

Utilizing Text Mining and Feature-Sentiment-Pairs to Support Data-Driven Design Automation Massive Open Online Course

<https://doi.org/10.3991/ijet.v16i01.17095>

Nasa Zata Dina^(✉), Riky Tri Yunardi, Aji Akbar Firdaus
Universitas Airlangga, Surabaya, Indonesia
nasazatadina@vokasi.unair.ac.id

Abstract—This study aimed to develop a case-based design framework to analyse online user reviews and understanding the user preferences in a Massive Open Online Course (MOOC) content-related design. Another purpose was to identify the future trends of MOOC content-related design. Thus, it was an effort to achieve data-driven design automation. This research extracts pairs of keywords which are later called Feature-Sentiment-Pairs (FSPs) using text mining to identify user preferences. Then the user preferences were used as features of an MOOC content-related design. An MOOC case study is used to implement the proposed framework. The online reviews are collected from www.coursera.org as the MOOC case study. The framework aims to use these large-scale online review data as qualitative data and converts them into quantitative meaningful information, especially on content-related design so that the MOOC designer can decide better content based on the data. The framework combines the online reviews, text mining, and data analytics to reveal new information about users' preference of MOOC content-related design. This study has applied text mining and specifically utilizes FSPs to identify user preferences in the MOOC content-related design. This framework can avoid the unwanted features on the MOOC content-related design and also speed up the identification of user preference.

Keywords—Feature-sentiment-pairs, massive open online course, online review, sentiment analysis

1 Introduction

Massive Open Online Course, often abbreviated to MOOC, is online courses held by a university with the option of free or open registration. The teachers are faculty members of the university or general practitioner in their fields. Classes are conducted by weekly lectures using videos, online assessments, and discussion forums. However, some educators assume that the MOOC's quality of learning is different from face-to-face class because it cannot replace face-to-face classroom engagement, laboratory, fieldwork, and any other aspect [1][2][3].

The purpose of Data-Driven Design is to design a system based on the provided data. The pattern of the available data is discovered using data mining algorithms [4]. The result will be used as a new design system. According to [5], the challenges of Data-Driven Design usage as a product evaluation process are data structure, data understanding, incomplete information, nature of the data, and individual cognitive limitation. The data are often unstructured and heterogeneous; thus, it is mandatory to process the data into information. Hence, the main challenge of data-driven design is to transform the large-scale unstructured data into meaningful information so that people are able to utilize the knowledge.

Text mining in the context of this study is to transform unstructured text into vectors of numbers. It can be said that the unstructured text refers to the qualitative data, and the vectors of numbers are the quantitative data. Later, those vectors will be used as input for a process on the framework. The process will involve a data mining algorithm to predict classification results [6]. Multidisciplinary fields have applied text mining to help their business process, for example, knowledge management, customer care service, social media data analysis, and spam filtering. In 2020, [7] applied text mining in the tourism sector which found out the most frequent words that described the negative and positive experience while staying in a particular hotel. It helped the hotel's management to improve its customer care services. Meanwhile, [8] in 2019 proposed a novel approach using semantic-based Naive Bayesian classifier in text analytics to filter spam emails. Text mining still has the potential to be explored in further studies and can be applied directly to solve practical problems in many fields.

This study aimed to help the designer of the MOOC learning platform to understand the users' needs without having to browse the online reviews every time. The users here are the students who use MOOC as a learning platform. The MOOC learning platform we used was Coursera. Then, the users' online reviews were extracted from Coursera in the review section. Then, a design automation system was built to analyze the users' online reviews and obtain features that appear most often on the online review data. Text mining was used to acquire keyword then it was paired as Feature-Sentiment-Pairs (FSPs). Finally, the result of the system was referred to machine model and it was compared to human-generated result that was called the human model. The purpose of the comparison was to find out the machine model's performance. The human model was used as a control variable.

The organization of the article is written as follows: Section 2 explains the method used of the proposed framework. Section 3 represents the result and it is followed by Section 4 that is the discussion. Finally, the conclusion of this research is written in Section 5.

2 Methods

The proposed framework is illustrated in Fig. 1. The framework aims to find out the difference between the machine model and the human model. There are four steps to generate FSP automatically.

2.1 Collecting the data

The data were collected using Webharvy Web Scrapper [9] from the Coursera website. The example of the review page of Coursera can be seen in Fig. 2. Webharvy needs the URL page of each course’s online review where the original dataset in. The users determine the attribute that would be crawled. In this research, we crawled “online review”, “course’s name” and “star review”. Table 1 gives a sample of the crawled data. Besides these attributes, there were other attributes given on the page such as “username” and “date”. However, we excluded them since we did not categorize the online reviews based on those attributes. Fig. 3 displays the interface of Webharvy Web Scrapper.

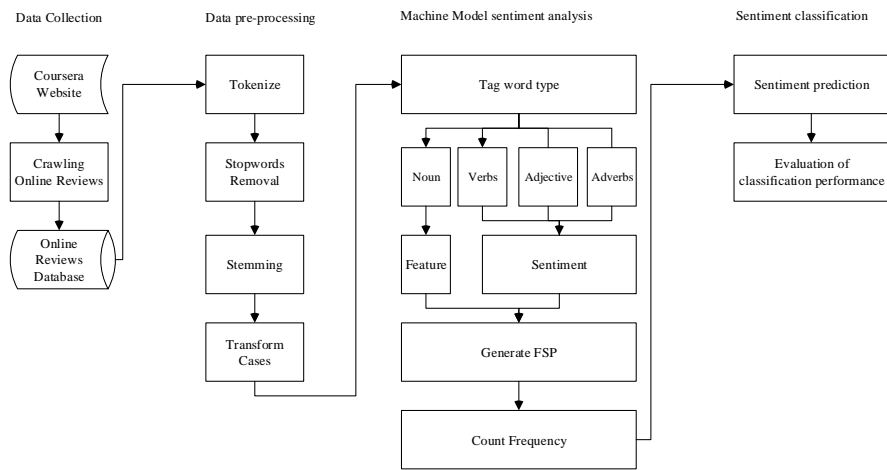


Fig. 1. Proposed Framework

The number of online review data crawled in total was 9677 reviews. To balance the data based on the sentiment review, they were divided into two types, namely negative and positive sentiment. The sentiment was used as a label of each review datum. The label had been manually labeled based on the star review in each review: 1-2-star review was categorized as a negative sentiment while 4-5-star review was a positive sentiment. The categorization was based on [10]. Table 2 shows the total number of data.

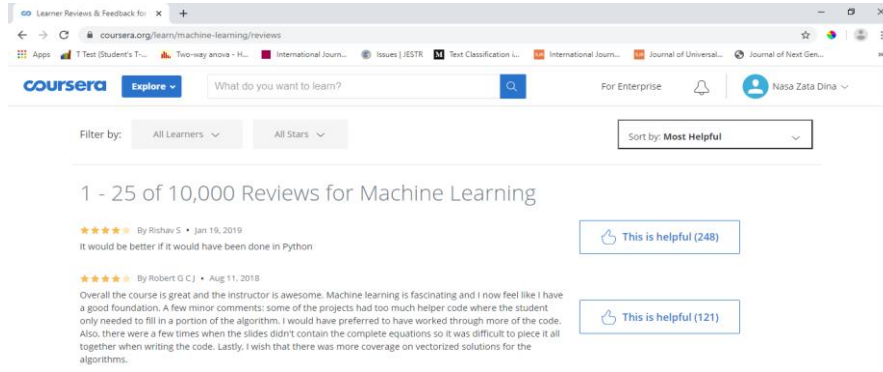


Fig. 2. Coursera Review Page

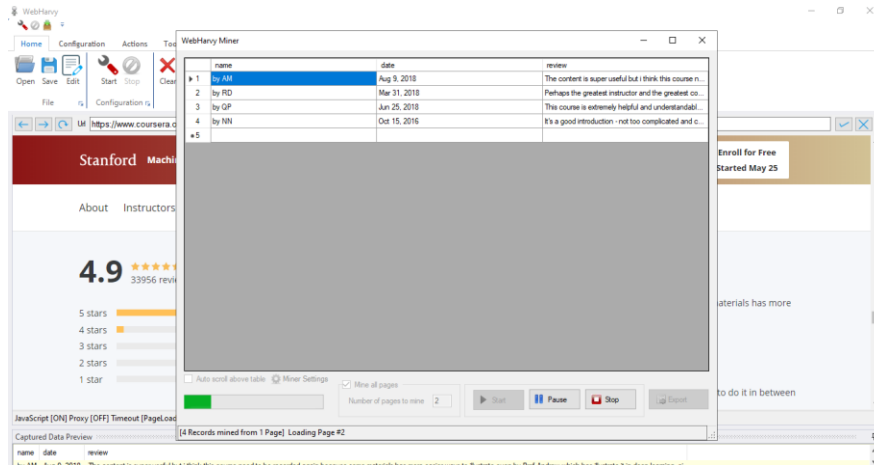


Fig. 3. Webharvy Web Scraper

Table 1. The example of online review data

Course's name	Review	Star review
html	Easy and understandable, really great of beginners!	5
analytics-excel	Great course for learning some data analysis with excel that is also accessible even if you have had little experience in excel before	5
angular-js	Interesting content, but so slow. 20 minutes long videos could have been done in 5 minutes. Moreover, the teacher's voice and tone did nothing to make the content live.	2

Table 2. The number of online review data

Label of review data	Number
Positive sentiment	4957
Negative sentiment	4720
Total	9677

2.2 Data pre-processing

Data pre-processing is the earliest step in which the data are being transformed into some features so that the algorithms are easily interpreted. Data pre-processing in text mining area includes four basic sub-processes, such as:

- 1) Tokenize
- 2) Stop words removal
- 3) Stemming
- 4) Transform cases

Tokenize is a process when a sentence was split into each word so that the review data for each record are divided into the number of words. Meanwhile, stop words removal removes the unimportant words include prepositions, articles, pronouns, numbers, and punctuation. Stemming is a process where words are breaking down the form of a word into its basic word form and deleting the suffix and unnecessary characters. Finally, transform cases means converting all the characters into upper case or lowercase.

The pre-processing step in this research had been done using RapidMiner tools. RapidMiner is open-source software for data mining, and it is popular among data scientists for its ease of use. Several studies in the text mining often use RapidMiner, such as the studies conducted by [11] [12]. Even though they used the data taken from different websites, such as Glassdoor web and Twitter web in which their data structure was different, Rapid Miner could process the data well. The example of the data pre-processing step in RapidMiner is illustrated in Fig. 4.

2.3 Machine model of sentiment analysis

There are three phases to complete this machine model:

- 1) Tag words types
- 2) Determine FSPs
- 3) Train the model

The objective of this machine model is to form Feature-Sentiment-Pairs (FSPs) from Noun, Verb, Adjective, and Adverb.

2.4 Tag word types via Part-of-Speech (POS) Tagger

The POS tagger used in this research was Stanford POS Tagger according to [13]. Stanford POS Tagger is software that can analyze the English text and determine its part-of-speech. The advantage of using Stanford POS Tagger is that it can be implemented using Java and Python languages. Table 3 shows the assumption of POS tags according to [14]. The POS tags were divided into nouns, verbs, adjectives, and adverbs. Other elements of part-of-speech such as prepositions, articles, and pronouns had been eliminated in the pre-processing phase called sub-process stop words removal [15].

2.5 Generate feature-sentiment-pairs from sentences

Feature-Sentiment-Pairs (FSPs) consist of Feature and Sentiment. Feature was from Noun and sentiment was from Verb, Adjective, and Adverb. They were paired after the data had been labeled as a positive and negative sentiment. If there are two Nouns written in the same review data, then both Nouns are paired with all the sentiment words [14]. Lastly, the FSPs were counted based on their appearance in the review data. The FSPs were also divided according to their label.

2.6 Train the model to predict sentiment

The Support Vector Machine (SVM) Algorithm is a classification algorithm that is considered as the best text classification method [16]. It is categorized as a supervised classification. The data were divided into several elements, and then each element was placed in n-dimensional space as vector coordinate [17]. A hyper plane is formed to distinguish each class. Fig. 5 illustrates that there is more than one hyper plane but only one that is the most optimal. The most optimal hyper plane is colored by the red line. It is called as an optimal hyper plane because it perfectly divides the classes.

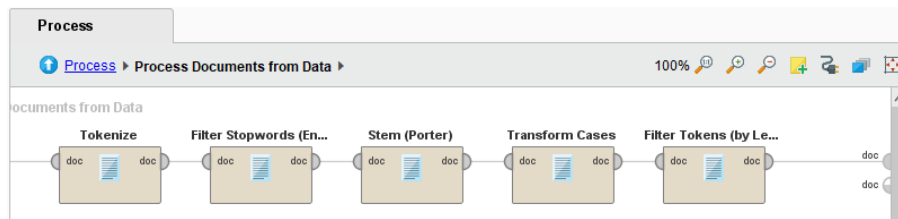


Fig. 4. Pre-processing step in RapidMiner

Table 3. Assumption for POS tags

Word types	POS tag	Assumption
Nouns	NN (noun singular)	All tags with “NN” = nouns
	NNS (noun plural)	
	NNP (noun proper)	
	NNPS (noun proper plural)	
Verbs	VB (verb base)	All tags with “VB” = verbs
	VBD (verb past tense)	
	VBG (verb present participle)	
	VBN (verb past participle)	
Adjectives	JJ (adjective)	All tags with “JJ” = adjectives
	JJR (adjective comparative)	
	JJS (adjective superlative)	
Adverbs	RB (adverb)	All tags with “RB” = adverbs
	RBR (adverb comparative)	
	RBS (adverb superlative)	

The optimal hyper plane is obtained by calculating the largest distance between the closest points of each class [18]. The hyper plane can be presented algebraically using Eq. 1 [17].

$$f(x) = w^T x + b \quad (1)$$

Eq. 2 is considered as training hyper plane where x is the closest point from the hyper plane. The closest value to the hyper plane is referred to support vectors. The length between a point x_i and the hyper plane (r_i) is calculated using Eq. 3. The formula of the margin (M) is shown in Eq. 4.

$$|w^T x + b| = 1 \quad (2)$$

$$r_i = y_i \frac{w^T x_i + b}{|w|} \quad (3)$$

$$M = \frac{2}{|w|} \quad (4)$$

It should be noted that SVM requires more data training so that the accuracy could be higher [19]. SVM is better than other classification algorithms if multiple classifications in the large dataset are involved. The pseudo-code of SVM is shown in Table 4.

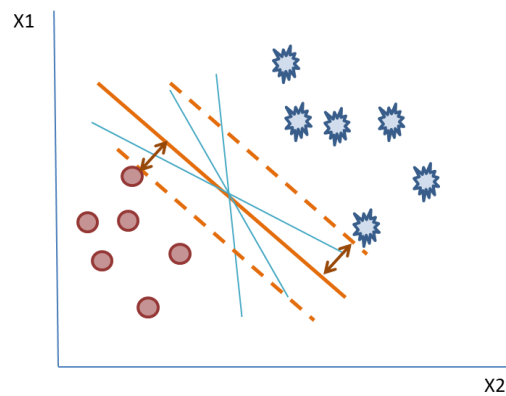


Fig. 5. The illustration of SVM Model

Table 4. Pseudo code of Support Vector Machine

Input	:	Input data
Output	:	<i>Support vectors set</i>
Begin		
Step 1	:	<i>Divide the data set into two groups and labelled based on the each class</i>
Step 2	:	<i>Add it into support vector set V</i>
Step 3	:	<i>Continue to divide the data items in n-times</i>
Step 4	:	<i>If there is data that is not assigned in a class then add data into set V</i>
Step 5	:	<i>Break if there is incomplete data</i>
Step 6	:	<i>end loop</i>
Step 7	:	<i>Train the data using the SVM classifier model and test the data to validate over the unlabeled data.</i>
End		

In general, the order of the text classification using SVM is term weighting, data training, and data testing. Term weighting used the occurrence number of terms in the documents. Term weighting is often called feature extraction. Feature extraction is used to transform a text document from any format into a list of features/terms that can be easily processed by text classification techniques. Feature extraction is one of the significant pre-processing techniques in the text classification that computes features/terms value in the documents. There are several methods of term weighting/feature extraction, such as Term Frequency (TF), Inverse Document Frequency (IDF), and a combination of TF and IDF which is called TF.IDF. Term Frequency (TF) is the frequency of occurrence of the term (t) in the document (d_i) as shown in Table 5. For example, the term “Increase” appears in four documents, and each document has a different number of Term Frequency (TF). Meanwhile, Document Frequency (DF) is the number of documents in which a term (t) appears. From Table 5, DF was calculated as illustrated by Table 6. After the DF value is obtained, then the IDF value is calculated by Eq. 5. The result of TF.IDF is obtained by multiplying the TF and IDF as seen on Eq. 6. The results of IDF and TF.IDF are shown in Table 6. The result of weighting is the formation of feature vectors by looking at the existence of a word in the document.

$$IDF = \frac{1}{DF} \tag{5}$$

$$TF.IDF = TF * IDF \tag{6}$$

Table 5. TF Weighting

Term (t)	Document 1 (d ₁)	Document 2 (d ₂)	Document 3 (d ₃)	Document 4 (d ₄)	Document 5 (d ₅)
Increase	0	1	2	5	4
Poor	1	4	0	0	3
Short	2	2	0	3	1
Course	0	0	1	2	0
Teacher	5	0	2	6	0

Table 6. DF, IDF and TF.IDF Weighting

Term (t)	DF	IDF	TF.IDF				
			(d ₁)	(d ₂)	(d ₃)	(d ₄)	(d ₅)
Increase	4	0.25	0	0.25	0.5	1.25	1
Poor	3	0.33	0.33	1.32	0	0	0.99
Short	4	0.25	0.5	0.5	0	0.75	0.25
Course	2	0.50	0	0	0.5	1	0
Teacher	3	0.33	1.65	0	0.66	1.98	0

Furthermore, a feature vector space was formed. Feature vector space was obtained by counting the number of terms present in the entire document. If there are 5 different terms in the entire document as shown in Table 5, then the feature vector will have 5 dimensions, where each part of the vector represents one term. Vector dimensions can be reduced by deleting insignificant words. After the vector space is formed, the vectors representing a document were formed using one of the weighting methods. In this study, TF weighting was chosen as weighting method. If in the first document there are 1 term “Poor”, 2 terms “Short”, and 5 terms “Teacher” then the feature vector for the first document is $x_2 = 1$, $x_3 = 2$, and $x_5 = 5$. After the feature vectors were formed, complete with their respective labels, they are ready to be inserted into the SVM to be used as data training [20]. The output of the SVM from the training process is the best hyperplane to be used as a classifier. The formula of hyperplane can be seen on Eq. 2. At the time of testing, the feature vectors used as testing data are entered into the SVM without labeling them. The output of the test results is in the form of a classification result class label. The class labels in this study determine the sentiment of the review whether it is positive or negative sentiment. To check the accuracy of the classifier, the output label is then compared with the original label as described in sub section 4 about evaluation of classification performances.

2.7 Evaluate classification performance

The classification performance is evaluated using a 2x2 confusion matrix according to [21]. The accuracy, precision, recall, and F1-score were calculated according to Table 7 and Eq. 7-10. The confusion matrix consists of actual class and predicted class. Accuracy calculates the number of true positive and true negative in the actual class and it is divided by the sum of total observation. Total observation is the sum of all the numbers in the confusion matrix as seen in Table 7. Meanwhile, precision calculates the ratio of true positive of the total number of true positive and false positive. Recall is the ratio of true positive of the total number of true positive and false negative. F1 Score is the mean of the precision and recall.

Table 7. 2x2 Confusion matrix

		Actual Class	
		Positive	Negative
Predicted Class	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{F1 Score} = \frac{2*(\text{Recall}*\text{Precision})}{(\text{Recall}+\text{Precision})} \quad (10)$$

2.8 Human model of sentiment analysis

In this research, we asked for annotators' assistance to model the sentiment analysis. The annotators were involved to label the review data and determined the FSPs. The results of the annotators' analysis was used as a control variable.

Interpret data: Annotators observed the sentiment analysis categorization apart from the results of star review generalizations from the crawl review data. In this phase, the annotators checked whether the sentiment analysis was in accordance with the review data.

Infer patterns from data: Unlike machine models, annotators could find synonyms and sarcasm. For example, there were two reviews: "great professors" and "good teacher". The annotators knew that both of the reviews were identical, "professor" means the "teacher" who taught the course. The annotators formulated the FSPs easier than machine model.

Measure the frequency of feature-sentiment-pairs: The frequency for every FSPs' appearance in the review was counted. To find out the significance of the FSPs, the easiest way was to calculate its appearance on a review. The output of this annotators' model was to select the highest frequency of FSPs.

3 Results

3.1 Sentiment classification result

The model was trained using the SVM algorithm. The review data were divided into two groups: Data training and data testing. For evaluation purposes, accuracy, recall and precision were calculated. The result is shown in Table 8. Ratio means the ratio of data training and data testing. In the first experiment, data training and data testing were on the same ratio, 50% of the total amount of the data. In the second experiment, data training was 80%, and data testing was only 20%. The evaluation result was different and the best result was when the ratio was 80:20. If the total data were 9677, then the number of data training on 80:20 ratios was 7741 and the number of data testing was 1936.

Table 8. The evaluation result

Ratio	Accuracy	Recall	Precision	F1 Score
50:50	77.64 %	77.21 %	81.25 %	79.17%
80:20	81.86 %	81.55 %	84 %	82.75%

The result of the machine model is illustrated in Fig. 6 and Fig. 7. The y-axis from both charts shows the frequency of negative and positive FSPs. The x-axis represents the FSPs. The most frequent negative FSPs in Fig. 6 indicate that the course was basic, poor, difficult and bored. Fig. 7 shows the most frequent positive FSPs are great, good, understand, and interesting.

3.2 Result of machine model sentiment analysis

As displayed in Fig. 6, the users of the course thought that the course was too basic as they expected. They also thought that the course was poor and boring. Also, they faced difficulties while following the course. The positive features can be seen in Fig.7. The users felt that the course was great and good. Since both words are synonym, so it is implied that the course was good enough. They found that the course was understandable and interesting to be followed. The positive and negative features are contrary to each other, so the results are compared with numbers. For example, the word “good” is the antonym of “poor”, but the frequency of the occurrence of “good” is higher than “poor”. This also applies to the words “understand” – “difficult” and the words “interesting” – “bored”.

3.3 Result of human model sentiment analysis

The result of the human model is illustrated in Fig. 8 and Fig. 9. The y-axis from both charts shows the frequency of negative and positive FSPs. The x-axis represents the FSPs. The most frequent negative FSPs in Fig. 8 indicated that the course only offered a few materials and they were difficult to comprehend. Fig. 9 shows the most frequent positive FSPs are course was good and interesting, and the introduction was great.

As displayed in Fig. 8, the users thought that only a few materials provided in the course, which did not meet their expectations. They also thought that the course was bad and its material was difficult. The positive features can be seen in Fig. 9. The users felt that the course was good and interesting. They found that the introduction of each course was great. The positive and negative features are contrary to each other, so the results are compared with numbers. For example, the word “good” is the antonym of “bad”, but the frequency of occurrence of “good” is higher than “poor”.

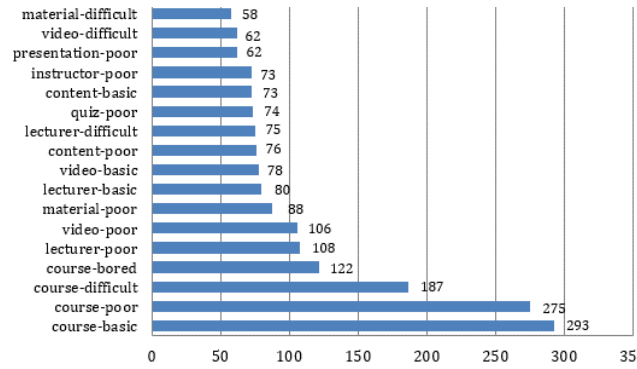


Fig. 6. Frequency of Negative FSPs from Machine Model

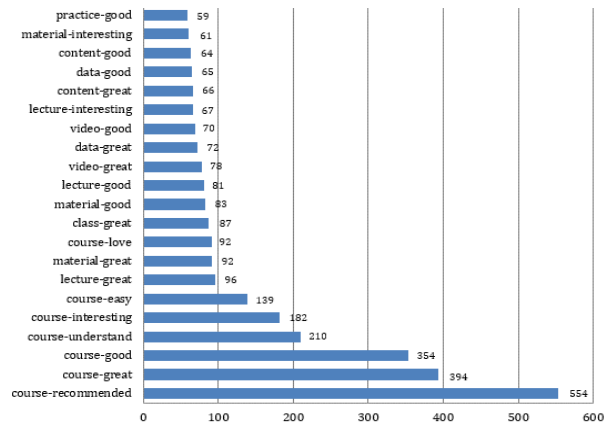


Fig. 7. Frequency of Positive FSPs from Machine Model

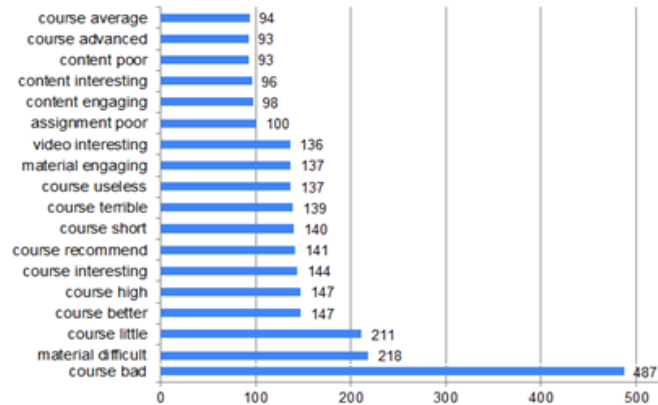


Fig. 8. Frequency of Negative FSPs from Human Model

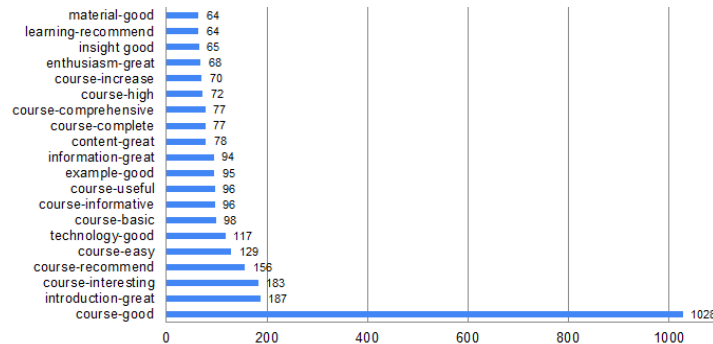


Fig. 9. Frequency of Positive FSPs from Human Model

4 Discussion

The Machine Model and Human Model were compared to measure the success of the framework. The Human Model was also referred to ground truth because the result of the Human Model came from the expert analysis. The success of the framework was potentially shown by the similarity of the Human Model and Machine Model. The positive features produced by the Machine Model were “course-good”, “course-interesting”, “course-easy”, “course-understand”, “course-recommended”, and “material-good”.

Those positive features were also the most frequently occurred FSPs in the Human Model. Even though the number of each positive feature such as “course recommended” in the Machine Model does not exactly have the same number in the Human Model, the frequent FSPs in the Human Model and Machine Model are mostly the same. As illustrated in Fig. 7 and Fig. 9, most of the positive FSPs in the Machine Model appear in the Human Model. Similarly, in the negative features, the most frequently occurred FSPs produced by the Machine Model are also nearly the same as those produced by the Human Model. Further study needs to be carried out to improve the accuracy of the number of FSPs in both Machine Model and Human Model.

The result of FSPs from the Machine Model towards the Human Model proves that the framework managed to transform the unstructured data into meaningful information. Since the data were used as the main source in decision making in this framework, the approach used in this framework was data-driven design. The data-driven design made decisions regarding the development of content-related design and system design fully based on the data collected from the MOOC, specifically the data about how MOOC users interact with the system as seen on MOOC discussion forum.

There were similar studies which used the same data from MOOC discussion forum. For example, Wise et al in 2016 developed a linguistic model to categorize and identify the post in MOOC discussion forum whether or not they are substantially related to the course content by searching for predefined keywords [22]. It helps the

instructors to identify the content-related question from the learners, so that the instructors could improve their course material. However, the predefined keywords which they proposed were only for a certain course. Thus, if they changed the course, then the keywords had to be changed according to the course. Brinton et al used large scale statistical analysis of forum discussion in Coursera [23]. They investigated the user behaviour on the forum discussion and looked for the most course-relevant discussion. They ranked the discussion topic based on its relevance with the course using a unified generative model. Gamage et al [24] also studied user behaviour as done by [23]. The difference was the method used. [24] applied an ethnographic method by using a deep interview with two groups of participants. The first group never used MOOC, and the other one used MOOC. From the deep interview, the researcher could conclude the recommendation for MOOC design. Agrawal et al made Stanford MOOCPost Dataset which was tagged manually according to six dimensions including confusion, question, answer, opinion, sentiment, and urgency to address the confusion in the MOOC discussion forum. They also gave a recommendation to the user to open instructional video clips [25]. They classified the confusion by using “tracking log data” from log learners’ actions [25].

The limited number of datasets in this study was under 10000 online reviews, but the Machine Model was able to produce information to support the decision-making process in MOOC. The data itself were divided into equal number of positive and negative labels as shown in Table 2. The number of data of each label should be equal to avoid deviation towards the dominant label. The accuracy, precision, recall, and F1 score were above 80%. The value was high enough even though the number of the dataset was relatively moderate. Thus, adding more data is compulsory in further studies to obtain higher accuracy, precision, recall, and F1 score. The framework transforms qualitative online review data into quantitative data. Using this framework, the designer does not need to do a manual analysis to identify user preference for each feature in MOOC. All the decision was generated automatically based on online review data as an input.

The quantitative result helps the designer to find words and their frequency that often appear in the review data. It will give suggestions to the designer to add new features or improve the existed features in MOOC especially content-related features. Although using quantitative results is easier to validate and analyse the data, it also works well in large scale data. However, there are several weaknesses in using quantitative results. Unlike qualitative data, quantitative did not have more specific information because the sentences from online review data were separated into each word. Since they had been separated, it never revealed the causation on why the user indicated the positive and negative review about particular features. FSPs only paired the adjectives and nouns that were contained in a sentence but it was lack of details, and it never revealed the causation, too. The designer used FSP to identify certain popular comments about content-related features on the data. Since FSP consisted of adjectives and nouns, the designer was able to conclude which features that received a positive response and vice versa.

There were differences between the results as shown in Fig. 6-9. If we compared the result between Machine Model and Human Model, the Human Model was always

more accurate than Machine Model in terms of grouping the FSP. Some of the causes of computational processing errors include:

- 1) Failure to produce “stemmed words”
- 2) Incorrectly giving tagged word of “part of speech” to each word
- 3) Failure to generate FSP in a sentence

The first cause often happened when the stemming algorithm failed to generate stem words. There were some stemming algorithms that were often used. There were Porter’s Stemmer, Lovins Stemmer, Dawson Stemmer, Krovetz Stemmer, and Paice/Husk Stemmer [26] [27]. Among some stemming algorithms, in this study, we used Porter’s Stemmer because it produced the best output as compared to other stemming algorithms and the error rate was quite low [26] [27]. The weakness of Porter’s Stemmer which often happened and still could not be avoided was over-stemming when the suffix was erased, and it caused a change in the meaning and wrong-stemming when the part of the root was also deleted. The second cause was incorrectly giving tagged word type. In this study, we tagged the words only for Noun and Adjective because those were the main elements of FSP. Since FSP consisted of “features” and “sentiments”, features were derived from words labelled with nouns while sentiments were derived from words labelled with adjective. As the process of tagging the word type was carried out after the data pre-processing, the output of data pre-processing greatly affected the result of tagging word type. Finally, FSP was often missing from the sentence. It means that, in a sentence, there were no Noun and Adjective. If both Noun and Adjective were not in a sentence, then that sentence was not considered as a dataset. It could reduce the number of valid data. Because of these flaws, there was a possibility of valid data reduction, and it affected the number of FSP on the result. [28] offered a solution to improve the Machine Model’s accuracy to pair sentiment with features. [28] measured the distance between Noun and Adjective. This strategy will be very suitable to be applied in further studies. Another limitation of this study was the number of dataset which was under 10000 data. The dataset was decreased after some processes were applied. Due to the moderate dataset, there were many missing data. The greater quantity of the dataset will increase the accuracy and the information will be more varied for the designer.

5 Conclusion

The proposed framework produced useful information for supporting data-driven design in MOOC. MOOC could use FSP as an evaluation of its content. It supports data-driven decision making in content-related features. The sentiment analysis classification had been done using Support Vector Machine, and the number of accuracy, precision, recall, and F1 score was above 80%. The positive and negative FSPs had successfully described the user preferences in MOOC, especially in Coursera. The output was a list of specific features and the frequency of the FSP so that it could be ranked based on the appearance of the most frequent FSP. The designer could evaluate MOOC content-related features to improve the product’s content features. Further

study is needed to improve the accuracy of the result of Machine Model by adding more dataset and reducing computational processing errors.

6 References

- [1] Cooper, S., Sahami, M. (2013). Reflections on Stanford's MOOCs. *Communications of the ACM*, 56(2), pp. 28-30. <https://doi.org/10.1145/2408776.2408787>
- [2] Harder, B. (2013). Are MOOCs the future of medical education? *British Medical Journal*, 346, pp. 1-3.
- [3] Martin, F. G. (2012). Will massive open online courses change how we teach? *Communications of the ACM*, 55(8), pp. 26-28. <https://doi.org/10.1145/2240236.2240246>
- [4] Anand, S.S. and BÜCHNER, A.G., (1998). Decision support using data mining. *Financial Times Management*.
- [5] Bertoni A. (2018). Role and Challenges of Data-Driven Design in the Product Innovation Process. In: *Proceedings of 16th IFAC Symposium on Information Control Problems in Manufacturing*, Italy, 2018. Bergamo: Elsevier, pp. 1107-1112. <https://doi.org/10.1016/j.ifacol.2018.08.455>
- [6] Miner, L., Bolding, P., Hill, T., Nisbet, R., Walton, N. and Miner, G. (2014). *Practical Predictive Analytics and Decisioning Systems for Medicine*. 1st ed. United States: Academic Press. <https://doi.org/10.1016/b978-0-12-411643-6.00044-2>
- [7] Dina, N.Z. (2020). Tourist sentiment analysis on TripAdvisor using text mining: a case study using hotels in Ubud, Bali. *African Journal of Hospitality Tourism and Leisure*, 9(2), pp. 1-10.
- [8] Mallik, R., Sahoo, A. K. (2018). A novel approach to spam filtering using semantic based naive bayesian classifier in text analytics. In: *Proceedings of IEMIS, India, 2018*. Singapore: Springer, pp. 301-309. https://doi.org/10.1007/978-981-13-1498-8_27
- [9] WebHarvy: Intuitive Powerful Visual Web Scraper, <https://www.webharvy.com/>
- [10] Decker, R. and TRUSOV, M. (2010). Estimating Aggregate Consumer Preferences from Online Product Reviews. *International Journal of Research in Marketing*, 27(4), pp. 293-307. <https://doi.org/10.1016/j.ijresmar.2010.09.001>
- [11] Dina, N.Z. and Juniarta, N. (2020). Aspect based Sentiment Analysis of Employee's Review Experience. *Journal of Information Systems Engineering and Business Intelligence*, 6(1), pp. 79-88. <https://doi.org/10.20473/jisebi.6.1.79-88>
- [12] Vyas, V. and UMA, V. (2018). An Extensive study of Sentiment Analysis tools and Binary Classification of tweets using Rapid Miner. *Procedia Computer Science*, 125, pp. 329-335. <https://doi.org/10.1016/j.procs.2017.12.044>
- [13] Toutanova, K., Klein, D., Manning, C. and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: *Proceedings of HLT-NAACL, Canada, 2003*. Edmonton: ACM, pp. 252-259. <https://doi.org/10.3115/1073445.1073478>
- [14] Ireland, R. and Liu, A. (2018). Application of data analytics for product design: Sentiment analysis of online product reviews. *CIRP Journal of Manufacturing Science and Technology*, 23, pp. 128-144. <https://doi.org/10.1016/j.cirpj.2018.06.003>
- [15] Verna, T., Renu, R. and GAUR, D. (2014). Tokenization and Filtering Process in RapidMiner. *International Journal of Applied Information Systems*, 7, pp. 16-18. <https://doi.org/10.5120/ijais14-451139>

- [16] Jin, J., Ji, P., Liu, Y. (2016). Identifying comparative customer requirements from product online reviews for competitor analysis. *Engineering Applications of Artificial Intelligence*, 49, pp. 61-73. <https://doi.org/10.1016/j.engappai.2015.12.005>
- [17] Ray, S. Understanding Support Vector Machine Algorithm from Examples, 6.10.2015 [Online]. Available: <https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/>[Accessed 6 July 2020]
- [18] Manning, C.D., Raghavan, P., Schütze, H. (2008) *Support Vector Machines: The Linearly Separable Case*. Cambridge University Press.
- [19] Adomavicius, G., Tuzhilin, A. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp. 734–749. <https://doi.org/10.1109/tkde.2005.99>
- [20] Purnamawan, I. (2015). Support Vector Machine pada Information Retrieval. *Jurnal Pendidikan Teknologi dan Kejuruan*, 12(2), pp. 173-180. <https://doi.org/10.23887/jptk.v12i2.6481>
- [21] Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. (Article in Press)
- [22] Wise, A. F., Cui, Y. and Vytasek, J. M. (2016). Bringing order to chaos in MOOC discussion forums with content-related thread identification. In: *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, United Kingdom, 2016. Edinburgh: ACM, pp. 188–197. <https://doi.org/10.1145/2883851.2883916>
- [23] Lam, H., Liu, Z. and Wong, F. M. F. (2014). Learning about Social Learning in MOOCs: From Statistical Analysis to Generative Model. *IEEE Transactions On Learning Technologies*, 7(4), pp. 346-359. <https://doi.org/10.1109/tlt.2014.2337900>
- [24] Agrawal, A., Venkatraman, J., Leonard, S. and Paepcke, A. (2015). YouEDU: addressing confusion in MOOC discussion forums by recommending instructional video clips. In: *Proceedings of the 8th International Conference on Education Data Mining*, Spain, 2015. New York: ACM, pp. 297-304.
- [25] Gamage, D., Perera, I. and Fernando, S. (2020). Exploring MOOC User Behaviors Beyond Platforms. *International Journal of Emerging Technologies in Learning*, 15(8), pp. 161-179. <https://doi.org/10.3991/ijet.v15i08.12493>
- [26] Moral, C., De Antonio, A., Imbert, R. and Ramírez, J. (2014). A survey of stemming algorithms in information retrieval. *Information Research*, 19(1), pp. 605.
- [27] Jivani, A. G. (2011). A comparative study of stemming algorithms. *International Journal of Computer Technology and Applications*, 2(6), pp. 1930-1938.
- [28] Hu, M. and Liu, B. (2004) Mining opinion features in customer reviews. *American Association of Artificial Intelligence*, 4(4), pp. 755–760.

7 Authors

Nasa Zata Dina is a lecturer in the Department of Engineering, Faculty of Vocational Studies at Universitas Airlangga, Surabaya, Indonesia. Her research interests include data mining, text mining, and engineering education. She is also an editorial board member of *Journal of Information Systems Engineering and Business Intelligence*.

Riky Tri Yunardi is a lecturer in the Department of Engineering, Faculty of Vocational Studies at Universitas Airlangga, Surabaya, Indonesia. His research interests include medical electronics, robotics, image processing, artificial intelligence, and engineering education.

Aji Akbar Firdaus is a lecturer in the Department of Engineering, Faculty of Vocational Studies at Universitas Airlangga, Surabaya, Indonesia. His research interests include power systems simulation, power systems analysis, power systems stability, renewable energy, artificial intelligence, and engineering education.

Article submitted 2020-07-17. Resubmitted 2020-09-05. Final acceptance 2020-09-07. Final version published as submitted by the authors