Chotisarn, N., Siriphongwatana, P., Thanisuwiphat, P., Gulyanon, S., & Nadee, W. (2022). Finding customer behavior insights for content creation in material and product sourcing using specialized topic analysis. In Li, E.Y. *et al.* (Eds.) *Proceedings of The International Conference on Electronic Business, Volume 22* (pp. 177-184). ICEB'22, Bangkok, Thailand, October 13-17, 2022

Finding customer behavior insights for content creation in material and product sourcing using specialized topic analysis

Noptanit Chotisarn [1]
Phanuphong Siriphongwatana [2]
Phattranit Thanisuwiphat [2]
Sarun Gulyanon [3]
Winai Nadee [1,*]

*Corresponding author
[1] Department of Management Information Systems, Thammasat Business School, Thammasat University, Bangkok, Thailand, {noptanit, winai}@tbs.tu.ac.th
[2] Data Science and Innovation Program, College of Interdisciplinary Studies, Thammasat University, Pathum Thani, Thailand, {phanuphong.sir, laksika.sue}@dome.tu.ac.th
[3] Data Science and Innovation Program, College of Interdisciplinary Studies, Thammasat University, Pathum Thani, Thailand, sarung@staff.tu.ac.th

ABSTRACT

In content creation, customer behavior insights are very important as they help creators find and create the content that drives sales. To comprehend customer needs, content creators need not just generalized information but also specific information, which can be different across markets and cultures. This information also needs some standards so it can be analyzed systematically. This paper aims to obtain customer insight into web content. Inside the web content, one possible source of this information is the tags based on customer feedback and the related entities. In this case, the product review data were collected and analyzed. However, manually analyzing feedback is a time-consuming activity. In this work, we formulated the topic analysis problem specialized for material and product sourcing, which could benefit product analysis and development. Technically, we also compared different text processing and classification methods, which set the benchmarks for reviewing the model performance in the future.

Keywords: Text classification, Customer behavior insights, Content creation, Specialized topic analysis, Material and product sourcing.

INTRODUCTION

Customer behavior insights are most important in marketing technology. Without this information, the content creators have no clues about the products or services that can attract customers, so they can only guess at their best. Without the right content, it is next to impossible to drive sales. The cost of not knowing your customers include spending resources without bringing any value and, in the worst case, losing the customers we worked so hard to acquire. This might be referred to as an unknown known problem since we do have data, but we still need to dig deeper to reach something called insights (Agrawal et al., 2018).

To comprehend the customers' needs, content creators need to obtain specific information, e.g., identifying recurrent themes or topics that align with the stakeholders' needs, which can be different across markets and cultures. One way to obtain consumer insights is to analyze the contents that customers usually engage with since they indicate the types of content that customers are interested in.

Since fully manual analysis of contents and articles is a laborious and error-prone process, text classification can be used to organize and understand extensive collections of text data by assigning tags or categories to each text's topic, theme, or entities (i.e., products) of interest. Then, the manual analysis is limited to only relevant tags, which makes it more feasible but still an uphill task as the number of tags can be high, so it is difficult for the analysis to include all the relevant and correct tags.

Moreover, the assigned tags must be defined at the right level of specifications; otherwise, if it is too generalized, the tags give no new information, while if they are too specific, only a handful of text data will fall into the categories. We identified this problem as the narrow topic analysis problem.

In this work, we first formulated the narrow topic analysis problem for material and product sourcing in the architecture industry. Our narrow topic analysis involves the classification of text articles based on two types of tags: product categories and themes. This problem is challenging because product category tagging is a hierarchical classification. At the same time, the notion of themes, in this case, involves abstract, vague, and debatable concepts, such as styles and trends, since they are based on the audience's point of view. As a result, there are no consensus labels for the theme tags.

To address the above challenges of the narrow topic classification problem for creating content in material and product sourcing, we adopted the framework as shown in Fig.1, which consists of data processing for standardizing the text, feature extraction for finding the text embedding, and machine learning model for narrow topic classification. Another issue that must be addressed the formulation of tags for the narrow topic analysis problem. When the task is ill-defined, the machine learning solution can be beneficial as it can act as the referee and gives the standard, which everyone can follow, to the problem.

However, the key ingredient needed for the AI-based solution is that all stakeholders (i.e., IT and business units in our case) must settle for the tags. All stakeholders do not need to agree on all tags, but each stakeholder must approve that the tags they need are covered.

Hence, in this paper, our contribution includes (a) a case study of formulating the narrow topic analysis problem in understanding customer behavior for creating content in material and product sourcing, which shows how the problem can be formulated even when the task is vague and ambiguous, and (b) the comparison of methods within the proposed framework for the narrow topic classification.

RELATED WORK

This article is inspired by a work categorizing products with images by a twin neural network for underspecification analysis in product design matching models (Chotisarn et al., 2021). But for this work, we changed the business domain for categorizing articles with language processing techniques.

Web extraction is an integral part of business analytics these days. World Wide Web has been considered the main source of customer behavior data. To collect data for further problem-solving, data must be extracted from origins and stored in temporary storage. Next, data will be parsed and transformed to extract the relevant information. This process might be conducted regularly (Prutsachainimmit & Nadee, 2018).

A review of text classification by Minaee et al. (2021) shows multiple applications of text classification tasks, including Sentiment Analysis, News Categorization, Topic Analysis, Question Answering (QA), and Natural language inference (NLI). The methods for automatic text classification can be categorized into two groups: rule-based methods and data-driven-based methods.

Rule-based methods classify text into different categories using a set of pre-defined rules defined by experts in the corresponding domains. The main advantage of this kind of method is interpretability, which is the ability to explain how the outcome is derived understandably. However, the downfall is that some tasks exist (i.e., theme tag classification in our case), that even the experts cannot agree on a consensus among themselves, so agreeable rules cannot be derived.

Data-driven based methods or machine learning-based methods have gained lots of attention recently. Typical machine learning-based models follow the two-step procedure, including feature extraction and classifier.

The first step involves computing some hand-crafted features from the article or any other textual unit of interest. The second step is to feed these features to a classifier to predict if they are new/unseen data or provide them to train a classifier if their outputs are known. The popular choices of classification algorithms include Naïve Bayes (John & Langley, 1995) and support vector machines (SVM) (Cortes & Vapnik, 1995).

METHOD

Our method follows the steps shown in Fig.1. First, we defined the tags, which are the expected outputs of the narrow topic analysis. Then, data processing is explained to normalize the texts. Next, feature extraction is discussed to find the article's representation (i.e., the embedding) suited for the problem. Finally, the machine learning models are described and compared with their results on our task of the narrow topic analysis problem in understanding customer behavior for creating content in material and product sourcing.

Defining Tags

Two tag sets are needed by the content creator: theme tags and product tags. These two tags are derived from every article. Their definition is given as follows:

1) Theme tags are the keywords that repeatedly appear in the articles. These tags then go through consultation between IT and business units to settle the set of tags that are useful for the business unit, while the IT unit keeps track of the complexity of the problem. The problem is defined as the multilabel classification, where there are possibly multiple numbers of targets for each article, while the target cardinality is two, which indicates whether the article is considered in this category.

In our case study of creating content in material and product sourcing, there are 57 tags at the end. The 57 categories can be grouped as an overview as follows;

a. Design style, for example, modern, loft, vintage.
b. Part of construction, for example, pole, beam, wall.
c. Room types, for example, bedroom, bathroom, living room.
d. Exterior, for example, terrace, garden, parking.

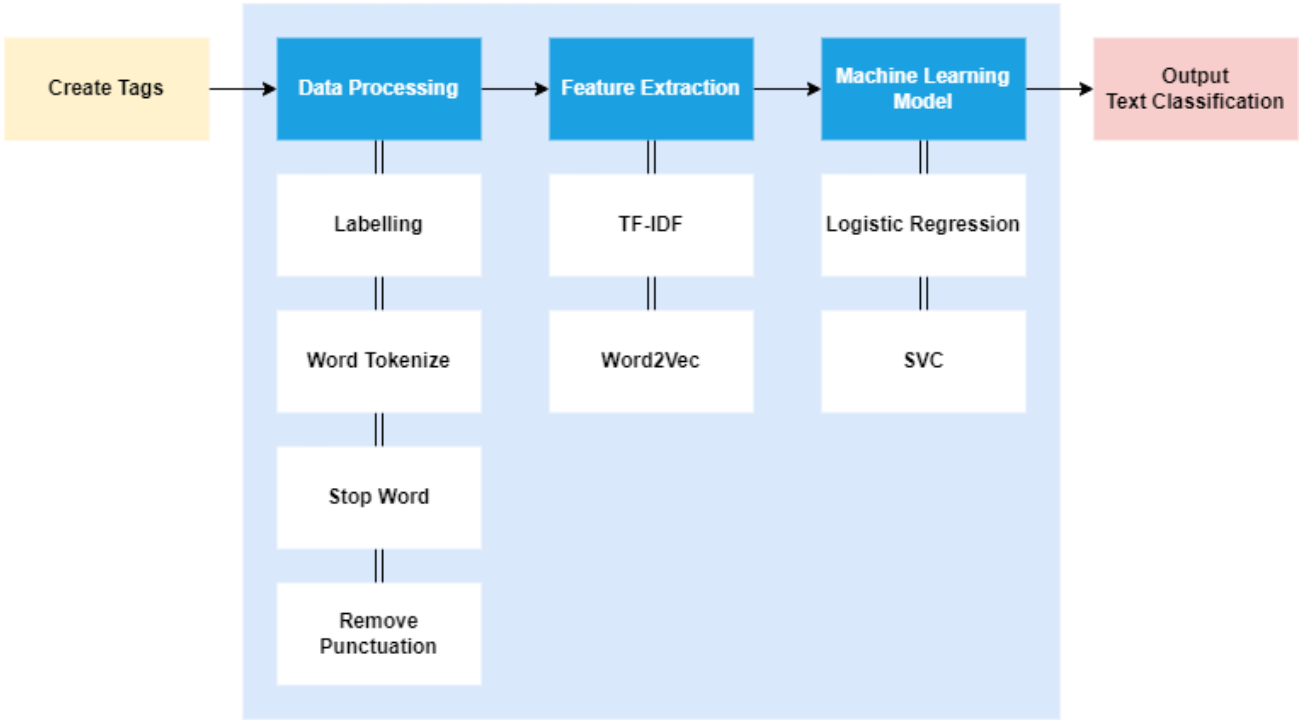

Fig.1 shows the overall steps of our method.

2) Product tags classify construction equipment by material type, shape, and how it is used. This information is provided by the authors of the articles but as a non-hierarchical tag, which may be incomplete or inconsistent according to the tag hierarchy we adopted. Therefore, the classification of product tags is meant to categorize and subcategory articles that are pre-tagged with human resources.

The product tag we adopted in this work consists of 23 categories and 167 sub-categories. The 23 categories can be grouped as an overview as follows;

a. Provide information, ideas, inspirations, and history.
b. Hidden advertising.
c. Summary of the seminars, events, and interviews.
d. Product and service review.

Fig.2 shows the example of labelling of the contents.

Data Preprocessing

In this section, we prepared data to be ready for feature extraction. We perform the preprocessing step on the content and title of the articles. First, load data and fill fields that are not labeled with the number 1, add the number 0 to create a label and detect numbers and links, and label them to eliminate the unimportant parts of the text before the training model, as shown in Fig.2.

Then, clean data in the 'title' and 'content' columns with th-simple-preprocessor to remove the special characters. Word tokenization is performed with the content column and title column using "newmm[1]" — Dictionary-based Thai Word Segmentation using the maximal matching algorithm and Thai Character Cluster (TCC) (Theeramunkong et al., 2000) tokenizer extracted from PyThaiNLP as shown in Fig.3.

Word tokenize which are the process of splitting a sentence, phrase, an entire text document into smaller units, each of these smaller units is called tokens. Tokens help find patterns, and tokenization allows substituting sensitive data elements with non-sensitive data elements (Phatthiyaphaibun et al., 2016). To remove stop words in Thai, we import thai_stopwords from pythainlp.corpus.common[2].

Word Vectorization

It is the process of converting data into a form that machine learning techniques can utilize. Feature extraction aims to reduce the number of features in a dataset by creating new features from the existing ones. Two feature extraction methods are studied in this work, including TF-IDF and Word2Vec (Mikolov et al., 2013).

1) TF-IDF or Term Frequency – Inverse Document Frequency is a combination of the Term of frequency (TF) and Inverse document frequency (IDF). TF-IDF is a technique that primarily considers the composition of words in the document without considering the sequence of words within the document.

   The term frequency (TF) measures how many times a term is present in a document. For example, document "A1" contains 1,000 words, and the word "Sky" is present in document "A1" 25 times. So, the occurrence of terms in the document is divided by all the words the record contains. In this example, the term frequency of "Sky" will be TF = 25/1,000 = 0.025.

   Inverse document frequency (IDF) measures the matter of words that appear in all documents. Words that frequently appear in all documents tend to be less important. For example, if there are 20 documents and the term "Sky" appears in 5 documents. In this example, the inverse document frequency will be IDF = loge(20/5) = 0.602.

2) Word2Vec (Mikolov et al., 2013) is a classical method for creating word embeddings in the field of Natural Language Processing (NLP). It was developed by Tomas Mikolov and his team at Google in 2013. Word2Vec represents a "word" as a "vector," where these vectors are chosen using the cosine similarity function, indicating the semantic similarity between words.

```
from newmm_tokenizer.tokenizer import word_tokenize

text = 'หากคุณกำลังสนใจการตกแต่งให้พื้นที่เล็กๆ ภายในบ้านให้ดูสวย มีระดับ หรูหรา วิจิตรอลังการ
words = word_tokenize(text)

print(words)
```

```
['หาก', 'คุณ', 'กำลัง', 'สนใจ', 'การ', 'ตกแต่ง', 'ให้', 'พื้นที่', 'เล็ก', 'ๆ', ' ', '
```

Fig.3 shows the usage of "newmm" library, for example, a part of a content article in the Thai language.

Text Classification Models

In this work, three machine learning models are studied, including Logistic regression, SVM, and Label Powerset with the Gaussian Naïve Bayes model.

[1] https://github.com/wisesight/newmm-tokenizer

[2] https://github.com/stopwords-iso/stopwords-th

1) Logistic Regression (LR) is selected as the baseline as it is one of the most common and fundamental models for classification problems, with a dependent variable as a discrete variable. Logistic regression uses the sigmoid function to map predicted values to probabilities.

2) SVM (Cortes & Vapnik, 1995) is a supervised machine learning algorithm that can be used for classification or regression problems. They are many possible hyperplanes that can be chosen separate two classes of the data point. So, the objective of this algorithm is to find the optimal hyperplane in an N-dimensional space; N is the number of features. That distinctly classifies the data points. N is the number of features. The optimal hyperplane, with the maximum margin, provides some reinforcement so that future data points can be classified more confidently in an N-dimensional space that distinctly organizes the data points.

3) Label Powerset with Gaussian Naive Bayes (Label-GB) uses Gaussian Naive Bayes to classify the hierarchical label seen as a flat label through the powerset. Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. Naive Bayes is an algorithm based on the Bayes theorem. It is a simple classification technique but has high functionality. They find use when the dimensionality of the inputs is high. We use Label Powerset to transform the problem into a multi-class problem with one multi-class classifier, which is trained on all unique label combinations found in the training data.

EXPERIMENTS

We scraped and collected from "https://www.wazzadu.com/" during June and July 2021. The dataset used in the experiments contains 3,139 articles, which is split into training and test data with the ratio of 2:1. In all experiments, for word vectorization, TF-IDF uses all available words after the data preprocessing and Word2Vec uses the pre-trained vector of 300 dimensions publicly available from PyThaiNLP (Phatthiyaphaibun et al., 2016). All possible combinations of two word-vectorization techniques and three classification methods are used to perform theme tag and product tag classification (Label-GB is only applicable to the hierarchical tag, like product tags).

Two word-vectorization techniques use Python packages that include sklearn (Pedregosa et al., 2011, Buitinck et al., 2013), Word2Vec, and PyThaiNLP as follows;

1) sklearn.feature_extraction.text.TfidfVectorizer converts a collection of raw documents to a matrix of TF-IDF features equivalent to CountVectorizer followed by TfidfTransformer. In our research, except for using TfidfVectorizer directly, we also use the CountVectorizer followed by TfidfTransformer.

2) For this article, we will start Word2Vec with the Thai language. First, install pythainlp which only supports Python 3. Next, make Thai Word2Vec in Python with the Gensim module with import Word2Vec from gensim.models and import word_tokenize from pythainlp.tokenize.

Three classification methods use Python packages as follows;

1) Logistic Regression was implemented in Python by importing LogisticRegression from sklearn.linear_model. Classifier for Logistic Regression (also known as logit or MaxEnt). In this research, the "multi class" option is set to "multinomial," and the training algorithm uses the cross-entropy loss.

2) Support Vector Machine was implemented in Python by sklearn.svm. In this research, sklearn.svm.SVC (C-Support Vector Classification) was selected for implementation, which is built on top of libsvm. Fit time scales at least quadratically with sample number and may be impractical beyond tens of thousands of samples. Consider using LinearSVC or SGDClassifier instead for large datasets.

3) Lable Powerest with Gaussian Naive Bayes (Label-GB) was implemented in Python by importing LabelPowerset from skmultilearn.problem_transform and importing GaussianNB from sklearn.naive_bayes. Lable Powerest is transforming a multi-label problem into a multi-class problem. The multi-label problem is the multiple output classes at once, while the multi-class problem is only one output at a time.

The GaussianNB is a classification using probability principles to help calculate. Lable Powerest with Gaussian Naive Bayes (Label-GB) can use together as initialize Label Powerset multi-label classifier with a gaussian naive bayes base classifier "classifier = LabelPowerset(GaussianNB())"

The metrics used for evaluation (Géron, 2022) include precision, recall, and f1-score. Precision is the fraction of relevant instances among the retrieved instances, computed by formula (1), where TP is true positives and FP is false positives. The recall is the fraction of retrieved relevant cases calculated by the formula (2), where FN is a false negative. The F1-score is the harmonic mean of precision and recall, formula (3).

$$Precision = TP / (TP+FP) \quad (1)$$

$$Recall = TP / (TP+FN) \quad (2)$$

$$F1 = 2PR / (P+R) \quad (3)$$

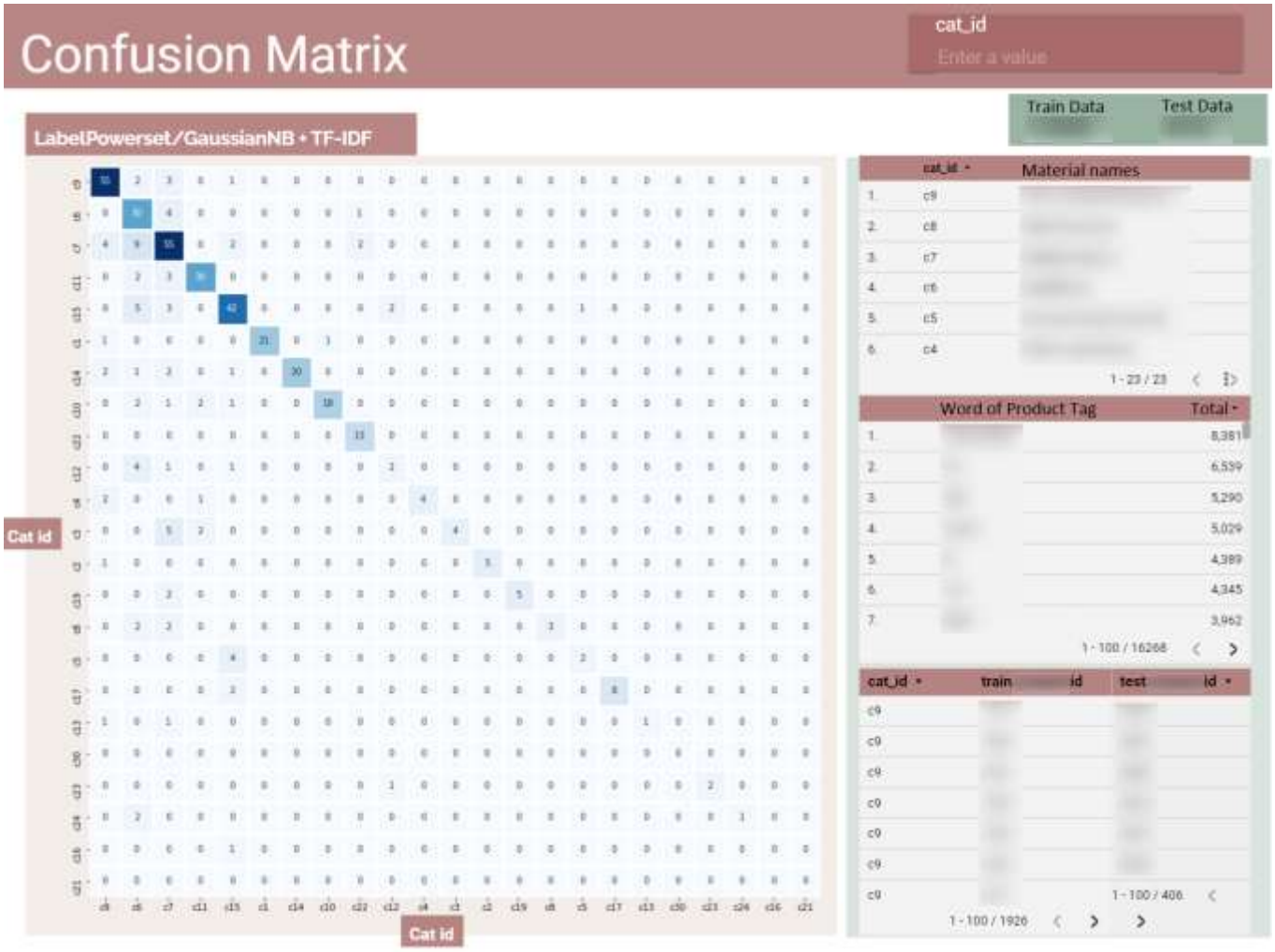

Fig.4 A confusion matrix to view model results and visualize with Looker Studio.

RESULTS

Initially, we created a confusion matrix (Géron, 2022) to view model results and visualize with Looker Studio (Fig.4). Looker Studio, formerly Google Data Studio, is an online tool introduced by Google for converting data into customizable informative reports and dashboards (Kemp & White, 2021). Google announced a free Data Studio version for individuals and small teams in May 2016. In the example, we looked at LR + TF-IDF. The confusion matrix is used to analyze whether the predictions are confused or not in each category.

For example, balcony and terrace, there may be confusion that appears in the confusion matrix that the balcony is predicted to be the terrace and the terrace is expected to be the balcony. In this case, we group it as the name c, which stands for the material category, followed by a number instead of the name. The remaining confusion matrix can be viewed to view the remaining results. But we tend to look at that matrix more broadly.

For precise predictions, the results are dark colors at the intersection of the x and y axes in the same category. We sorted them by color intensity to see which type predicted well, and the expected group had poor results. The confusion matrix helps to analyze how the model is wrong or our grouping is wrong. When viewed alongside the Precision, Recall, and F1-Score results to further improve our classification.

However, the confusion matrix is not suitable for viewing results to compare each model. We, therefore, summarize each model with an F1-score, as shown next.

Tables 1 and 2 show the test results of theme tag and product tag classification, respectively. The results show that SVM combined with TF-IDF gives the best theme and product tag classification outcome. SVM is one of the most popular methods since it usually outperforms other methods and is versatile in flat and hierarchical tags.

On the other hand, TF-IDF outperforms Word2Vec in this task, although many works pointed out that Word2Vec is better at capturing semantic attributes. We hypothesize that Word2Vec needs fine-tuning to our dataset before it can be effective.

Table 1: Experiment results of theme tags classification.

Techniques	**Precision**	**Recall**	**F1-score**
LR + TF-IDF	0.652	0.222	0.332
LR + Word2Vec	0.584	0.222	0.280
SVM + TF-IDF	**0.658**	**0.262**	**0.375**
SVM + Word2Vec	0.559	0.180	0.273

Table 2: Experiment results of product tags classification.

Techniques	**Precision**	**Recall**	**F1-score**
LR + TF-IDF	0.9556	0.3443	0.5062
LR + Word2Vec	0.9395	0.2307	0.3704
SVM + TF-IDF	**0.9657**	**0.8692**	**0.9149**
SVM + Word2Vec	0.9269	0.5945	0.7244
Label-GB + TF-IDF	0.8453	0.8397	0.8425
Label-GB + Word2Vec	0.8507	0.8419	0.8463

DISCUSSION

This paper demonstrates the process of finding customer insights from data extraction, preprocessing, feature extraction, and model selection and evaluation. We can see a big difference between Precision and Recall for the model evaluation from the theme tags classification results. And when taking it to find the f1-score, the f1-score is not much higher. It is at a level that may be called low, which is less than 0.5, which means it almost can't classify the articles. This might imply the quality of data or the availability of the models we can choose to apply in this context.

This result can be assumed to be due to the combination of Theme tags with too many categories, which makes a relatively significant difference. It's also not very good at categorizing. For example, balcony and terrace, which, when interpreted in Thai, have the same meaning. But the details are different in English. But this architectural knowledge, the labeling team may not have as much this knowledge as they should.

On the other hand, product tags, which have high precision and recall and a high f1-score, show that if product tags categorize, they can be organized well. However, looking at the data in detail, we find that the product tags are not categorized deeply in the first layer but more detail in the subsequent layers. It must be divided by the main category before going into sub-categories later. It can be said that a large number of sub-categories may not be necessary to consider, which, if taken into account, would have the same effect as the previous theme tags classification, which would be less effective.

This highlights the significant gap in language understanding in humans. Educating the human teams is still relevant and high cost. The tasks that require human hands are still subjective and error-prone. It is not feasible to find the optimum number of categories. Reinforcement or self-reinforcement learning might be able to tackle this problem. However, it is subjected to vocabulary and computation limitations.

Moreover, we use the confusion matrix with Precision, Recall, and F1-Score. The lower the value, the more confusing matrix looks like it's scattered and not concentrated. The following interpretation can complement the understanding of low numbers. It can point down to which category this small value is due to an inaccurate prediction.

CONCLUSION

We presented a case study of formulating the narrow topic analysis problem in understanding customer behavior for creating content in material and product sourcing. We started by defining the problem as tags classification and giving a guideline for determining the standard tags if none exist. The framework we used for text classification consists of data processing, word vectorization, and variety. We compared two different word vectorization methods and three different classification methods and found that SVM combined with TF-IDF gives the best results on our dataset.

Limitation

This study collected data only from one source. It would add more generalizability if data could be collected from various sources. The implications would be more complications due to different characteristics and societal backgrounds. For example, obtaining data from social media platforms like Facebook groups vs. Twitter vs. Instagram might result in various categories. Expanding the data collection scope, of course, require more computing power, storage, and resources.

Future Works

The analysis of tags for finding customer behavior insights is left for future work, along with the experiments on modern classification techniques like deep learning and building on our current project visualization regarding the use of tags as bubble visualization (Chotisarn et al., 2021). Furthermore, data is not always available and ready for analysis. The process might involve data extraction from the source as part of the data acquisition (Prutsachainimmit & Nadee, 2018).

REFERENCES

Agrawal, A., Gans, J., & Goldfarb, A. (2018). Prediction Machines: The Simple Economics of Artificial Intelligence. Harvard Business Press.

Buitinck L., Louppe G., Blondel M., Pedregosa F., Mueller A., Grisel O., Niculae V., Prettenhofer P., Gramfort A., Grobler J., & Layton R. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Chotisarn, N., Lu, J., Ma, L., Xu, J., Meng, L., Lin, B., ... & Chen, W. (2021). Bubble storytelling with automated animation: a Brexit hashtag activism case study. Journal of visualization, 24(1), 101-115.

Chotisarn, N., Pimanmassuriya, W., & Gulyanon, S. (2021). Deep Learning Visualization for Underspecification Analysis in Product Design Matching Model Development. IEEE Access, 9, 108049-108061.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. "O'Reilly Media, Inc."

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In Eleventh Conference on Uncertanty in Artificical Intelligence, Proc., 1995 (pp. 338-345).

Kemp, G., & White, G. (2021). Google Data Studio for Beginners: Start Making Your Data Actionable. Apress.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. ACM Computing Surveys (CSUR), 54(3), 1-40.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., & Vanderplas J. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Phatthiyaphaibun ,W., Chaovavanich, K., Polpanumas, C., Suriyawongkul, A., Lowphansirikul, L., and Chormai, P., "PyThaiNLP: Thai Natural Language Processing in Python," Jun. 2016. [Online]. Available: http://doi.org/10.5281/zenodo.3519354.

Prutsachainimmit, K., & Nadee, W. (2018). Towards data extraction of dynamic content from JavaScript Web applications. 2018 International Conference on Information Networking (ICOIN), 750–754.

Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., & Chinnan, W. (2000, November). Character cluster based thai information retrieval. In Proceedings of the fifth international workshop on on Information retrieval with Asian languages (pp. 75-80).