E-Commerce Merchant Classification using Website Information

Galuh Tunggadewi Sahid
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
galuh.tunggadewi@ui.ac.id

Rahmad Mahendra
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
rahmad.mahendra@cs.ui.ac.id

Indra Budi
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
indra@cs.ui.ac.id

ABSTRACT

With the rapid growth of the e-commerce landscape, classifying e-commerce merchants has become an important task as it is an integral part of various processes in e-commerce. One of the examples is merchant on boarding, where the category of an e-commerce merchant has proven to be a good indicator of the risk of the merchant. However, since most of e-commerce businesses do not have brick-and-mortar stores from which we can assess it directly, the only source of information regarding the merchant itself is its website. Thus, we can view this problem as a web classification problem, where we classify e-commerce websites into a category. In this research, we aim to build an end-to-end classification system for e-commerce websites. There are a few challenges such as the number of pages to be processed, imbalanced dataset, and the language of e-commerce websites that can be mixed language. We built a website classification system and experimented with case study of Indonesian and English e-commerce webs, that are classified into 37 different categories. Our best result achieved an F-score of 0.83.

CCS CONCEPTS

• **Information systems** → *Web searching and information discovery.*

KEYWORDS

Web mining, Classification, Text processing, E-commerce

ACM Reference Format:
Galuh Tunggadewi Sahid, Rahmad Mahendra, and Indra Budi. 2019. E-Commerce Merchant Classification using Website Information. In *9th International Conference on Web Intelligence, Mining and Semantics (WIMS2019), June 26–28, 2019, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3326467.3326486

WIMS2019, June 26–28, 2019, Seoul, Republic of Korea

ACM ISBN 978-1-4503-6190-3/19/06…$15.00
https://doi.org/10.1145/3326467.3326486

1 INTRODUCTION

As the e-commerce landscape continues to grow, e-commerce business models become more varied. Some business models are considered having higher risk than others based on credit card processing data. For instance, the category "Marketplace" can be an indication of high-risk merchants. There are also business models that are considered illegal, such as e-commerce websites that sell drugs. Institutions such as payment gateways are careful in handling merchants that will pose high risks, because merchant account providers are financially liable for all merchant losses. Thus, before using a payment gateway service, all merchants must undergo a thorough merchant onboarding process.

The rapid growth of e-commerce websites means institutions must find a way to automate this process, because otherwise this process will cost so much time and resource. One task that of onboarding process that can be automated is a categorization task, where each merchant is classified into one category according to the goods or services that they provide. The category of a merchant has been shown to be a good indicator of a merchant's risk. One such institution that utilizes website category to determine a merchant's risk is Midtrans, Indonesian biggest payment gateway company. According to a Risk Assessment Report conducted by Midtrans in 2017, the average time to assess one website manually usually takes from 15 minutes to 1 hour. This is because when we want to categorize each merchant manually, we have to inspect the merchant's website one by one and analyze its content to determine its category.

It is important that we have an automatic approach of categorizing merchants with high precision and recall. Precision is important because getting one high-risk merchant can also turn into a loss because payment gateways have to account for the losses should things go wrong. On the other hand, recall is also important because a loss of a potential merchant means a loss for the company.

The challenging part is that online merchants are only represented by their online store. Institutions must be able to determine the category of a merchant when only given the URL to the merchant's online store. Therefore, one can see this problem as a web classification problem. Most of the studies on web classification revolve around general web classification problems instead of domain-specific web classification problems. Web classification in a specific domain, such as e-commerce, might pose specific challenges that are not present in other domains or in a more general case. In this paper, we investigate the following issues:

(1) Abundant number of web pages to be evaluated: processing all pages of an e-commerce website can take a long time considering there can be a page for each product
(2) Imbalanced dataset: some categories have a larger number of websites than others. For example, Fashion websites outnumber many other categories because Fashion merchants vary in size from small businesses to enterprises and they also vary in the kinds of they sell, whereas there are not a lot of Medicine websites because starting a medical business is more complex and thus there are not many of them
(3) Mixed language: e-commerce websites in Indonesia may contain mixed languages of both English and Indonesian within the same website. The phenomenon of code switching of Indonesian-English itself is used in a lot of situations and circumstances [17], especially in informal situations where there are no regulations on having to use proper Indonesian.

These challenges prompt the need for an end-to-end classification system for e-commerce websites, which we propose in this paper.

Many recent developments in the various methods that can be used to build a classification system come from machine learning. Thus, we can also utilize machine learning to approach our web classification problem. However, even after we have narrowed down our approach to machine learning, there are still a lot of different methods, classifiers, ways to choose which pages to be evaluated, and even sets of parameters at our disposal. This means it is important for us to find the best way to classify merchants with a web classification approach to address the issues stated previously.

2 RELATED WORK

Some previous works have approached web classification problems with the same approach as text classification problems. Wang et al. [21] performed web page classification according to the Library of Congress classification scheme using naive Bayes classifier which is also used in this research. Selamat and Omatu [16] performed web classification on news web page and proposed an approach using term-weighted words as inputs to a neural network classifier. In our research, we do use neural network classifier as one of the classifiers used, but we do not give weight to any of the terms. Onan [9] utilized weighted tokenized texts as inputs to ensembled classifiers with base learners consisting of boosting, bagging, dagging, and random subspace. Chen and Hsieh [2] classified sports news into their corresponding sport. They performed additional methods before the classification process like LSA and PCA to reduce the size of the documents that are being processed as well as extract syntactical meaning of the texts. Our research also aims to investigate whether reducing the size of the documents using LSA make a more efficient classification process with a result that is comparable to the result obtained without LSA. Shen [18] took the web pages' texts a step further by performing summarization before classification.

Web pages have their own structures and components that can be used as features during classification. Aside of features related to the text of the web page's content, several previous works have utilized components such as HTML tags and anchor texts, which were shown to improve accuracy when used in addition to content-related features. Sun [19] used a combination of text features, web title, and anchor texts contained in the web pages with a support vector machine classifier. Kwon and Lee [5] devised an approach using k-nearest neighbor to classifying the texts contained in the website with HTML markup tags that were represented using a term-weighting scheme. The pages that were processed were selected using connectivity analysis based on the page's in-degree, out-degree, and the page's distance from the website's homepage. In our research, we also perform page selection in order to see which pages yield the best classification result. However, instead of using connectivity analysis, we divide our dataset into three different datasets based on their hierarchy.

Xiaoguang and Davison [11] made use of the neighbors of the website in a link graph, which are divided into four categories: parents, children, siblings, and spouses. They found that the method proposed was able to improve classification by around 20%. Sibling pages were found to be the most important. Materna [7] aimed to address the problem of short web pages by analyzing the web pages that are referenced by the website. A few works have utilized visual components to perform segmentation of web pages [4]. Another variation of approach to perform web classification task is by utilizing the web page URL only without using its web pages' content or structure [3].

Moiseev [8] performs web classification on e-commerce websites to classify e-commerce websites based on 14 different categories of the products that they're selling. A total of 1448 websites were used for categorization. However, they specify e-commerce as strictly a business-to-customer (B2C) or business-to-business (B2B) e-commerce websites, which is not the case in our research. Our problem scope is wider since it also includes customer-to-customer (C2C) e-commerce websites such as marketplace or classifieds websites as well as customer-to-business (C2B) websites. Moiseev ignored links of different websites, therefore not making use of the external links that reference to the website. They also ignored links with anchor text contains terms such as "delivery" and "contacts" which are terms commonly found in anchor texts in e-commerce websites. Unlike Moiseev, we do not perform these steps in our research to minimize manual human intervention. While Moiseev only used Support Vector Machine (SVM) as the classifier, we experiment with five other supervised models in addition to SVM.

3 DATA AND METHODOLOGY

We see this problem as a web page classification problem, where we assign a web page to a predefined category label.

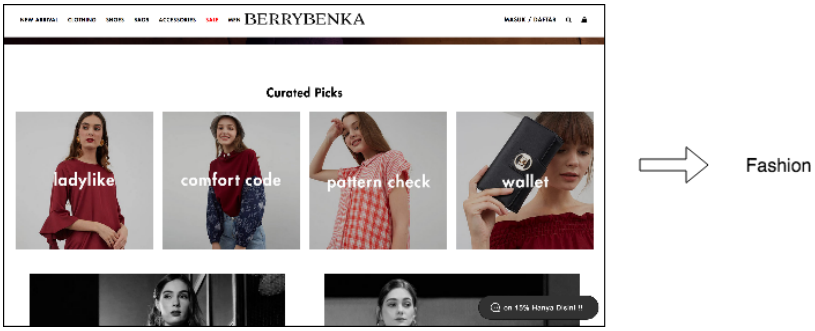

Figure 1: An example of a webpage and its category.

Since our source of data is a set of websites, we first need to gather the unstructured text from the web page that we want to scrap. Next, we determine the recurring patterns behind the information that we are looking for. Finally, we apply these patterns to the unstructured text to extract the information.

3.1 Website Dataset

The data used in our experiment is 866 e-commerce websites that are scrapped from Internet. We strip away all of the HTML tags to obtain the texts contained in each website. The dataset is annotated by Midtrans' fraud analysts. They devised 37 categories as described in Table 1 which they also used in the company. The resulting dataset has a skewed distribution, as the number of websites for each category and each language is not uniform. For example, we have 72 websites that fall into the "Fashion" category, while we only have 23 websites that are in the "Baby/Kids/Mommies" category.

The languages that are used by the websites of e-commerce in Indonesia are not only Indonesian, but also English, and even a mix of both. Our dataset is splitted into those in Indonesian and English. When a website contains both Indonesian and English text, we considered which language is more predominant.

On the other hand, we expect to see which pages are most effective in terms of classifying the website. Therefore, we need to separate the data into three different datasets.

3.1.1 Homepage. This dataset consists of the homepage of each e-commerce web- site. The text data is obtained directly from the homepage.

3.1.2 Homepage + First Level Pages. This dataset consists of the homepage of each e-commerce website as well as its first level pages. A first level page is defined as a lower level web page that immediately follows the web- site's homepage and the slash after that. An example of a first level page is http://example.com/about, because it immediately follows the website's homepage (example.com) and the first slash. The text data is obtained by concatenating the texts from the homepage with all of the first level pages.

3.1.3 All Pages. This dataset consists of all pages in the website, with a maximum limit of the first 100 pages encountered. The text data is obtained by concatenating the texts contained in every page in the website.

3.2 Data Pre-processing

First, we extract the sentences from the HTML files of each website that have been saved in the previous step. The text is obtained by stripping away all HTML tags such and scripts, thus leaving only the text. However, it is worth nothing that since the web is semi-structured, the texts obtained might not be in a perfect form like the ones found in a formal document.

After extracting the sentences, we remove numbers and words which length is less or equal to two characters since those words are either stop words or gibberish characters and thus do not give any meaning. This is also done to speed up the training process, since the scraping process is not perfect and might include random characters due to the semi-structured nature of the web.

At this point, we create two different datasets: one dataset undergoes further text pre-processing and another one without further text-pre-processing. The data that does not undergo text pre-processing goes straight to the tokenization step after they are stripped from numbers and gibberish characters. Meanwhile, the text pre-processing consists of several steps, that are stop words removal, stemming, and POS tagging. We utilize a number of language resources and tools, including Indonesian stop words [20], English stop words [15], Indonesian POS tagger [13], Indonesian stemmer [1], and English stemmer [10].

Stop words are highly frequent used words in the text. Usually those do not give much meaningful information. By removing stop words, we can decrease the time needed to both train our data and predict new data because there are less features to be considered.

The process of POS tagging labels each word with a Part-Of-Speech (POS) category, such as noun, verb, and adjective. There are two main objectives of applying POS tagging as a part of our system. First, POS tagging output determines which stemmer is appropriately used for particular word. We assume naively that all words in the text are Indonesian, so

Table 1: The statistics of the dataset used based on the category and language

31cm**Category**	**Language**	
	English	**Indonesian**
Adult Content	3	6
Airlines	11	2
Animal Breeder	21	12
Automotive & Parts	2	8
Baby/Kids/Mommies	9	12
Beauty Products	12	19
Booking Engine	12	19
Books/Magazines	10	15
Cash Withdrawal	0	16
Cloud/Web Hostings	15	24
Courier Service	5	12
Dating & Escort Services	2	13
Donation/Charity	13	15
Education	21	18
Electronics	8	18
Events/Tickets	15	19
Fashion	46	26
Financial Service	3	14
Flowers	21	18
Food & Beverages	16	23
Furniture	18	6
Gambling	6	19
Herbal Medicine	11	10
Insurance	11	14
Marijuana	3	3
Marketplace	4	7
Medical Service	2	11
Online Games	3	14
Payment Service	4	15
Property Sale/Rent	14	12
Property Sale/Rent Ads	5	3
Service Platform	3	9
Social Media	8	17
Telecommunication	8	4
Tobacco & Vape	7	11
Tour Package	12	11
Weapons	4	12

that we proceed them with Indonesian POS Tagger. When a word is actually English, POS Tagger is expected to tag that word as "Foreign Word" label. Second, POS information help us to identify terminologies, which are oftentimes multi tokens. In this case, we filter sequence of nouns and treat them as single word feature.

Word representation of data (both data that have undergone text pre-processing and one that have not) is transformed into a document-term matrix, either using a binary representation, TF, or TF-IDF.

3.3 Feature Engineering

We perform feature engineering where each feature engineering method is combined with different datasets and data pre-processing method.

3.3.1 Latent Semantic Analysis (LSA). We apply LSA to term-document matrices which use term-frequency (TF) scheme or term frequency- inverse document frequency (TF-IDF) scheme. LSA is also performed on data of which both with and without text pre-processing are applied. We want to see if having a more compact feature space with a richer set of features yield better result or not. We configure the number of components to be 100, which is a recommended value for LSA.

3.3.2 Word Embedding. We trained our own word embedding model using the e-commerce websites dataset that we have created, because we are performing a research on a more specific domain, which is e-commerce. After obtaining vectors for each word, we clustered all of the vectors into 100 clusters using k-means clustering. Each tokenized word that is obtained from the website is represented by its cluster. Instead of using either TF or TF-IDF, we create a binary term-document incidence matrix to represent the document. Thus, if a word from a cluster exists, the value of the cell in the matrix becomes 1.

3.3.3 Keyword Extraction. The keywords of each website are extracted by using the Rapid Automatic Keyword Extraction (RAKE) algorithm which is already implemented in Python[1]. RAKE accepts documents of text that consist of sentences as the input, so each website is represented by all the sentences that are contained in it. We select the top 50 keywords with the highest value. If the length of the keyword is equal to two words, we transform it into unigram components. Likewise, if the length of the keyword is equal to more than two, we transform it into both unigram and bigram components.

4 EXPERIMENT

We have conducted extensive experiments with the following objectives:

(1) to determine which pages (Homepage, Homepage + First Level Pages, or All Pages) should be used for classification
(2) to determine whether pre-processing (stemming, stopwords removal, and POS tagging) gives better result
(3) to determine the scheme that is best to represent the features (TF or TF-IDF)
(4) to determine whether further feature engineering (LSA, Keyword Extraction, or Word Embedding) is needed and if so, which one gives the best result
(5) to determine the classifier that gives the best result

[1] https://github.com/aneesha/RAKE

4.1 Experimental Setup

For each dataset, we perform seven different scenarios to find the best configuration and classifiers for e-commerce website classification using its content. These seven different scenarios are:

(1) Term Frequency (TF)
(2) Term Frequency-Inverse Document Frequency (TF-IDF)
(3) Term Frequency + Latent Semantic Analysis (TF LSA)
(4) Term Frequency-Inverse Document Frequency + Latent Semantic Analysis (TF-IDF LSA)
(5) Word Embedding (WE)
(6) Keywords Frequency (KF)
(7) Keywords Frequency-Inverse Document Frequency (KF-IDF)

We also compare the performance of models that do not use text pre-processing and models that use text pre-processing. For scenario TF, TF-IDF, TF LSA, and TF-IDF LSA, we also compare the results between experiments that do not undergo text pre-processing steps (Wo-TP) and experiments with text pre-processing steps (W-TP) applied. The flow of all experiment scenarios are illustrated in Figure 2.

To evaluate the performance of the models, each dataset is partitioned into five subsets, where one subset is then be used as a validation set and the rest of the k-1 sets are used as training data.

4.2 Model Comparison

We perform experiments using six different machine learning algorithms for classification: decision tree, k-nearest neighbor (k-NN), naive Bayes, support vector machine (SVM), logistic regression, and multilayer perceptron (MLP). For all machine learning algorithms, we use the implementation provided by Python's scikit-learn[2] library.

Each of the classification algorithm is then combined with seven different schemes: TF, TF-IDF, TF with LSA, TF-IDF with LSA, word embedding, TF with keywords, and TF-IDF with keywords. For TF, TF-IDF, TF with LSA, and TF-IDF with LSA, we perform experiments on both data that with text pre-processing applied and data without text pre-processing applied. Meanwhile, for word embedding, TF with keywords, and TF-IDF with keywords, we only perform experiments on the data without pre-processing applied.

4.2.1 Decision Tree. We use the Gini impurity as a measure of the split quality. In each node, we choose the best split instead of choosing the best random split. The depth of the tree is not limited so all nodes are expanded until all leaves have become pure leaves or until all leaves contain less than the minimum number of samples needed to split an internal node, which is set to 2.

4.2.2 k-Nearest Neighbor (k-NN). Since the number of neighbors used is configurable, we set the number of neighbors used to be 5. All points in each neighborhood are set to have equal weights.

[2]https://scikit-learn.org

4.2.3 Naive Bayes Classifier. In this research, we use multinomial naive Bayes classifier. We also use Laplacian smoothing to handle the edge case where none of the words that appear in our training data also appear in our testing data. The naive Bayes classifier is used for all datasets except those which uses LSA as its feature engineering method, because naive Bayes cannot take negative values that were produced during the LSA process.

4.2.4 Support Vector Machine (SVM). We set the penalty to use L2 regularization so that high-value weights are penalized to avoid overfitting. The loss function used is hinge loss, which is a common loss function used for SVM [14]. The multi-class strategy that is used is one-versus-rest, which is the most commonly used strategy for multiclass classification. In one-versus-rest, a single classifier is trained per class using the samples that belong to that class as positive samples while using the rest of the samples as negative samples. We choose the class which classifier yields the highest probability estimate.

4.2.5 Multilayer Perception. Following previous work on multilayer perceptron for text classification task [6], we use a multilayer perceptron architecture with two hidden layers, where the first hidden layer has 1024 neurons and the second hidden layer has 512 neurons. The size of the input depends on the dictionary generated during training, while the size of the output layer is equal to the number of categories. Meanwhile, the activation function for the hidden layers that is used is rectified linear unit function (ReLu), which returns $f(x) = max(0, x)$. ReLu has been found to yield a better accuracy than other activation functions such as sigmoid and tanh in text classification tasks [12]. The solver for weight optimization used is adam which is a stochastic gradient-based optimizer.

4.2.6 Logistic Regression. We use L2 regularization as the penalty function. Like SVM, we also use one-versus-rest as the strategy for multi-class classification.

4.3 Evaluation

We perform the evaluation using stratified k-fold cross-validation, with k = 5. We implement stratification to ensure that each class is approximately represented equally across each fold.

In order to choose which dataset and configuration yield the most best results, we measure the performance of our model using F-score. The best model is the one with the highest classification F-score with macro-averaging. We choose F-score with macro-averaging because it treats all classes equally, while in contrast micro-averaging F-score favors bigger classes. Since we have an imbalanced dataset where some categories are more populated than other categories, models that are not biased towards more populated categories are more preferable.

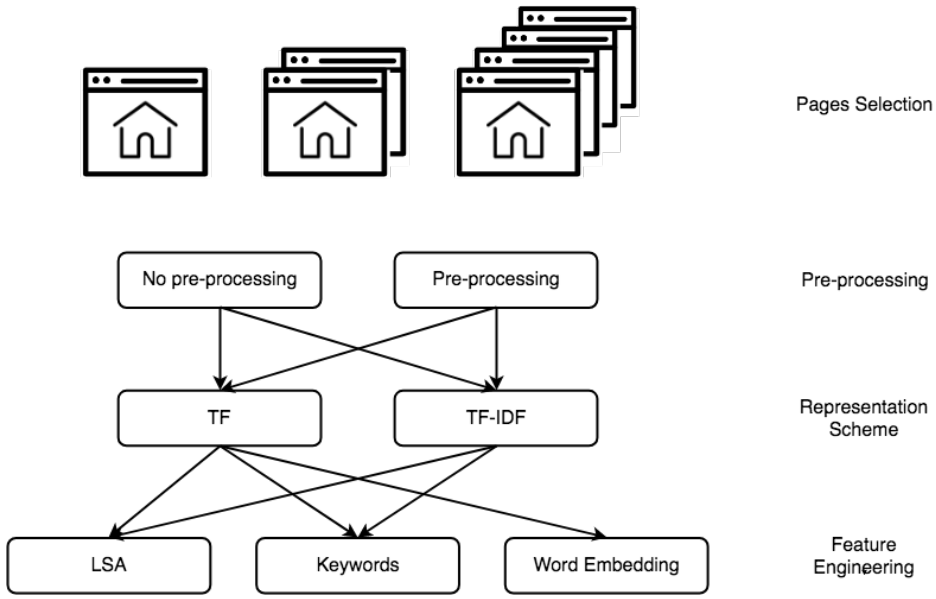

Figure 2: Flow of all experiment scenarios.

Pages	Classifier	TF		TF-IDF		TF LSA		TF-IDF LSA		KF	KF-IDF	WE
		Wo-TP	W-TP	Wo-TP	W-TP	Wo-TP	W-TP	Wo-TP	W-TP	Wo-TP	Wo-TP	Wo-TP
Homepage	Decision Tree	0.46	0.47	0.51	0.51	0.24	0.30	0.44	0.40	0.50	0.52	0.27
	k-NN	0.31	0.36	0.61	0.66	0.30	0.40	0.52	0.49	0.56	0.66	0.22
	Naive Bayes	0.59	0.69	0.25	0.33	-	-	-	-	0.67	0.25	0.31
	SVM	0.57	0.62	0.78	0.79	0.34	0.56	0.71	0.71	0.56	0.49	0.50
	Multilayer Perceptron	0.56	0.62	0.78	0.78	0.51	0.61	0.73	0.72	0.56	0.49	0.51
	Logistic Regression	0.63	0.71	0.52	0.53	0.56	0.62	0.50	0.49	0.69	0.44	0.51
Homepage	Decision Tree	0.45	0.48	0.49	0.51	0.22	0.27	0.41	0.43	0.49	0.52	0.27
+ First Level	k-NN	0.25	0.41	0.59	0.69	0.24	0.39	0.52	0.52	0.61	0.67	0.24
Pages	Naive Bayes	0.57	0.62	0.20	0.30	-	-	-	-	0.63	0.25	0.29
	SVM	0.48	0.54	**0.83**	0.82	0.50	0.53	0.70	0.73	0.56	0.79	0.61
	Multilayer Perceptron	0.58	0.62	0.52	0.77	0.56	0.59	0.47	0.74	0.58	0.51	0.59
	Logistic Regression	0.59	0.67	0.77	0.53	0.53	0.61	0.72	0.49	0.66	0.45	0.62
All Pages	Decision Tree	0.47	0.48	0.32	0.69	0.22	0.27	0.43	0.44	0.45	0.49	0.27
	k-NN	0.29	0.48	0.32	0.69	0.27	0.39	0.51	0.52	0.61	0.67	0.60
	Naive Bayes	0.70	0.73	0.21	0.30	-	-	-	-	0.74	0.24	0.27
	SVM	0.44	0.53	0.80	0.80	0.44	0.53	0.72	0.73	0.48	0.79	0.59
	Multilayer Perceptron	0.55	0.65	0.53	0.78	0.54	0.60	0.49	0.76	0.62	0.54	0.59
	Logistic Regression	0.61	0.65	0.73	0.53	0.54	0.61	0.71	0.49	0.62	0.46	0.60

Table 2: Macro-average F-score for all scenarios and datasets.

5 RESULTS

5.1 Experimental Results

In this section, we present the results and analyses of our experiments with seven different scenarios that are performed on the Homepage, Homepage + First Level Pages, and All Pages dataset. The full result of our experiments is presented in Table 2.

From the table, we can see SVM, MLP, and Logistic Regression tend to perform better than others. SVM performs well when the TF-IDF scheme is used, while logistic regression performs well when the TF scheme is used. MLP also gives a comparable result, but it is only able to achieve such result with thousands of neurons in the hidden layers, taking a longer training time. From this perspective, SVM is better since its training time is much faster.

From the dataset, it can be observed that adding more pages to the dataset do not necessarily mean obtaining better results. In the case of classifying e-commerce websites, the Homepage + First Level Pages dataset outperforms other datasets. This shows that homepage and first level pages are the most optimal combination of pages. Despite that, it is interesting to note that the Homepage dataset also performs quite well, although not as well as the Homepage + First Level Pages dataset. This shows that the homepage of an

e-commerce website alone is already a good representative of what an e-commerce website is about and what kind of products or services that it offers, and the first level pages do enhance the information. This also means that when training time is a concern, using only the homepage gives an advantage of faster training time due to the small size of data that has to be processed, yet still manages to give a result that is comparable to when both the homepage and first level pages are used. The small impact of the All Pages dataset in terms of the model's performance shows that pages other than the homepage and first level pages are neither noises nor pages that contain information that have not been presented before. Thus, when classifying, it is reasonable not to use the entire pages of the website because the results are comparable.

We find that TF-IDF dominates as the scheme with the best performance. Seeing that TF does not perform as well, we suspect there are a lot of words which inverse document frequency we also need to take into account. These words could be words such as *cart* or *sell* which do appear in a lot of e-commerce websites regardless of what their category is. We also find that neither LSA nor word embedding are on the table, because they did not perform well. This is consistent with the speculation that every feature in the texts does contain important information, thus reducing the dimensions with LSA or word embedding will result in a great loss of information and decrease the performance of the model.

We also learn that pre-processing also does not give significant impacts in terms of improving the model which already have achieved good results. Models that use SVM or MLP as the classifier already perform good enough or even better when text pre-processing is not applied. However, pre-processing does improve the performance when the TF scheme is used and LSA is applied. Therefore, we can conclude that the choice of whether pre-processing should be used or not depends on the classifier. We conjecture that SVM and MLP are good enough for text classification problems, so they no longer need text pre-processing steps because they are able to handle the issues themselves. However, other classifiers that may not be that suitable for text classification in the first place did find text pre-processing useful to some extent, although still not as good as SVM or MLP.

5.2 Further Analysis on the Best Model

From all the experiments conducted, SVM classifier that uses the TF-IDF scheme without pre-processing outperforms all other configurations and classifiers using the Homepage + First Level Pages dataset. This model manages to achieve an F-score of 0.83.

To further assess the performance of this particular model, we inspect the precision, recall, and macro-average F-score for each category which is presented in Table 3.

Based on the results on Table 3, we observe that the top five categories with the highest F-score values are "Marijuana", "Cash Withdrawal", "Flowers", "Herbal Medicine", and "Cloud/Web Hosting". All e-commerce websites that belong to the category "Marijuana" are correctly classified and no websites that belong to other category are incorrectly classified as "Marijuana". This shows that "Marijuana" contains features that are distinct enough from other categories so that our model is able to classify them perfectly. These features could be in the form of words that frequently appear in either category but rarely appear in other categories. An interesting finding is that "Marijuana" has a relatively small data compared to other categories, with only 6 websites included in the dataset due to difficulty in finding websites that sell marijuana. However, the model is able to perfectly correctly classify websites that belong to the "Marijuana" category.

The next category with the highest F-score is "Cash Withdrawal". Based on our result, there is one website that is incorrectly classified as belonging to the "Tobacco & Vape" category. However, no other websites are misclassified into the "Cash Withdrawal" category. It should be noted that Cash Withdrawal also has a relatively small data with only 16 websites.

The third highest F-score is "Flowers", with an F-score of 0.96. There are two websites that are misclassified into other categories, which are "Animal Breeder" and "Furniture" respectively. Lastly, both "Cloud/Web Hosting" and "Herbal Medicine" have an F-score of 0.95. The recall of "Cloud/Web Hosting" is 1, meaning that all websites that belong to the "Cloud/Web Hosting" category are all correctly classified. Meanwhile, "Herbal Medicine" has a value of 0.95 for both precision and recall.

Besides of the aforementioned categories, another category with a recall value of 1 is "Dating & Escort Services". It is possible that websites which category has a recall value of 1, which are "Herbal Medicine", "Marijuana", "Cloud/Web Hosting", and "Dating & Escort Services" have distinct features from other categories so that the model is able to tell them apart. Meanwhile, "Weapons" has a precision value of 1 which means that no other website is incorrectly classified as "Weapons".

The top five categories with the lowest F-score are "Marketplace", "Medical Service", "Service Platform", "Tobacco & Vape", and "Baby/Kids/Mommies". "Marketplace" has a really low recall, which is 0.18. The model only manages to classify two "Marketplace" websites correctly. We suspect that since "Marketplace" websites do provide a wide range of items, some of these items may also exist in other categories. For example, a Marketplace website might sell beauty products as well as clothes, thus the possibility of it being misclassified into either Beauty Products or Fashion. This shows that the model is still not able to handle such cases and may need other features that are able to help the model to tell the difference between websites that belong to the "Marketplace" category and websites that do not. We find an example in our data in which a website which belongs to the "Marketplace" category also sells tops and dresses. This possibly leads the model to incorrectly classify it as a "Fashion" website.

"Medical Service" also has a recall value of 0.38. We suspect this is due to a wide range of websites that are categorized as "Medical Service". An example is a website that specifically offers vaccination services, so the website only contains terms that are very specific to vaccine. We also find another website, which is an aesthetic clinic, that only contains terms related to beauty treatments, aesthetic surgery, or dermatology treatments. To improve the performance of the model, categories with this characteristic may need a larger dataset so the model can learn specific terms from more data. We also find a website that contains *lorem ipsum* in some parts, which could be an issue. This also shows that aside of handling non-operational websites, the classification system also needs to handle cases like this.

The next category, "Service Platform", does have a relatively high precision value which is 0.83. However, its recall is rather low, which is 0.42. All websites in English which belong to the "Service Platform" category are all misclassified into other categories. Considering that the "Service Platform" data is rather imbalanced, with only 3 websites in English and 9 websites in Indonesian, the model may have difficulty to classify Indonesian websites because it may lack Indonesian websites in the training data.

For "Tobacco & Vape", it is found that some websites are no longer working correctly, thus leading to misclassified websites. Since in practice the process of web scraping is not always perfect, this shows that the system also needs to handle cases where the websites are no longer operating or are under maintenance.

However, some other websites that are still working well are misclassified into "Fashion". It is a small chance though that websites that belong to the "Fashion" category also sell tobacco and vape, thus there is probably another explanation for the misclassification. We suspect that whenever the model fails to find enough features that are distinctive towards a particular category, the model is biased towards "Fashion", which is one of the most represented category in the dataset.

"Baby/Kids/Mommies" websites have a low recall of 0.57. In this case, we suspect that the reason is the same as "Marketplace": there are products sold in "Baby/Kids/Mommies" that are also sold in other websites from other categories. Six "Baby/Kids/Mommies" websites are incorrectly classified as "Fashion" websites. This is possible since many "Fashion" websites also happen to sell products for baby, kids, and mommies, although not exclusively for them only. This issue is the same issue as the one encountered in the "Marketplace" category, where the model is still unable to handle cases where products sold or offered in a particular category also happen to be sold or offered in another category.

An interesting point is that categories with samples less than 10 do not seem to be affected by the fact that they have a relatively small size of data. "Automotive & Parts", "Adult Content", "Property Sale/Rent", and "Marijuana" obtains an F-score value of 0.71, 0.82, 0.88, and 1 respectively. This shows that in some cases, small datasets are not a big issue for our model, provided that each category is quite distinctive of each other.

Although there are some differences between our research and the previous work done by Moiseev [8] in the e-commerce domain, we find that our best model slightly performs better than to the best model obtained by Moiseev, which achieves an F-score of 0.81. Meanwhile, our best model reaches an F-score of 0.83. Moiseev's best model uses SVM classifier with TF-IDF scheme that uses tag weighting. This configuration is similar to our best model, which also uses SVM and TF-IDF to represent each website although without the tag weighting scheme.

The main difference between our best model and Moiseev's, aside from the tag weighting scheme, is the selection of pages. Both of our researches agree that using only the homepage of the website does not give the classifier enough information. However, our research finds that using the homepage and the first level pages only yield a better result than using all pages like Moiseev did. This difference is a possible explanation to the question why our research performs slightly better than Moiseev's best model.

It should also be noted that our best model solves a more general problem–our model is trained on 37 different categories, which is more than half of the categories that Moiseev trained their model on. Our model also works to classify websites in both Indonesian and English, as opposed to Moiseev's model which only works for a language (Russian).

6 CONCLUSION

In this paper, we built an end-to-end classification system for e-commerce websites. We collected and labeled e-commerce websites to be used as training and testing data with one of the 37 pre-defined categories. With an F-score of 0.83, we conclude that the best classification model is the one with SVM classifier and TF-IDF scheme that only processes the homepage and first level pages, with no pre-processing and further feature engineering needed.

Regarding the challenges faced by classifying e-commerce websites, we find that we do not need to process all pages contained in a website to obtain a good result, as the first level pages already have enough information for our classifier. This finding is important because one may naturally think to process all pages, which takes a longer time but does not yield a better result. To address the second challenge, which is imbalanced dataset, we find that our classification system can handle imbalanced dataset to some extent. Based on the analysis of the best model, our model is able to correctly classify Marijuana websites despite the relatively small training data (six websites) compared to other categories. However, our classifier falls short when classifying categories such as Medical Service, which contains websites that are too diverse within the category itself, thus not providing features that are distinctive enough for our classifier. Lastly, to address the third challenge which is mixed language, we find that pre-processing efforts such as POS tagging did improve models that use certain classifiers such as k-NN, but does

Table 3: Precision, recall, and F-score of each category using SVM classifier with TF-IDF scheme and without pre-processing

Category	Precision	Recall	F-score
Adult Content	0.88	0.78	0.82
Airlines	0.80	0.62	0.7
Animal Breeder	0.74	0.97	0.84
Automotive & Parts	0.86	0.60	0.71
Baby/Kids/Mommies	0.80	0.57	0.67
Beauty Products	0.93	0.90	0.92
Booking Engine	0.84	0.54	0.64
Books/Magazines	0.95	0.80	0.87
Cash Withdrawal	1.00	0.94	0.97
Cloud/Web Hosting	0.91	1.00	0.95
Courier Service	0.74	0.82	0.78
Dating & Escort Services	0.83	1.00	0.91
Donation/Charity	0.87	0.90	0.88
Education	0.81	0.90	0.85
Electronics	0.87	0.77	0.82
Events/Tickets	0.81	0.88	0.85
Fashion	0.71	0.93	0.80
Financial Services	0.75	0.71	0.73
Flowers	0.97	0.95	0.96
Food & Beverages	0.87	0.87	0.87
Furniture	0.97	0.82	0.89
Gambling	0.96	0.92	0.94
Herbal Medicine	0.95	0.95	0.95
Insurance	0.86	0.96	0.91
Marijuana	1.00	1.00	1.00
Marketplace	0.50	0.18	0.27
Medical Services	0.83	0.38	0.53
Online Games	0.93	0.76	0.84
Payment Services	0.81	0.89	0.85
Property Sale/Rent	0.88	0.88	0.88
Property Sale/Rent Ads	0.88	0.88	0.88
Service Platform	0.83	0.42	0.56
Social Media	0.86	0.96	0.91
Telecommunication	0.80	0.67	0.73
Tobacco & Vape	0.69	0.61	0.65
Tour Package	0.86	0.83	0.84
Weapons	1.00	0.88	0.93
Average	0.86	0.81	0.83

not give a better result for models that use SVM and MLP as their classifier.

Future work may attempt to perform a classification using non-textual features such as images, HTML tags, or hyperlinks. One may also use the information from the meta tags of each website. Secondly, one can try to see this task as a semi-supervised or an unsupervised problem since the e-commerce landscape is developing rapidly and it will be difficult to depend on labeled categories. Thirdly, one may also see this as a multi-labeling problem since from our research, it is found that a merchant may also sell products or offer goods that also exist in a merchant from another category. Meanwhile, as a general suggestion, future work is encouraged to use a standardized vocabulary to describe aspects that are specifically related to e-commerce, such as the categories. One such standardized vocabulary is provided by GoodRelations[3], an ontology to annotate various aspects of e-commerce on the Web.

7 ACKNOWLEDGEMENT

We gratefully thank Universitas Indonesia for the International Publication Grants (Hibah PIT-9) Year of 2019.

REFERENCES

[1] Mirna Adriani, Jelita Asian, Bobby Nazief, Seyed MM Tahaghoghi, and Hugh E Williams. 2007. Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing (TALIP)* 6, 4 (2007), 1–33.

[2] Rung-Ching Chen and Chung-Hsun Hsieh. 2006. Web page classification based on a support vector machine using a weighted vote schema. *Expert Systems with Applications* 31, 2 (2006), 427–435.

[3] Min-Yen Kan. 2004. Web page classification without the web page. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*. ACM, 262–263.

[4] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, and Veljko Milutinovic. 2004. Visual adjacency multigraphs-a novel approach for a web page classification. In *Proceedings of SAWM04 Workshop, ECML2004*.

[5] Oh-Woog Kwon and Jong-Hyeok Lee. 2003. Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing & Management* 39, 1 (2003), 25–44.

[6] Ladislav Lenc and Pavel Král. 2017. Deep neural networks for Czech multi-label document classification. *arXiv preprint arXiv:1701.03849* (2017).

[7] Jirí Materna. 2008. Automatic web page classification. *RASLAN* 8 (2008), 84–93.

[8] George Moiseev. 2016. Classification of E-commerce Websites by Product Categories.. In *AIST (Supplement)*. 237–247.

[9] Aytuğ Onan. 2016. Classifier and feature set ensembles for web page classification. *Journal of Information Science* 42, 2 (2016), 150–165.

[10] Martin F Porter. 2001. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/texts/introduction.html

[11] Xiaoguang Qi and Brian D Davison. 2006. Knowing a web page by the company it keeps. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. ACM, 228–237.

[12] Shuang Qiu, Ming-yang Jiang, Zhi-li Pei, Yi-nan Lu, et al. 2017. Text Classification Based on ReLU Activation Function of SAE Algorithm. In *International Symposium on Neural Networks*. Springer, 44–50.

[13] Fam Rashel, Andry Luthfi, Arawinda Dinakaramani, and Ruli Manurung. 2014. Building an Indonesian rule-based part-of-speech tagger. In *Asian Language Processing (IALP), 2014 International Conference on*. IEEE, 70–73.

[14] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. 2004. Are loss functions all the same? *Neural Computation* 16, 5 (2004), 1063–1076.

[15] Gerard Salton. 1971. *The SMART retrieval system–experiments in automatic document processing*. Prentice-Hall, Inc.

[16] Ali Selamat and Sigeru Omatu. 2004. Web page feature selection and classification using neural networks. *Information Sciences* 158 (2004), 69–88.

[17] Dedy Setiawan. 2016. English Code Switching in Indonesian Language. *Universal Journal of Educational Research* 4, 7 (2016), 1545–1552.

[3] http://purl.org/goodrelations/

[18] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, and Wei-Ying Ma. 2004. Web-page classification through summarization. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 242–249.

[19] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. 2002. Web classification using support vector machine. In *Proceedings of the 4th International Workshop on Web Information and Data Management*. ACM, 96–99.

[20] Fadillah Z Tala. 2003. A study of stemming effects on information retrieval in Bahasa Indonesia. *Institute for Logic, Language and Computation, Universiteit van Amsterdam, The Netherlands* (2003).

[21] Yong Wang, Julia Hodges, and Bo Tang. 2003. Classification of web documents using a naive bayes method. In *Proceedings for the 15th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 560–564.