundary between separate, independent regions in plain text or other data streams. For example, in a wrapper based IE system, the wrapper will use the context of the goal information as a delimiter. Since a wrapper is pre-defined, usually a wrapper can only deal with a kind of source data. To extract different kinds of information, a wrapper based IE extraction may need to apply a series of wrappers.

Compare with the NLP based method, wrapper based method will only apply the delimiter to find out the location of the goal information without applying NLP to try to understand the meaning of the source text. STALKER[28], SOFTMEALY[20], WIEN[21] and some the other IE systems apply wrapper based method. STALKER use embedded catalogue tree (EC tree) to represent the hierarchical relations of the

complex source content. We will try to use STALKER as an example to explain this kind of IE system how to work.

STALKER will apply sequential covering algorithm to analyze the sample pages which are pre-marked by users and the page structure information which is represented by EC tree, and create the delimiter base extraction rule progressively. EC tree is quite important in this method. It is a pattern that defined by users which is based on the web page structure. It not only can provide the information about the web pages structure but also store the information about the pattern and semantic. There are three kinds of nodes in an EC tree: root node, list node and leaf node. The root of the EC tree contains the sequence of all tokens. Each token can be considered as a represent of a piece of text of the web page. Each child nodes will inherit the sequence of tokens of its parent node. The extraction of goal content actually is a process of inferring the extraction rule of the EC tree itself. For example, the STALKER starts with traversing the entire EC tree till to goal token is found. The sequence of the tokens can be used to locate the destination of goal content.

Ontology- based method This kind of method mainly uses the source itself to realize the extraction. BYU[14] and QUIXOTE[9] are the typical system uses this kind of method.

In this kind of IE system, the source data usually belongs to some specific area which contains some particular semantic element. by analyzing the semantic element, the IE system can extract the related part. For example, a description of a computer mainly contains some attributes of this production.

2.5 Review on production matching

2.5.1 Natural Language Processing

Natural language processing (NLP) is an interaction area of computer science, computational linguistics and artificial intelligence[8]. The NLP researchers are aiming at using computers to understand and play with natural language. Natural language processing is widely used in multiple areas and aspects. We will only talk about part of its researched tasks in this paper. Natural language analysis (NLA) is one of the main branches of natural language processing[2]. Stages like tokenization, lexical analysis, syntactic analysis, semantic analysis are run into in this brunch[37].

2.5.2 Tokenization

Tokenization is a necessary process of lexical analysis to break text into useful elements. The elements including words, symbols, and phrases are called tokens. Tokenization is also known as text segmentation in linguistics[17].

English Tokenization

Tokenization is a process of segmenting text into words and sentences[17]. To pre-process text before further analyzing, segmenting text into linguistic units needs to be done. The linguistic units include words, punctuation, numbers, alphanumerics, etc. In natural

language processing, we call this process tokenization. Generally, English words are separated by blanks between words. However, this white space is not equal. For example, New Zealand is known as a country. If we simply separated it into New and Zealand, both words will direct into a very different meaning and lost its original fact. Another example could be some individual thought like rock n roll, where n is a very special expression and abbreviation.

Tokenization is an identification of basic units to be processed[40]. As we can see, it is impossible to extract correct and accurate basic units clearly without a proper and basic segregation. Only after taking these basic units for granted, we will be able to carry out a further generation or analysis. Furthermore, in this section, a miss is as good as a mile. We should prepare well and prevent any errors at later stages.

Chinese Tokenization

Different from English, languages like Chinese dont have a space between words. Nearly 1.2 billion people of the world speak some form of Chinese, which makes it quite important to consider this language variety. Some common methods to deal with this kind of language will be introduced below[29]:

Character-based segmentation Based on characters, this approach is not aimed at words, it cares about characters. The text will be classified as characters with a label to mark their positions. As Weiwei Sun claimed in Word-Based and Character-Based Word Segmentation Models: Comparison and Combination, the character-based segmentation assigning labels to the characters in a text indicating the positions and priority of the characters. A single character, which is at the begin, middle or end of a multi-character word. A linear model is always used to disambiguate character. A sequence of character labels are usually defined as:

$$\begin{aligned}\hat{y} &= \arg \max_{y \in \gamma^{\|e\|}} \Theta^T \Psi(c, y) \\ &= \arg \max_{y \in \gamma^{\|e\|}} \Theta^T \sum_{i=1}^{\|c\|} \psi(c, y_{[1:i]})\end{aligned}\tag{2.1}$$

Lexicon-based segmentation [5] The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document (Turney 2002). In this section, we can create dictionaries of words indicating the words semantic orientation. Also, the lexicon-based segmented can be created manually. First, we need to have a list of adjectives and corresponding values to put in a dictionary. After that, for a new text, the adjectives will be extracted with their value and scores.

2.5.3 Syntactic Analysis

Word Classes (Parts of Speech)

All words belong to categories called word classes (or parts of speech) according to the part they play in a sentence. The Oxford dictionary[10] listed nine main word classes

as: Noun, Verb, Adjective, Adverb, Pronoun, Preposition, Conjunction, Determiner and Exclamation. The word classes give a significant amount of information about the word and its neighbors. It also is used widely in informational retrieval stemming when we need to take morphological affixes for words. It is also helpful selecting the nouns, like product name in other words.

Parsing

Parsing is defined as a combination of recognizing inputs and assigning structure to it[1]. A parser is applied to analyze syntactic structure of text with its components. The approach to constructing parser is listed by UC Davis as below:

1. Represent source language by a meta-language, Context Free Grammar
2. Use algorithms to construct a recognizer that recognizes strings generated by the grammar. Tools like YACC can be used in this step.
3. Parse strings of language using the recognizer.

Semantic analysis is analyzing the meaning of linguistic utterances[16]. In this section, we will mainly talk about the information retrieval as a part of the semantic analysis. To respond to users requests, information retrieval is needed to retrieve and storage of text documents. For most of the systems of information retrieval, they tend to use an interpretation of the compositional semantics as their bases. The common solutions of these approaches often ignore syntactic information as below[25].

- I like what I drink.
- I drink what I like.

These two sentences mean the same in these systems. Therefore, the approaches are often referred as a bag of words methods. The bag of words is usually consisted by all the tokens of the text and some dictionaries we input.

2.5.4 Product Search by Semantic Web Technology

Vandic, Dam et al present their own platform XploreProducts.com for faceted product search[39]. Their paper involved a research by semantic web technology. They mentioned a variety of problems while searching products. The e-commerce shops use different product names, category standards, category hierarchies, currencies, review rating scales and review rating information. Each of the aspects will make it harder in product search. To solve this problem, they present two main solutions.

1. Identical product recognition to aggregate information across different sources. By product name identification, we are able to determine if two products are given names equally. The Levenshtein distance is used here to measure the difference between two strings. By the product name identification algorithm, they reach 91% in precision, accuracy, recall and specificity, which presents us an example to solve the product name issue.

Algorithm 1 Product name identification

Require: The input: product names a, b
Require: The out: *true* if the product names a and b represent the same product, *false* otherwise
Require: The parameters:

- α , the threshold for the cosine similarity of the product names (*nameCosineSim*)
- β , the weight for the cosine similarity of the product names (*nameCosineSim*)
- δ , the weight for the cosine similarity of the identified ‘model words’ in the product names (*modelWordSimVal*)
- ϵ , the threshold for the final similarity between two product names (*finalNameSim*)

Require: *calcCosineSim* (a, b) gives the cosine similarity, as defined by Equation 2, between product names a and b
Require: *extractModelWords* (a) returns a set of ‘model words’ (words which contain both numeric and alphabetic characters), extracted from product name a
Require: *avgLvSim* (X, Y) returns a similarity $\in [0, 1]$ between two sets of ‘model words’ X and Y
Require: *avgLvSimMW* (X, Y) returns a similarity $\in [0, 1]$ between two sets of ‘model words’ X and Y , considering only elements $x \in X$ and $y \in Y$ such that the non-numeric parts are approximately equal ($x \approx y$).

```

1: nameCosineSim = calcCosineSim ( $a, b$ )
2: if nameCosineSim >  $\alpha$  then
3:   return true {the product names represent the same product}
4: end if
5: modelWordsA = extractModelWords ( $a$ )
6: modelWordsB = extractModelWords ( $b$ )
   {analyze the two sets of ‘model words’, comparing each ‘model word’
   from one set to each ‘model word’ from the other set (called a pair)}
7: if found a pair where non-numeric characters are approximately the same
   AND numeric characters are not the same then
8:   return false {the product names do not represent the same product}
9: end if
   {compute initial product name similarity}
10: finalNameSim =  $\beta \times \textit{nameCosineSim} + (1 - \beta) \times \textit{avgLvSim}(a, b)$ 
   {check if we have a pair of ‘model words’ which are likely to represent
   the same}
11: if there is at least one pair where the non-numeric characters are approx-
   imately the same AND the numeric characters are the same then
12:   modelWordSimVal = avgLvSimMW (modelWordsA, modelWordsB)
13:   finalNameSim =  $\delta \times \textit{modelWordSimVal} + (1 - \delta) \times \textit{finalNameSim}$ 
   {updated the calculated product name similarity}
14: end if
15: return finalNameSim >  $\epsilon$  {the product names represent the same product
   if true, false otherwise}

```

Figure 2.4: Product Name Identification

Algorithm 2 Category mapping algorithm

Require: The input: a new category c to be matched to an existing category from the framework

Require: The taxonomy T (existing categories provided by the framework, for example, the categories from Shopping.com)

Require: The function $\text{cleanAndSplit}(a)$ 'cleans' a category name, i.e., replaces 'and', 'or', '&', '(', ')', '.', and other special characters with a space and returns set of words obtained by splitting a on the space character

Require: The vector w where each elements represent the weight for the similarity, i.e., w_0 is the weight for the leaf nodes similarity, w_1 is for their parents, etc.

Require: The function $\text{getCatSim}(a, b)$ gives a similarity $\in [0, 1]$ between category names a and b

Require: The minimum similarity threshold ϕ

```

1:  $S = \{\}$ 
2: for  $target \in T$  do
3:    $a = \text{cleanAndSplit}(c)$ 
4:    $b = \text{cleanAndSplit}(target)$ 
5:    $sim = \text{getCatSim}(a, b) \times w_0$ 
6:    $i = 1$ 
7:   for  $targetAncestor \in \{\text{ancestors of } target\}$  do
8:      $cAncestor = \text{nextAncestorOf}(c)$ 
9:     if not exists  $cAncestor$  then
10:      return
11:    end if
12:     $a = \text{cleanAndSplit}(cAncestor)$ 
13:     $b = \text{cleanAndSplit}(targetAncestor)$ 
14:     $sim = sim + \text{getCatSim}(cAncestor, targetAncestor) \times w_i$ 
15:     $i = i + 1$ 
16:  end for
17:   $S = S \cup \{(target, sim)\}$ 
18: end for
  {The category that is the best match for  $c$  is the one with the highest similarity in the set  $S$ }
19: return  $\{x | (x, m) \in S, m \geq \phi, \forall (y, n) \in S : n \leq m\}$ 

```

Figure 2.5: Category Mapping Algorithm

2. Besides the product names, the category difference is also an aspect. Both the syntactic variations and semantic variations are difficulties in this section. To deal with this kind of issue, they put forward the category mapping algorithm. They compare their algorithms with Park & Kim, and the result shows that the method of Park & Kim significantly (5% significance level) outperforms our approach and that of PROMPT with respect to precision.

2.5.5 Synthesized similarity method (SSM)

As we explained, Vandic, Dam et al successfully find algorithms to match the product name and product category. However, to use it properly, it requires us to recode the product name with the formula it provided and show as much as significant information in their names, which is far harder to process real database. As far as we research, most e-commerce websites only provide some basic information including in the name. Therefore, we want to include as much specific information as we can in the product name and enhance the accuracy of product name matching. For instance, besides mobile phone name and version, we add its product dimensions, item weight, item model number and machine type (international or local). Apparently, with a large amount of data and specific product information, we will get a more accurate result.

Feature	Description
Brand	Canon
version	SX410
Color	Black
Item Dimensions	2.72*3.35*4.09 inches
Maximum Aperture Range	F3.5 - F5.6
Battery Type	Lithium Ion

Table 2.2: Example of key-value pare

To present the product priority, we will use key-value pair to link data items[26]. A key is the item of data, and the value is the data or its location. A simple example is as followed: The combined similarity of product:

$$SynthesizedSim = \alpha * titleSim + \beta * mKVPSim + \gamma * nmPerc, \alpha + \beta + \gamma = 1 \quad (2.2)$$

There are three parts of the similarity measures: titleSim, mKVPSim, nmPerc. The titleSim is based on Vandic, Dam s algorithms to describe the similarity of two product name. Since the features are either described with words or numbers. The other two parts mKVPSim and nmPerc are used to calculate the similarity of product features.

Chapter 3

Methodology

3.1 Gather Data

3.1.1 The Structure of the Web Crawler

Based on the character of the E-commerce websites, we make a focused web crawler with some optimizations and expansion of the functions in order to extract product detail from the E-commerce website.

The web crawler we made include 4 main parts:

- URL list management part: setting the initial seed URL and judge the URL to be crawl.
- Web page crawling part: crawl the website according to the URL list.
- Web page analyzing part: extract the specified content base on the setting.
- Web content storage part: building the database and store the analyzing result.

The entire process of the web crawler is: the URL list management part analyzes the URL from the seed site base on BFS, try to crawl as much as URL and put them into the URL list. After filtering out the irrelevant URL, Web page crawling part will use BeautifulSoup[31] to parse the site, and then the web page analyzing part starts to extract the goal content. In the end, the cleaned content will be store in database by web content storage part.

3.1.2 The Technique of the Web Crawler

The tool which is used in this project:

Beautiful Soup is a **Python** library for pulling data out of **HTML** and **XML** files. It works with parsers to provide idiomatic ways of navigating, searching, and modifying the parse tree. **Requests** is a **Python** library can send organic, grass-fed **HTTP/1.1** requests without the need for manual labor.

3.1.3 Setting the rule for filtering URL and extract product detail.

In this project, our web site crawler is designed for specific web site in order to extract particular content, we will set a series of rules to help crawler arrive product detail page faster.

Table 3.1: Tools of Web Crawler

Module	Tool	Programing Language /Platform
URL list management part	BeautifulSoup	Python
Web page crawling part	BeautifulSoup/ regular expression/ Requests	Python
Web page analyzing part	BeautifulSoup/ regular expression/ Requests	Python
Web content storage part	mysql.connector	Python/MySQL

In this case, after observing a series of page of The Farmers web site, we found out that the home page of The Farmers is <http://www.farmers.co.nz>. All the products are classified into the following categories: women, men, children, toys, home, electrical, gift cards, promotions, clearance. A big category page is looked like, <http://www.farmers.co.nz/beauty>. Following a big category, there will be some sub-categories, for example, a product name Panasonic 55" Full HD Smart LED TV with FreeviewPlus TH-55CS650Z, its URL is <http://www.farmers.co.nz/electrical/tv-technology/tvs-cabinets/panasonic-55-full-hd-smart-led-tv-with-freeviewplus-th-55cs650z-6049392>. Some pages which are not product pages, such as contact page looks like this: <https://www.farmers.co.nz/contactus.aspx>. In a product page (fig1), there must be some information about the product: product title, product image, price, sales price (optional), product description, product detail bullet list. Which means there will be the following tag in the product page: `< divdata – product – sku = " * * * * * " >< /div >`, `< divclass = "new – price" >< /div >`, `< divclass = "product – detail – bullet – list" >< /div >`

```

1 <div class="span6 cms-productImagery"
2 data-product-sku="6049392">
3 <h1>Panasonic 55" Full HD Smart LED TV with FreeviewPlus
4 TH-55CS650Z </h1>

```

The HTML snippet of representing the product price

```

1 <div class="new-price">$1,899.00</div>

```

The HTML snippet of representing the product detail

```

1 <div class="product-detail-bullet-list"><ul><li>Lace: 70\%
2 nylon, 20\% cotton, 10\% polyester; Lining: 100\%
3 polyester</li></ul></div>


```

So we can set some rule for crawling products detail of the Farmers website:

After apply the above rules, we can easily retrieve the specific content from the site by the help of BeautifulSoup. BeautifulSoup can parse the website source file. By using `soup.find()` function, we can easily retrieval any attribute or tag in the html file.

An HTML file which is parsed by BeautifulSoup looks like this.

Panasonic 55" Full HD Smart LED TV with FreeviewPlus TH-55CS650Z



Now \$1,899.00 Was \$2,699.00 **You save \$800.00**

Product #6049392

This 55" Panasonic Smart TV is elegantly designed, and has exceptional picture performance; pictures look crisper and more dynamic. This TV has user-friendly smart features including TV Anytime with a Twin Tuner and My Home Screen.

1

BUY NOW

SAFE AND SECURE SHOPPING

CHECK STOCK IN STORE

Find out if this product is available in a store near you

DELIVERED TO YOUR DOOR
From \$65 Nationwide. Delivery date & cost displayed on the Payment page

OVERSIZED ITEM

Description

Delivery

Returns

Key Benefits:

- Super bright Plus.
- TV Anytime.
- My Home Screen.
- Netflix.
- Built-in WiFi.
- FreeviewPlus: enjoy better picture quality thanks to an integrated Freeview HD tuner.

To receive Freeview you need a UHF aerial and must be within the coverage area. See www.freeviewnz.tv for further information on your coverage area.

*Smart TV requires access to an active broadband connection. Additional hardware may be needed.

- Full HD (1920 x 1080) resolution panel
- 139 cmv
- 3x HDMI inputs
- 2x USB inputs
- 100Hz refresh rate
- FreeviewPlus
- Manufacturer's 1-year warranty

SPECIAL OFFER

Add To Wishlist

Email A Friend

You May Also Like

Recently Viewed





Figure 3.1: Screenshot of Farmers Website

Function	Rule
Judge URL	The URL must contain <code>www.farmers.co.nz</code> .
Get product title	It is the only element of <code>h1</code> tag
Get product SKU	<code>Itisembedin < divclass = "ish - productNumber" > tag</code>
Get product description	<code>Itisembedin < divclass = "product - description" > tag</code>
Get product price	<code>Itisembedin < divclass = "std - price" > tag</code>
Get product image	<code>Itisembedin < aclass = "pintertestetail" > tag</code>

Table 3.2: Product Detail of Farmers Website

```

1 <div class="span6_cms-productImagery" data-product-sku="6049392">
2 <h1>Panasonic 55" Full HD Smart LED TV with FreeviewPlus
3 TH-55CS650Z</h1>
4 <div class="visible-phone mobile-price">
5 <div class="ish-priceContainer-salePrice
6 kor-product-sale-price" data-dynamic-block-id=
7
8 "Df6sEBaO3MgAAAFcgapmsOfA" data-dynamic-block-name="SalePrice">
9 <span class="kor-product-sale-price-value
10
11 ish-priceContainer-salePrice-value">
12 <div class="new-price">$1,499.00</div>
13 <div class="old-price">$2,699.00</div><div class="save-price">
14
15 You save $1200.00</div>
16 </span>
17 </div>
18 <div class="ish-priceContainer-scalePrice">
19 <span></span>
20 </div><div class="clearfix row-fluid"><div class="span12">
21 <div class="ish-productNumber" data-dynamic-block-id=
22
23 "Df6sEBaO3MgAAAFcgapmsOfA" data-dynamic-block-name="ItemNumber"
24 data-product-sku="6049392">
25 Product #6049392</div>
26 </div></div>
27 </div>

```

```

1 def get_soup(url):
2     url=str(url.split())[0])
3     response = requests.get(url)
4     soup = BeautifulSoup(response.content, "html.parser")
5     return soup
6 def get_description(url):
7     try:
8         soup=get_soup(url)
9         if soup.find('div', class_='product-description'):
10             for i in soup.find('div', class_='product-description'):
11                 i.string
12                 return i
13         else:
14             return None
15     except:
16         print("Unexpected_error:{ }".format(sys.exc_info()[0]))

```

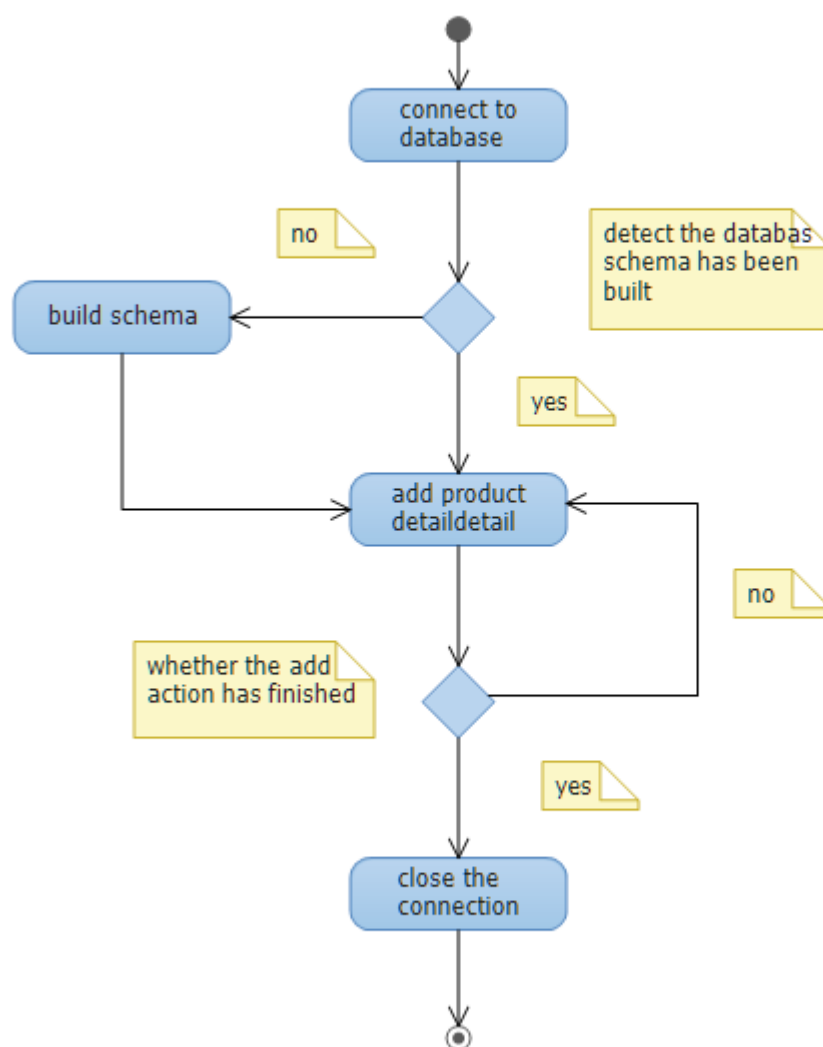


Figure 3.2: Activity Diagram of Information Storage

The code above is the snippet of retrieving description fr om the farmers product detail page.

3.1.4 Information Storage

After crawling the data from website, we need to store them into database for further analysis. Here is the activity diagram of information storage.

Build Schema

We need to define the attribute for each row in the schema. Here is the code snippet of building schema in MySQL using python.

```

from __future__ import print_function
import pymysql
from pymysql import ProgrammingError
try:
    cnx = pymysql.connect(user='root',password='***')
except ProgrammingError as e:
    print ("Caught a Programming Error:")
    print (e)
else:
    cnx.close()
DB_NAME = 'THE_FARMERS_WEBSITE'
  
```



```

TABLES = {}
TABLES['THE_FARMERS_WEBSITE'] = (
    "CREATE TABLE 'THE_WHOLE_WEBSITE' ("
    "  'url' varchar(250) NOT NULL ,"
    "  'description' text(65535) ,"
    "  'price' varchar(200) ,"
    "  'title' varchar(250) ,"
    "  'SKU' char(200) ,"
    "  PRIMARY KEY ('url') "
    ") ENGINE=InnoDB")
cnx = pymysql.connect(user='root',password='***')
cursor = cnx.cursor()
def create_database(cursor):
    try:
        cursor.execute("CREATE DATABASE {} DEFAULT CHARACTER SET 'utf8'".
                        format(DB_NAME))

    except pymysql.Error as err:
        print("Failed creating database: {}".format(err))
        exit(1)
cursor.close()
cnx.close()

```

Insert Data into Database

Here is the code snippet of insert data into database in MySQL using python.

```

def insert_data(path):
    cnx=pymysql.connect(user='root',password='***', database='
                        the_farmers_website')

    queue=deque()
    visited=[]
    queue.append(path)
    cur = cnx.cursor()
    url_query=("SELECT url FROM 'the_farmers_website'.'the_whole_website' "
              "where 'the_whole_website'.'url'='%s")
    url = queue.popleft()
    url1=(url,)
    cur.execute(url_query,url1)
    for i in cur:
        visited.append(i)
    if not visited:
        if get_SKU(path):
            print("crawl product's website {}".format(path))
            data_url=str(url)
            data_description=str((get_description(path)),)
            data_price=str((get_price(path)),)
            data_title=str((get_title(path)),)
            data_SKU=get_SKU(path)
            add_product = ("INSERT INTO the_whole_website"
                           " (url,description,price,title,SKU)"
                           " VALUES (%s,%s,%s,%s,%s)")
            cur.execute(add_product,(data_url,data_description,data_price,
                                     data_title,data_SKU))

    cnx.commit()
    cnx.close()

```

```

        return True
    else:
        print("crawl general website {}".format(path))
        add_product = ("INSERT INTO the_whole_website "
                        "(url) "
                        "VALUES (%s)")
        data_url=(url,)
        cur.execute(add_product, data_url)
        cnx.commit()
        cnx.close()
        return True
    else:
        print("already insert {}".format(path))
        cnx.commit()
        cnx.close()
        return False

```

3.2 Production Matching

On the product page, a brief description will be given for consumers to understand the product. A product description usually includes the name of the product, the basic information and the key features that can attract consumers. Therefore, to specify and highlight the attraction, different retailers may have different explanations in the description. However, for accuracy, their description will still spread the same information or similar meaning.

description	price	title	SKU	url
The Avengers***	27.99	Avengers Titan***	6005021	www.farmers.co.nz/****
Create the perfect***	59.00	Frozen Sing***	1997996	www.thewarehouse.co.nz/****

Table 3.3: Product description of Farmers and Warehouse

As we can see in both descriptions, words like Elsa, iconic Snow Queen Dress is mentioned more than once. They both described the height of the figure as one of the features. Therefore, its possible for us to find same or similar products based the description from the online retailers. To further analyzing the information we retrieved from these e-commerce websites, we will use natural language processing to process text by computer systems.

3.2.1 Clean the Data

First, we need to clean the data acquiring from websites. As we can see, the title variable of Farmers products include the package information, we will separate it into two different columns:

3.2.2 Tokenization the document

Here, we will introduce some steps of a low level tokenization:

	V1	V2	V3	V4	V5
1	url	description	price	title	SKU
2	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Florobotanica is the new fragrance from the botanical ...	72.00	Balenciaga Florobotanica Shower Gel, 200ml	5912225005
3	http://www.farmers.co.nz/beauty/bath-body-care/ba...	This pH-neutral body cleanser is gentle, thoroughly cl...	57.00	Clarins Eau Dynamisante Shower Gel, 150ml	7075876012
4	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Exquisite, gentle cleansing with this lightweight, airy f...	55.00	Clarins Eau Dynamisante Shower Mousse, 150ml	7075876005
5	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Clarins' concentrated cleansing treatment activates wit...	53.00	Clarins Relax Bath & Shower Concentrate, 200ml	7075891002
6	http://www.farmers.co.nz/beauty/bath-body-care/ba...	This concentrated cleansing treatment activates with t...	53.00	Clarins Tonic Bath & Shower Concentrate, 200ml	7075891001
7	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Calms and balances.	45.00	Dr Hauschka Almond Soothing Bath Essence, 100ml	5881282004
8	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Firms and refreshes.	45.00	Dr Hauschka Lemon Lemongrass Vitalising Bath Essen...	5881282003
9	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Soothes and protects.	50.00	Dr Hauschka Moor Lavender Calming Bath Essence, 10...	5881282001
10	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Harmonises and protects.	50.00	Dr Hauschka Rose Nurturing Bath Essence, 100ml	5881282002
11	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Refreshes and cleanses.	45.00	Dr Hauschka Sage Purifying Bath Essence, 100ml	5881282005
12	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Two embossed triple milled soaps with deliciously tro...	15.10	Pacifica Passionfruit, Papaya & Honey Boxed Soaps, 2-...	5976455
13	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Refreshing, fruity newcomer to Sukin's range.	16.79	Sukin Botanical Body Wash - Lime & Coconut, 500ml	6022264

Figure 3.3: Clean Data from Farmers

	V1	V2	V3	V4	V5	V6
1	url	description	price	SKU	title	
2	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Florobotanica is the new fragrance from the botanical ...	72	5912225005	Balenciaga Florobotanica Shower Gel	200ml
3	http://www.farmers.co.nz/beauty/bath-body-care/ba...	This pH-neutral body cleanser is gentle, thoroughly cl...	57	7075876012	Clarins Eau Dynamisante Shower Gel	150ml
4	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Exquisite, gentle cleansing with this lightweight, airy f...	55	7075876005	Clarins Eau Dynamisante Shower Mousse	150ml
5	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Clarins' concentrated cleansing treatment activates wit...	53	7075891002	Clarins Relax Bath & Shower Concentrate	200ml
6	http://www.farmers.co.nz/beauty/bath-body-care/ba...	This concentrated cleansing treatment activates with t...	53	7075891001	Clarins Tonic Bath & Shower Concentrate	200ml
7	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Calms and balances.	45	5881282004	Dr Hauschka Almond Soothing Bath Essence	100ml
8	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Firms and refreshes.	45	5881282003	Dr Hauschka Lemon Lemongrass Vitalising Bath Essence	100ml
9	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Soothes and protects.	50	5881282001	Dr Hauschka Moor Lavender Calming Bath Essence	100ml
10	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Harmonises and protects.	50	5881282002	Dr Hauschka Rose Nurturing Bath Essence	100ml
11	http://www.farmers.co.nz/beauty/bath-body-care/ba...	Refreshes and cleanses.	45	5881282005	Dr Hauschka Sage Purifying Bath Essence	100ml

Figure 3.4: Clean Data from the Warehouse

You have an input document that contains the following sentence:

"S&P today affirmed A+/A-1 ratings on China Light & Power."

The tokens are as follows (in order of creation, as lowercase adjacent terms with reasons):

Token	Description
s	Non-spaceless character
&	Single ampersand
p	Non-spaceless character
today	Contiguous non-spaceless characters
affirmed	Contiguous non-spaceless characters
a	Non-spaceless character
+	Single plus sign
a	The front slash (/) in "A+/A-1" is not concorded (that is, removed)
1	The hyphen in "A+/A-1" is not concorded (that is, removed)
ratings	Contiguous non-spaceless characters
on	Contiguous non-spaceless characters
china	Contiguous non-spaceless characters
light	Contiguous non-spaceless characters
&	Single ampersand
power	Contiguous non-spaceless characters. The period after "power" is not concorded (that is, removed).

Figure 3.5: Tokenization Example

1. Segment text in sentences and words. During this step, the white spaces between words will be replaced and the trailing quotation marks will be cut off.
2. Deal with abbreviations. In There is supposed to be a widely accepted standard of abbreviations for linguists to use. However, we still dont have a standard like this. The normal process of abbreviation nowadays is to maintain a dictionary of existing abbreviations. Naturally, the dictionary of the abbreviation affects accurate of the text being tailored. But sometimes, the same abbreviation may represent different meaning by the words around, which needs us to pay extra attention while creating the dictionary.
3. Deal with hyphenated words. Segmenting the hyphenated into one or two words is always a question. Most of the hyphenated words are segmented depending on the text. A usual process would be treating them as a single syntactic unit and tokenized as single tokens. Another approach is tokenized the hyphenated word separately if all the parts in it have their own meaning. For instance, 22-year-old can be tokenized into 22, year and old since we can still understand the information it expresses.
4. Special expressions. We sometimes will meet Date, time, URL and other special expressions. If we are not taking right processes, these can be very confusing. A normalization of the formal will be used in some pre-processor and transfer the entire format into the same one.

Below is a tokenization example:

By using packages from R, we are able to tokenize the sentences into tokens. Here is an example of Farmers product:

Florabotanica is the new fragrance from the botanical gardens of Balenciaga, inspired by the woman who is beautiful but dangerous like many rare botanical flowers.

	Comvita	Manuka	Honey	Active	UMF	18	Lozenges	Childrens	Natural	Lollipops
D1	1	1	1	1	1	1	0	0	0	0
D2	1	1	1	0	0	0	1	0	0	0
D3	1	0	0	0	0	0	0	1	1	1

Table 3.4: Vectors of Three Documents

"Florabotanica" "is" "the" "new" "fragrance" "from" "the" "botanical" "gardens" "of" "Balenciaga" "inspired" "by" "the" "woman" "who" "is" "beautiful" "but" "dangerous" "like" "many" "rare" "botanical" "flowers"

3.2.3 Syntactic Analyses

By tokenization, we are able to get words as the foundation of language processing. Jurafsky and Martin compared syntax to the skeleton, which describes the relationship between words. The way how words cluster into classes, groups with their neighbors and depend on others in a sentence. A lot of algorithms will be introduced to deal with this knowledge. After generating all the words from the document, a corpus of Farmers product is built. Then, we will remove the stopwords, punctuations and stem the words. In this part, word accurate and accuracy will be treated as the same word.

To do statistical analysis, a corpus will be built as a large and structured set of texts. We will also apply some syntactic analysis to make sure the words are classified and transferred properly.

3.2.4 Vector Space Model

Since we are going to find similar products on two different websites, we will build two matrices to represent the vectors from these two websites. The rows are products number while the columns are terms from the corpus.

In vector space model, all the documents and texts will be represented as vectors. Each vector contains the terms that occur within the bag.

The vector for document can be described as below:

$$d_j = (\omega_{1,j}, \omega_{2,j}, \dots, \omega_{t,j})$$

The dimensions of vectors represent the terms of document. The value of each dimension corresponds to the frequency of each term.

Here is a simplified example of the vector space model. The example documents are:

- D1: Comvita Manuka Honey Active UMF 18+
- D2: Comvita Manuka Honey Lozenges
- D3: Comvita Childrens Natural Lollipops

The vectors of three documents are shown as followed: In this section, all the vectors are combined with 0 and 1. However, in different situations, some terms are more important than others. For example, a marketing researcher may concentrate on the brands

Recommended TF-IDF weighting schemes		
weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Figure 3.6: Recommended TF-IDF weighting schemas

while a product developer is looking for similar products. Therefore, we need to add weights on these terms in order to satisfy all requirements.

3.2.5 Term Weight

The method to assign term weight in the documents has two factors: term frequency within a single document, and the distribution of terms across the whole collection.

To begin with, we can simply consider the terms occur more frequently are reflecting more importance than those less frequently. Thus, these terms should have higher weights. The factor here is called term frequency, which represents the frequency of a term within the given text.

Another factor associated with term weighting will be the distribution of terms across the dictionary as a whole. Assume we have N documents in the collection, and n_i represents the number of documents term i occur. This factor is called inverse document frequency term weight. The equation is:

$$idf_i = \log\left(\frac{N}{n_i}\right) \quad (3.1)$$

The combination of these two term frequency is defined as:

$$\omega_{i,j} = tf_{i,j} \times idf_i \quad (3.2)$$

Here are some common schemes of TF-IDF weighting[30]:

3.2.6 Measurement

Measuring the similarity of two texts can be converted into calculating the distance between two vectors. The most common measurement is cosine similarity:

$$\cos \theta = \frac{d_2 \bullet q}{\|d_2\| \|q\|} \quad (3.3)$$

Title	Number(capacity)
Clarins Relax Bath & Shower Concentrate	200ml
Clarins Tonic Bath & Shower Concentrate	200ml

Table 3.5: Description of Two Products

Using cosine similarity with TF-IDF weights:

$$sim(d_j, q) = \frac{d_2 \bullet q}{\|d_2\| \|q\|} = \frac{\sum_{i=1}^N \omega_{i,j} \omega_{i,q}}{\sqrt{\sum_{i=1}^N \omega_{i,j}^2} \sqrt{\sum_{i=1}^N \omega_{i,q}^2}} \quad (3.4)$$

For a more accurate result, we will combine TF-IDF weights with SSM as previously indicated. Each document vector $d_j = (\omega_1, \omega_2, \dots, \omega_m, \omega_{m+1}, \dots, \omega_n, \omega_{n+1}, \dots, \omega_t)$ is split into three parts:

$\omega_1 - \omega_m$: Product title information $\omega_{m+1} - \omega_n$: Product key-value pair information $\omega_{n+1} - \omega_t$: Product number information

The new similarity will be:

$$sim(d_j, q) = \alpha * \frac{\sum_{i=1}^m \omega_{i,j} \omega_{i,q}}{\sqrt{\sum_{i=1}^m \omega_{i,j}^2} \sqrt{\sum_{i=1}^m \omega_{i,q}^2}} + \beta * \frac{\sum_{i=1}^n \omega_{i,j} \omega_{i,q}}{\sqrt{\sum_{i=1}^n \omega_{i,j}^2} \sqrt{\sum_{i=1}^n \omega_{i,q}^2}} + \gamma * \frac{\sum_{i=1}^t \omega_{i,j} \omega_{i,q}}{\sqrt{\sum_{i=1}^t \omega_{i,j}^2} \sqrt{\sum_{i=1}^t \omega_{i,q}^2}} \quad (3.5)$$

3.2.7 Calculating Vector Similarity

Each product contains three different vectors: title, key-value pare, numbers. Since title usually contains the most important information, we set $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$ The sample distance matrix is shown as below:

Where c1 represents website 1, c2 represents website 2. In order to find similar products quickly, we calculated all the distance between every two products previously.

3.2.8 Find Similar Product

Once typing the product number of website c1, the products from c2 with least distance will show up.

Descriptions:

- Clarins' concentrated cleansing treatment activates with the heat of a warm bath or shower-releasing the soothing, aromatic virtues of Basil, Camomile and Petit Grain.
- This concentrated cleansing treatment activates with the heat of a warm bath or shower-releasing the invigorating, aromatic virtues of Rosemary, Mint and Geranium.

The distance between these two products is 2.645751.

3.2.9 Adjusting Parameters

However, the parameters of similarity affect the results. It depends on the definition of similarity the users are looking for.

	c1	c2	distance
1	1	2	6.324555
2	1	3	6.000000
3	1	4	5.385165
4	1	5	6.000000
5	1	6	6.403124
6	1	7	5.656854
7	1	8	6.000000
8	1	9	6.324555
9	1	10	5.916080
10	1	11	6.000000
11	1	12	5.916080
12	1	13	6.324555
13	1	14	6.855655
14	1	15	6.000000
15	1	16	5.830952
16	1	17	6.557439
17	1	18	6.480741
18	1	19	6.633250
19	1	20	5.916080

Figure 3.7: Distances between two websites

Group number	Title	Key value pair	number
1	0.8	0.1	0.1
2	0.7	0.2	0.1
3	0.7	0.1	0.2
4	0.6	0.1	0.2
5	0.6	0.2	0.2
6	0.6	0.3	0.1
...
36	0.1	0.8	0.1

Table 3.6: Adjusted Parameters

	Name	Date of Found
Website A	health****.co.nz	July, 2015
Website B	*****health.co.nz	Mar, 2014
Website C	***shop.co.nz	Nov, 2015
Website D	Health***.co.nz	Jan, 2016

Table 3.7: Basic Information of Four Website

Our parameters groups are shown as below:

3.3 Price Comparison

To apply our product similarity algorithm, we will use two products databases as an example to show a real-life example. However, when we look into the products databases of Farmers and the Warehouse, we find out that only 1% products are duplicated, which makes it hard to calculate the accuracy. For instance, since natural healthy products are popular in New Zealand, we find four online retail websites to further explore the results. Here is some basic information of these websites:

These are some key features about these stores:

1. They sale almost the same products. All of these stores sell more than 1000 products.
2. Although these websites are running in New Zealand, their target customers are Chinese. The customers dont know much about their brands, which makes product price an important impact in their purchasing behavior.

By web crawler, we extract product details from each website and build four databases A D to do our product matching test. In this section, we will use database A as our main searching key word. The products from other database, which is similar to the one from database A will be shown with their similarity value. To confirm the matching accuracy, we will check the results manually.

3.4 Price Guide Test

In order to find out the impact of putting the competitors price on the website. We choose two online retail websites A and B to do a pilot test. Website A has been established for



Prolife羊奶片巧克力味 500片
 Pro-Life Goat's milk & calcium 500 chewy tablets chocolate
 促进儿童生长发育 增强抵抗力 促进大脑发育 易吸收 description

商品编号: 5456 product number
 品牌: Pro Life 专业生活 brand
 商品重量: 535.00克 weight(gm)
 市 场 价: ~~NZ\$40.00~~ other website price
 销 售 价: **NZ\$ 38.60** price in this website
 约合人民币: ¥ 173.7 Chinese yuan
 当前汇率: 1新西兰元=4.50人民币 exchange rate
 该商品已被收藏 0 次, 已经被购买 0 件 (库存 0)
 可用性: 缺货(补货中) out of stock

添加到购物车



Prolife 羊奶片-巧克力味 500粒
 Pro-Life Goat's milk & calcium 500 chewy tablets chocolate

\$40.00

500 Tabs

1

+
-

加入购物车

♥

Add to Wishlist

编码: 9400514011133

Figure 3.9: Screenshot of Website B

two years while website B has been released for one year. Both these two website have their stable sales volume. These two website can be considered as a pair competitor. But one of the important situations is these two websites belong to the same company and most of the consumer don't know the relationship of these two websites which means we can track the sales data conveniently and all the price setting and the test will not impact the whole sales volume of this company. We are going to put some price information of website A on website B, and track the traffic of each website to see if customers will click website B through the link we put. We only put the website A product prices on website B when it is higher.

Period: June 2016 C July 2016

We randomly choose 120 products which are sold on both websites.

Figure 3.10 is the screenshot of website A. Figure 3.11 is the screenshot of website B.

Chapter 4

Result

4.1 The Test Environment

The software environment used in this test is:

4.2 The Result of Web Crawler

We try to crawl the Farmers and the Warehouse websites. Test time is 4 hour per round. We run 4 rounds to confirm the web crawler run well. The result can be seen in Table 4.3

After analyzing these two data resource, we find out that only 1% products of these two websites are duplicated. We decide to crawl four online retail websites. Deal to the product number is not bigger than the Farmers and the Warehouse. We run the crawler 2hours per round.

4.3 The Parameter Adjustment Product Matching

We choose five different products from each websites to show all the situation which happened during the process of production matching. We choose one product t from website A. For website B, C, D, each website has four products: the same product to product t, different product but under the same brand with t, similar product with different brand, none relative product.

A: Thompson's Grape Seed 19,000 One-a-Day Tablets 120 B:

- Thompsons Grape Seed 19000 One A Day 120 Tablets
- Thompsons Astamax One A Day 30 Capsules
- Swisse Ultiboost Grape Seed 180 Tablets

CPU	Intel Core i7-3620HQ @ 2.6Ghz
Memory	8G DDR3
Hard Drive	256G SSD
Network	20M optical fiber

Table 4.1: Hardware Enviroment

Operation system	Windows 10 Profesional
IDE	Eclipse Mars Release (4.5.0)
Database	MySQL
Program Language	Python

Table 4.2: Software Environment

Website	Product Number
The Warehouse	3920
Farmers	11982

Table 4.3: Result 1 of Web Crawler

- GO Healthy Go Magnesium Sleep

C:

- Thompson's Grape Seed 19,000 One-A-Day
- Thompson's Vitamin C 500mg Chewable
- Clinicians New Zealand Grape Seed 60,000mg 120 capsules
- GO Healthy Go Antibiotic Support 14 Vegecaps

D:

- Thompsons Grape Seed 19000 One-A-Day Tablets 120
- Thompsons Cholesterol Manager Tablets 120
- Good Health Grape Seed 25000 Extra Strength Antioxidant Capsules 120
- Go Healthy Celery 8000 VegeCapsules 120

We are going to search Thompson's Grape Seed 19,000 One-a-Day Tablets 120 in database subset B, C and D. If results show up from group1, which means the results is correct. We will score 10. If results show up from group2, where the products are either from the same brand or similar products, we will score 5.

If products from group 3 show up, where the products are nothing in common, we will score 1. The final scores are:

As we can see from table 5.3.1, the two matching results are almost the same, but they return to different distance. The reason is that when we acquire the text of product, 5000

Website	Product Number
health****.co.nz	1281
*****health.co.nz	1569
***shop.co.nz	1022
Health***.co.nz	978

Table 4.4: Result 2 of Web Crawler

Group number	Title	Key value pair	number	scores
1	0.8	0.1	0.1	10
2	0.7	0.2	0.1	10
3	0.7	0.1	0.2	10
4	0.6	0.1	0.2	8.3
5	0.6	0.2	0.2	8.3
6	0.6	0.3	0.1	10
...
36	0.1	0.8	0.1	4

Table 4.5: Scores of Product Matching

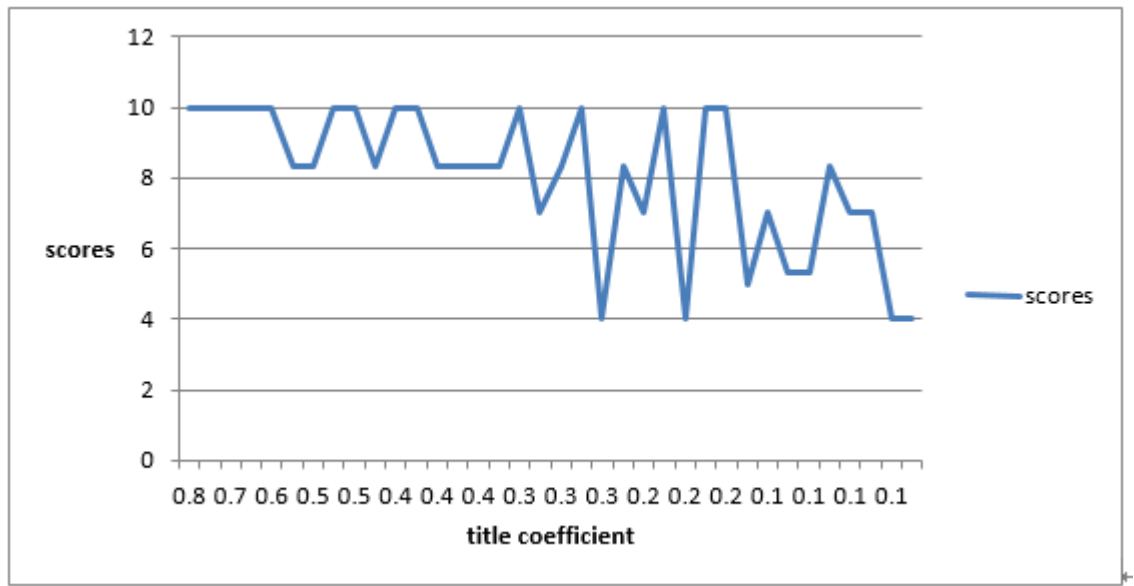


Figure 4.1: Plot of Matching Scores

and 60 are recognized as one word. This makes 500060 is different from 5000mg 60 in the numeric parameters part. At this stage, we are only able to solve these format problems manually, which will spend a lot of time. We also select 632 duplicated products in these four websites and calculate the matching accuracy manually. All of these four websites have a high correct accuracy. This ensures our reliability on these products.

The parameters adjustment part shows an impact of adjusting numbers. If we set title coefficient higher than the others, all the similar products will be from the same brand but sometimes return to different products.

The trend is not very clear, but we can still see the scores are getting lower when we decrease the title coefficient. By adjusting parameters, the website owners are able to decide which is more important when matching the products.

4.4 The Result of Product Matching

Since there are not enough duplicated products on the Warehouses and the Farmers website. We make a product matching test for the four online retail shops.

Shop	Product title(translated)	Distance
Health***.co.nz	Thompsons Celery One-A-Day 5000mg 60 Capsules	1
***shop.co.nz	Thompsons Celery One-A-Day 500060 Capsules	1.4324555

Table 4.6: Distance Results of Product Matching

Shop	Matching Products	Percent
health****.co.nz	589	93.1%
*****health.co.nz	538	85.1%
***shop.co.nz	576	88.2%
Health***.co.nz	520	82.3%

Table 4.7: Accuracy of Matching Product

Here is one example of product matching.

Input: "Thompson's Celery 5000 One-a-Day Capsules 60 (item from shop *****health.co.nz)"

Output:

After calculating the whole database, we found there 632 duplicated products in these four online shop actually. The matching accuracy reach the peak when we set the parameters $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$. The matching result can be seen in the following table.

4.5 The Result of Price Guidance Test

The test lasted a month from June 5th to July 5th. We chose two popular products: Aptamil Milk Powder and Anchor Milk Powder. The Sales information is monitored by the administration of two websites. Although we choose 120 products as our samples in this test, most of the products are sold less than 20 pieces in this month or only sold in less than five days, which makes it hard to track and compare the sales data between website A and B. Therefore, only two products have adequate selling data, which means these two products are sold every day and the selling amount is over twenty pieces. The selling line graphics are shown in figure 4.2 and figure 4.3:

As we can see, there is trend that the selling amount of website B has a relationship with website A. The Pearson correlation coefficient is calculated in table:

From the Pearson correlation coefficient, we can see the relationship between two websites is tracked. However, there are more variables need to be considered. The customers purchasing behavior is affected by multiple reasons. At this stage, we need to include more variables to prove the websites price comparison system.

Product	Pearson correlation coefficient
Aptamil	0.3273
Anchor	0.4216

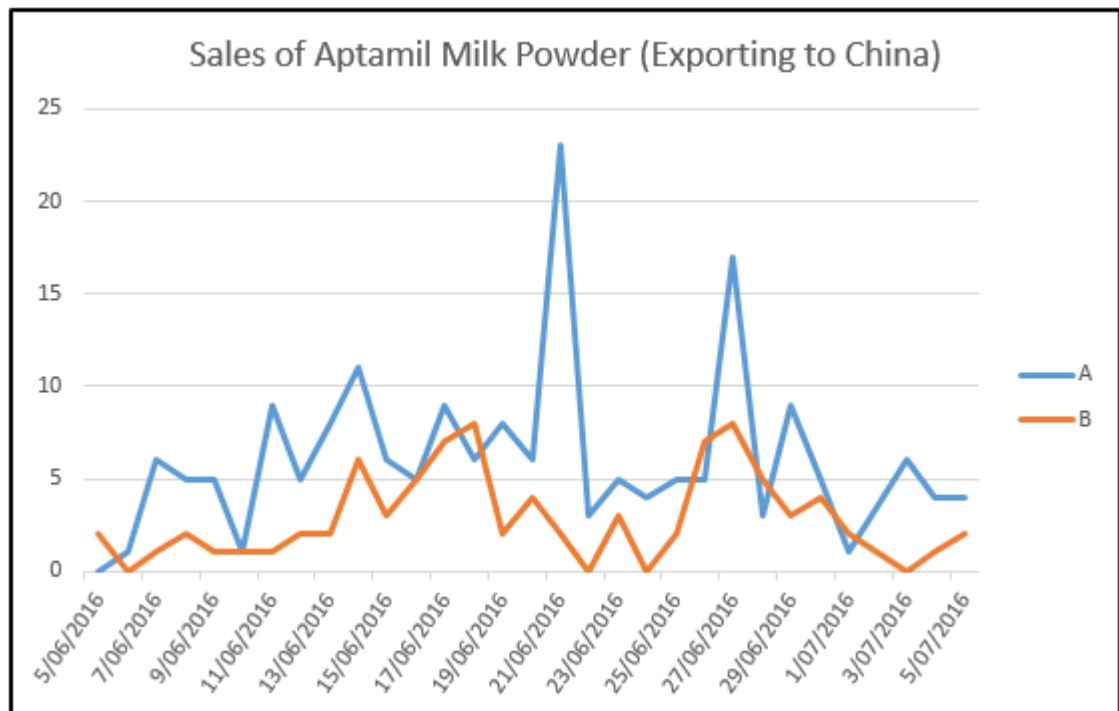


Figure 4.2: Sales of Aptamil

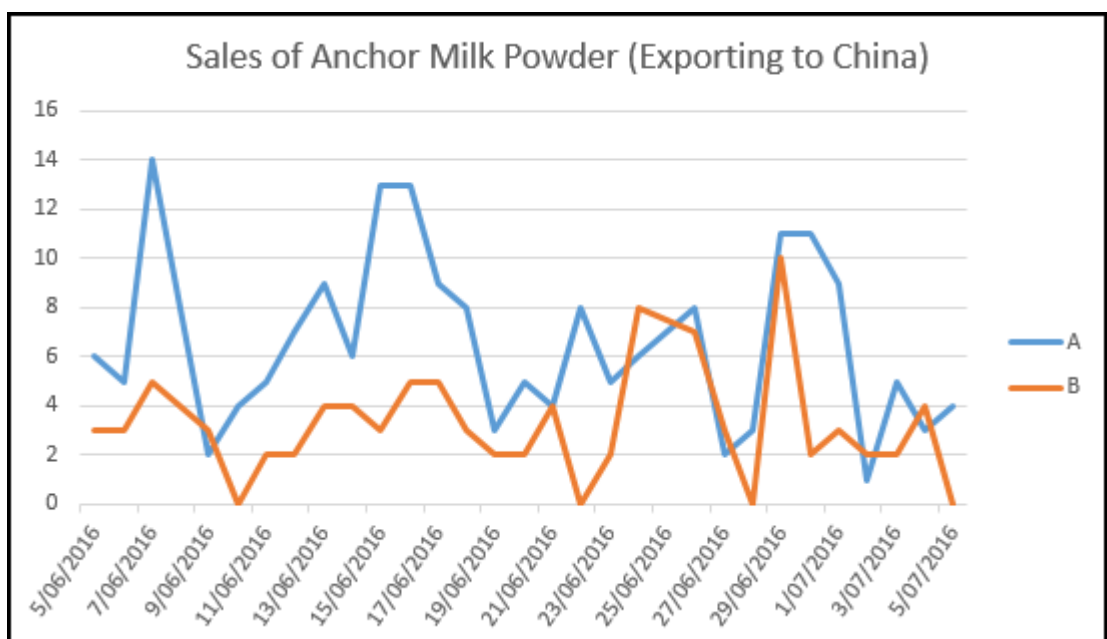


Figure 4.3: Sales of Anchor

Chapter 5

Discussion

5.1 Discussion about the Production Matching

In this thesis, we propose a system of crawling product detail from the internet and make product matching to realize the product price comparison function.

Our system has following features.

1. Extract product accurately

Our web crawler can extract product detail accurately after analyzing the target web site. It is suitable for the user who have clear goal web site.

2. The new key word can be added automatically

In this project, we not only separate the key words, but also add them into a corpus automatically.

3. While searching the similar product, we are able to provide a range of products from the entire database. If there are more than one product in the matching result with the same distance, all of them will be presented.

In this thesis, we use products from both English and Chinese websites located in New Zealand. Since these two languages are very different. First, English words are separated by spaces and punctuations. We can easily separate the words into single tokens. However, Chinese words are hard to segment. We can only separate them into tokens by some common combinations and its meanings. To improve the accuracy of Chinese tokenization, we import some existing dictionaries, which contain a lot of words combinations and groups. Secondly, English words can be various even if they represented the same meaning. Also, the high frequency of words like I, and, is can be disturbing. We use some popular dictionaries of Reuters to remove stop words, to lower and stem the words.

There are still some limitation in this project.

In our test section, due to the calculation consuming, we didnt have enough samples to show the differences when the coefficients changed. To find a larger sample is one of the key steps in the future research. On the other side, more attributes is helpful for analysis, so it is necessary to extract more useful matching item attributes.

Furthermore, our key-word matching method is running not so efficient while dealing with a large database. Those websites with large amount of products (over 3000) need to find a better way to reduce the calculation. Though improve the efficiency of the algorithm may cause the reduction of the accuracy, the balance of performance and accuracy.

5.2 Discussion about the price guidance test

There are follow reason may cause the result of the test:

- A large part of the consumers of these two website are very sensitive about the price. They put the price as the most important aspect when they choose the website to shopping.
- One month is still too short to get rid of the impact of the website normal sales data fluctuation.
- The website self-growth may also cause the result.

Due to the reasons above, the test can be improved in the future. First we can set a long time monitor to avoid the sales volume normal fluctuation. Then we can try to do the same test on more website to confirm the whether the impact is existing.

Bibliography

- [1] Steven Abney. Part-of-speech tagging and partial parsing. In *Corpus-based methods in language and speech processing*, pages 118–136. Springer, 1997.
- [2] Kate Burridge and Tonya N Stebbins. *For the Love of Language: An Introduction to Linguistics*. Cambridge University Press, 2015.
- [3] Soumen Chakrabarti, Martin Van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11): 1623–1640, 1999.
- [4] Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam. Accelerated focused crawling through online relevance feedback. In *Proceedings of the 11th international conference on World Wide Web*, pages 148–159. ACM, 2002.
- [5] Pi-Chuan Chang, Michel Galley, and Christopher D Manning. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*, pages 224–232. Association for Computational Linguistics, 2008.
- [6] Zhumin Chen, Jun Ma, Xiaohui Han, and Dongmei Zhang. An effective relevance prediction algorithm based on hierarchical taxonomy for focused crawling. In *Asia Information Retrieval Symposium*, pages 613–619. Springer, 2008.
- [7] YoungSik Choi, KiJoo Kim, and MunSu Kang. A focused crawling for the web resource discovery using a modified proximal support vector machines. In *International Conference on Computational Science and Its Applications*, pages 186–194. Springer, 2005.
- [8] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.
- [9] Christina Yip Chung, Michael Gertz, and Neel Sundaresan. Reverse engineering for web data: From visual to semantic structures. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 53–63. IEEE, 2002.
- [10] Oxford English Dictionary. Oxford: Oxford university press, 1989.
- [11] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C Lee Giles, Marco Gori, et al. Focused crawling using context graphs. In *VLDB*, pages 527–534, 2000.

- [12] Hai Dong and Farookh Khadeer Hussain. Sof: a semi-supervised ontology-learning-based focused crawler. *Concurrency and Computation: Practice and Experience*, 25(12):1755–1770, 2013.
- [13] Piotr Dziwiński and Danuta Rutkowska. Ant focused crawling algorithm. In *International Conference on Artificial Intelligence and Soft Computing*, pages 1018–1028. Springer, 2008.
- [14] David W Embley, Yuan Jiang, and Y-K Ng. Record-boundary discovery in web documents. In *ACM SIGMOD Record*, volume 28, pages 467–478. ACM, 1999.
- [15] Dayne Freitag. Machine learning for information extraction in informal domains. *Machine learning*, 39(2-3):169–202, 2000.
- [16] Cliff Goddard. *Semantic analysis: A practical introduction*. Oxford University Press, 2011.
- [17] Marti A Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64, 1997.
- [18] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.
- [19] Desmond J Higham. Google pagerank as mean playing time for pinball on the reverse web. *Applied Mathematics Letters*, 18(12):1359–1362, 2005.
- [20] Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Information systems*, 23(8):521–538, 1998.
- [21] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1):15–68, 2000.
- [22] Hongyu Liu, Jeannette Janssen, and Evangelos Milios. Using hmm to learn user browsing patterns for focused web crawling. *Data & Knowledge Engineering*, 59(2):270–291, 2006.
- [23] Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google’s deep web crawl. *Proceedings of the VLDB Endowment*, 1(2):1241–1252, 2008.
- [24] Alexander Maedche, Marc Ehrig, Siegfried Handschuh, Raphael Volz, and Ljiljana Stojanovic. Ontology-focused crawling of documents and relational metadata. In *Proceedings of the Eleventh International World Wide Web Conference WWW-2002*, 2002.
- [25] James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 710, 2000.
- [26] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

- [27] R Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, volume 334, 1999.
- [28] Ion Muslea, Steve Minton, and Craig Knoblock. A hierarchical approach to wrapper induction. In *Proceedings of the third annual conference on Autonomous Agents*, pages 190–197. ACM, 1999.
- [29] Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *EMNLP*, pages 277–284, 2004.
- [30] Constantin Orasan, Viktor Pekar, and Laura Hasler. A comparison of summarisation methods based on term specificity estimation. In *LREC*, 2004.
- [31] Leonard Richardson. Beautiful soup. *Crummy: The Site*, 2013.
- [32] Ellen Riloff, Rosie Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.
- [33] Margaret Rouse. Erp (enterprise resource planning). *TechTarget*, Retrieved July 14, 2015.
- [34] Mariano Rubiolo, María Laura Caliusco, Georgina Stegmayer, Mauricio Coronel, and M Gareli Fabrizi. Knowledge discovery through ontology matching: An approach based on an artificial neural network model. *Information Sciences*, 194:107–119, 2012.
- [35] Srinarayan Sharma, Vijayan Sugumaran, and Balaji Rajagopalan. A framework for creating hybrid-open source software communities. *Information Systems Journal*, 12(1):7–25, 2002.
- [36] Toshiyuki Takahashi, Hong Soonsang, Kenjiro Taura, and Akinori Yonezawa. World wide web crawler. In *Poster proceedings of the 11th international World Wide Web conference*. Citeseer, 2002.
- [37] Thomas Vestskov Terney. The combined usage of ontologies and corpus statistics in information retrieval. *Computer Science Research Report*, (126):1–155, 2010.
- [38] M Thelwall. *A web crawler design for data mining*. Journal of Information Science, 2001.
- [39] Damir Vandic, Jan-Willem Van Dam, and Flavius Frasincar. Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437, 2012.
- [40] Jonathan J Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*, pages 1106–1110. Association for Computational Linguistics, 1992.
- [41] Hai-Tao Zheng, Bo-Yeong Kang, and Hong-Gee Kim. Learnable focused crawling based on ontology. In *Asia Information Retrieval Symposium*, pages 264–275. Springer, 2008.