韓國IT서비스學會誌
第17卷 第2號
2018年 6月, pp.111-128

Journal of Information Technology Services
https://doi.org/10.9716/KITS.2018.17.2.111

Evaluation and Functionality Stems Extraction for App Categorization on Apple iTunes Store by Using Mixed Methods : Data Mining for Categorization Improvement*

Chao Zhang** · Lili Wan***

■ Abstract ■

About 3.9 million apps and 24 primary categories can be approved on Apple iTunes Store. Making accurate categorization can potentially receive many benefits for developers, app stores, and users, such as improving discoverability and receiving long-term revenue. However, current categorization problems may cause usage inefficiency and confusion, especially for cross-attribution, etc. This study focused on evaluating the reliability of app categorization on Apple iTunes Store by using several rounds of inter-rater reliability statistics, locating categorization problems based on Machine Learning, and making more accurate suggestions about representative functionality stems for each primary category. A mixed methods research was performed and total 4905 popular apps were observed. The original categorization was proved to be substantial reliable but need further improvement. The representative functionality stems for each category were identified. This paper may provide some fusion research experience and methodological suggestions in categorization research field and improve app store's categorization in discoverability.

Keyword : Mobile Application, App Categorization, Inter-Rater Reliability, Functionality Stem, Tf-Idf, Machine Learning, Lemmatization, Natural Language Processing

Submitted : May 9, 2018 1st Revision : June 17, 2018 Accepted : June 21, 2018

* This work was supported by Hankuk University of Foreign Studies Research Fund of 2018.

** First Author, College of Business, Hankuk University of Foreign Studies
*** Corresponding Author, College of Business, Hankuk University of Foreign Studies

1. Introduction

Nowadays, mobile applications have been widely developed to assist with our daily usage. Mobile applications are now changing the world and enriching people's different kinds of requirements and lives. The App Stores have grown into a vibrant ecosystem which consists of millions of apps, developers, and billions of users. Until April 2018, more than 3.9 million apps on iOS platform and more than 3.7 million apps on Google Android platform can be approved. Especially, more than 1.52 million available apps can be approved on iOS platform. In order to help users discover new apps to fit their needs, Apple iTunes Store and Google Play established several categories to manage their apps. For example, total twenty-four primary categories are used on Apple iTunes Store and total thirty-three primary categories are used on Google Play. When app designers or publishers want to upload their apps to Apple iTunes Store, they have to assign two categories to their apps which include a primary and a secondary category. Primary category shows the best descriptions of the main function of an app. When users browse the App Store and try to filter search results, primary category can help users discover what they need directly about an app's main function. Generally, there are several methods for users to search and locate a needed app. First, users pick up some keywords based on their searching abilities and experience to locate an app. Second, users can receive some external information (Word-of-Mouth, app advertisement, some marketing promotions, etc.) about an app and search app's name or some related information directly on the App Store. Third, users can browse and observe App Store directly and locate an app by using some internal information, such as using best apps top ranking list, top grossing list, popular app list, new app list, special collections, as well as browsing App Store's categories directly. App designers, for both Apple iTunes Store and Google Play, are required to following developer programs and guiding principles to design and submit their apps to a specific category. All of the apps in different categories should satisfy the design principles, such as Safety, Performance, Business, Design and Legal. Before app designers publish their apps online, assigning an accurate and effective category is particularly important for app's discoverability on the App Store. Category for an app means a map and a direction.

However, some problems about App Store's "Categorization" may cause usage inefficiency and confusion. For example, on Apple iTunes Store, some primary category names are unintelligible which contains several meanings and no further descriptions can be found, such as "Business", "Productivity", "Utilities", or "Reference". Some primary categories may cause cross-attribution problems for some apps, such as primary category "Magazines & Newspapers" and "News." Especially, both Apple iTunes Store and Google Play have no keywords for representative functionalities of different categories. Apple iTunes Store shows the category descriptions and examples as same as Google Play shows examples only on their developer's websites. Users cannot browse such descriptions directly on App Stores. Furthermore, app designers and publishers generally choose primary and secondary categories for their apps at the beginning when they submit them on App Store. That means apps are assigned to categories arbitrarily by developers. Both Apple iTunes Store and Google Play are lack of stricter management regulations for app categorization which may cause inappropriate classification. For example, an app named "Remind : School Communication" on Apple iTunes Store was classified as an educational app that belonged to the Education primary category just because this app supplied a communication platform for students. However, based on the app description and actual using test, the main functions of this app showed significant function characteristics of Social Networking category.

The purpose of this study focused on evaluating the reliability of app categorization on Apple iTunes Store, locating categorization problems, and making more accurate suggestions for representative functionality stems in the form of simple words (hereafter simplified as stems) of each primary category. In this paper, mixed methods research was performed to verify the categorization and related functionality stems. At the first research stage, we investigated the inter-rater reliability of categorization on Apple iTunes Store. Top ten popular apps were selected from each primary category. Total 210 apps from 21 categories were classified again by using Apple iTunes Store's original categorization except category Game, Stickers, and Magezines & Newspapers because of their secondary categories. The inter-rater reliability was used to check and locate the problems of the original categorization on Apple iTunes Store. At the second research stage, a data mining process for identifying the representative functionality stems of each category was performed. As we know, app description offers the most important information about the app and its features, functionality keywords, search terms, etc. It directly affects App Store's optimization process which can optimize the keywords searching results on App Stores. An efficient app description can make the app ranking higher and more discoverable to potential customers (Olabenjo, 2016; Berardi et al., 2015). Therefore, we selected 240 top popular apps for each category of total 21 categories and investigated the frequency of each functionality stems for each category by developing and conducting a mixed research method. After screening, total 4,905 apps were investigated. The functionality stems with high frequency were considered as the representative keywords for each specific category. Further verification process was also performed by using Term-Frequency-Inverse Document Frequency (TF-IDF) method. We performed another round of inter-rater reliability with Cohen kappa to evaluate whether the extracted functionality stems significantly represented the features of each category. The results showed that the extracted results of functionality stems had high probabilities to help app developers categorize their apps in a proper category. And it was also helpful for users to use those representative functionality stems for app searching on Apple iTunes Store.

2. Literature Review

2.1 App Store Categorization

Apple iTunes Store as a closed development platform is the official host for almost four million iOS applications. It plays a very important role in our daily lives for providing a variety of useful tools on iPhone, iPad, iPod touch, Mac, PC, and Apple TV. However, how to search and locate a useful one among millions of applications is a challenging thing. Apple iTunes Store has total twenty-four primary categories. Generally, app designers assign categories for their apps when they start to publish them on Apple iTunes Store. It caused a lot of problems, such as misplacing many applications in the wrong category (Olabenjo, 2016; Zhu et al., 2012; Zhu et al., 2014), or publishing inappropriate app descriptions and features. Based on a case study made in 2017 for Game primary and secondary categories on Google Play, the framework for app categorization (FRAC+) showed 0.35% to 1.10% game apps were detected as miscategorized (Surian et al., 2017). Several prior researchers believed that Game category might be easier for classifying by users or app developers than the other app categories because of its significant enigmatic and distinctive features. Even some classifiers misclassified game applications for their secondary categorization, it may not necessarily mean the app was classified wrongly from the primary categorization perspective (Lulu and Kuflik, 2016; Olabenjo, 2016). Therefore, making accurate categorizing approach according to app functionalities will potentially receive many benefits for developers, app stores, and users, such as improving the discoverability, increasing the using life of apps, improving app feature design by mutual learning, receiving long-term revenue, and revising categorization.

The existing categories for apps on App Stores are arbitrarily assigned by app developers based on their understanding of the functionalities of their apps and the category descriptions on App Stores. Some prior studies have tried to mining textual features which were extracted from app specifications (Gorla et al., 2014; Olabenjo, 2016). Generally, the specifics of a mobile application can be observed on App Stores, such as app name, subtitle, developer name, free or paid, in-app purchases, screenshots, app description, features, update information, ratings and reviews, etc. Such specific information is the main resources for researchers and users to analyze and understand the app's functionality characteristics. However, if the app description was not composed in a proper way required by App Stores, the analyzing process for textual features may have to deal with more complicated problems in data mining process. Some researchers have tried to analyze the categorization by using the comments or identifiers which were extracted from source code (McMillan et al., 2011). However, such methods are also not suitable if the source code cannot be matched by comments or identifiers.

2.2 Inter-Rater Reliability Statistics

Some prior researchers tried to use inter-rater reliability statistics to evaluate the classification methods for mobile applications. Inter-rater reliability refers to the degree of agreement among evaluation raters. In diagnostic tests, researchers hope to examine whether different diagnostic methods have a consistent diagnosis. For example, to evaluate the consistency of the test results of two test methods on the same sample or study object. At this point, the Kappa value can be used as an indicator of the degree of consistency of the evaluation judgment. Practice has proved that it is an indicator that describes consistency more ideally. Therefore, it has been widely used in medicine, sociology and other fields. Normally, the statistical measurement of Cohen's kappa shows the agreement between two raters who each classify N items into C mutually exclusive categories (Fleiss, 2003). The formula for Cohen Kappa is as follows :

$$\textit{Cohen Kappa} = \frac{P_O - P_e}{1 - P_e}$$

where P_O means observed agreement; P_e means chance agreement

$$P_O = \frac{a+d}{n};\quad P_e = \frac{(a+b)(a+c)+(c+d)(b+d)}{n^2}$$

where a, b, c, d represent the agreements between two raters as follows :

Cohen Kappa	Rater 2	
Rater 1	Relevant	Not Relevant
Relevant	a	b
Not Relevant	c	d

Another Kappa analysis is Fleiss' Kappa, which can be applied to the analysis of multi-rater or multi-method evaluation. It has been widely used in the measurement system analysis of attribute data. Fleiss' kappa shows the agreement among several raters.

$$\textit{Fleiss' Kappa} = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}}$$

where $\overline{P}$ means the mean of P_i, the extent to which raters agree for the i-th subject; $1-\overline{P_e}$ means the degree of agreement that is attainable above chance; $\overline{P}-\overline{P_e}$ means the degree of agreement actually achieved above chance.

$$\overline{P} = \frac{1}{Nn(n-1)}\left(\sum_{i=1}^{n}\sum_{j=1}^{k} n_{ij}^2 - Nn\right);\quad \overline{P_e} = \sum_{j=1}^{k} p_j^2$$

Cohen's kappa with more than 0.8 values can be considered as almost perfect agreement between two raters. And for Fleiss' kappa, as same as Cohen's kappa, when its value is higher than 0.8 means almost perfect agreement among several raters. For example, an inter-rater reliability with Cohen's kappa was used to evaluate a classification method for car apps based on data mining results (Zhang et al., 2016, 2017). Moreover, the inter-rater reliability with Fleiss' kappa was also used to evaluate a persuasive design guideline (Zhang et al., 2016).

2.3 Machine Learning with Naïve Bayes Theorem and TF-IDF

2.3.1 Term Frequency-Inverse Document Frequency

One of weighting scheme in text classification is Term Frequency-Inverse Document Frequency (TF-IDF), which emphasized those representative features in a document or emphasized an important word in a given set of documents (Eck et al., 2005). "Term Frequency" (TF) shows the word frequency or the number of occurrences of a word in a specific app. It needs to be standardized as follows in order to facilitate the comparison of different categories.

$$tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{k,j}}$$

Where $n_{i,j}$ means a word frequency in app d_j; $\Sigma_k n_{k,j}$ means the sum of occurrences of all words in app d_j.

Another thing is about the IDF. An important adjustment factor is always needed to measure whether a word is a common word. If a word is rare, but it appears a lot of times in a category, then it is likely to reflect the characteristics of this category, which is exactly what we need. That is, based on the frequency of words, it is necessary to assign an “important weight” to each word. The most common words should be given the smallest weight, the more common words should be given less weight, but the less common words should be given greater weight. It is called “Inverse Document Frequency” (IDF).

$$idf_i = \log\frac{|D|}{1+|\{j:\ n_i \in d_j\}|}$$

Where |D| means the total number of apps in a category.

$|\{j: t_i \in d_j\}|$ means the number of apps where the term n_i appears.

To avoid division-by-zero, it is therefore common to adjust the denominator to

$$1+|\{j: n_i \in d_j\}|$$

If the term frequency (TF) and inverse document frequency “IDF” can be confirmed, multiplying these two values yields the TF-IDE value of a word.

$$tfidf(n,\ d,\ D) = tf(n,\ d) \times idf(n,\ D)$$

The TF-IDF method is more preferable than a regular frequency count of tokens (Baeza-Yates and Ribeiro-Neto, 1999). Generally, it is useful and more accurate to extract feature data. However, it doesn’t always show the representative features for a specific category on app stores. For example, some prior researchers used TF-IDF to extract category features on Google Play. Some features, such as “quickly” and “feature” for Books and Reference category, “dynamics” and “android” for Business category, “offline” for Education category, etc. were completely unable to reflect the characteristics of related categories (Olabenjo, 2016). Therefore, in this study, we tried to combine a regular frequency count method with TF-IDF in order to extract more representative stems for each category.

2.3.2 Machine Learning and Naïve Bayes Theorem

Recently, some researchers tried to use Machine Learning in product classification or product promotion (Kreyenhagen et al., 2014), as well as the analysis of categorization on App Stores. Numerous prior studies also focused on improving categorization in health subjects and sentiment analysis. For example, Support Vector Machine (SVM) machine learning algorithm was used to classify products with fashion brands (Kreyenhagen et al., 2014) or classify patients with mental diseases such as schizophrenia (Schnack et al., 2014). Another example is about using Naïve Bayes algorithm to do sentiment analysis for polarity distribution (Nithya and Maheswari, 2014). Some researchers tried to use Multinomial Naïve Bayes Algorithm, Bernoulli Naïve Bayes Algorithm, and API analysis to analyze the categorization for Google Play according to some open resources (Olabenjo, 2016).

Bayes theorem describes the probability of an event based on the prior knowledge of conditions relating to the event. Generally, this theorem requires large-scale calculations in order to highlight a better effect. It has been widely used in many computer application fields, such as natural language spelling, machine learning, data mining, recommendation systems, image recognition, game theory, etc. Bayes theorem is defined mathematically as follows :

$$P(A\,|\,B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are events and $P(B) \neq 0$

- P(A) is the prior probability of A, P(B) is the prior probability of B
- P(A|B) is the probability of A occurring under the condition that B occurs.
- P(B|A) is the posterior probability of B due to the value of A.
- P(B|A)/P(B) refers to as standardized likelihood.

The conditional probability is the probability of occurrence of event A under the condition that another event B has occurred. The joint probability represents the probability that two events occur together. P(A|B) is the conditional probability that A is known after B occurs, and is also called the posterior probability of A because of the value obtained from B. P(B|A) is the conditional probability of B after the occurrence of A, and it is also called the posterior probability of B due to the value of A. The basic method of Bayes theorem can be described as follows : based on statistical data and some features, the probability of each category can be calculated in order to realize categorization.

If P(B) is replaced, then Bayes formula can be described as follows :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\neg A)P(\neg A)}$$

where P($\neg A$) means not A.

Actually, Bayes formula wants to describe how much the evidence can be trusted.

As we know, the Naïve Bayes Theorem can be used to deal with many problems, such as natural language processing (NLP), spam mail detection, personal email sorting, pornographic content detection, etc. although the assumption that "all features are independent of each other" is unlikely to be established in reality, it can greatly simplify calculations and have little effectiveness on the accuracy of categorization results. In order to generate a classifier, we considered those functionality stems as tokens and classify them into a particular class. In Naïve Bayes algorithm, the maximum a posteriori probability (MAP) needs to be calculated because MAP is an estimate of an unknown quantity. It equals to the mode of the posterior distribution. It can be used to estimate the prior distribution which is different from maximum likelihood estimation (MLE). That means MAP requires maximum for both MLE and prior probability.

$$\theta_{ML}(x) = arg \overset{\max}{\theta} f(x|\theta)$$

$$\theta_{MAP}(x) = arg \overset{\max}{\theta} f(x|\theta)g(\theta)$$

The Naïve Bayes Theorem is widely used in Machine Learning field which can be used to deal with natural language processing (NLP), such as spam mail detection (Yang et al., 2006), personal email sorting, ranking (Zhang and Sheng, 2004), text or review classification (Maalej and Nabil, 2015; Islam, 2014), pornographic content detection, etc. Naïve Bayes Theorem has a basic assumption, that is, the features of the data are conditionally independent. Based on this theorem, the Naïve Bayes Machine Learning Model was popular used for bag-of-words, Information Extraction (Freitag, 2000) and Information Extraction with smoothing method (Gu and Cercone, 2006), and for checking the Naïve Bayes algorithms' performance in classifying data (AI-Aidaroo et al., 2012; Qiang, 2010). Generally, Naïve Bayes Model can be used to solve numerous problems. For textual categorization, it is also called multivariate Bernoulli event model. The Multinomial Event Model is an improved Naïve Bayes Probabilistic model which is specifically designed for textual classification (Qiang, 2010).

3. Research Approach

The purpose of this paper focused on evaluating the original categorization on Apple iTunes Store to discover classification problems in order to make categorization improvement as well as more accurate suggestions for representative functionality stems of each primary category. A mixed methods research was conducted which consisted of two studies and four research stages.

3.1 Stage-1 : Evaluation for Original Categorization on Apple iTunes Store

The first research stage focused on evaluating the original categorization on Apple iTunes Store. We selected the top ten popular apps from each category. Total 210 apps from 21 primary categories were randomly selected except category Game, Magazines & Newspapers, and Stickers because of their secondary categories. Fleiss' Kappa with five raters (m = 5) was performed for total 210 apps (n = 210) of 21 categories marked as 24 results (k = 24, total 24 results consist of 21 categories, plus two primary category for Game and Magazine & Newspaper, and one "I don't know") to evaluate the original categorization on Apple iTunes Store.

The evaluation team consisted of two experts on mobile application design and three MIS major students with at least two years experience as assistant researcher on app design or MIS system design. In the first round of evaluation, five raters were asked to read and understand the categorization descriptions on Apple's developer website before they started to classify those selected apps. Moreover, they were asked to open each designated app's website and take each app's name, descriptions, and features as reference to classify it into which category it should be assigned. In the evaluation process, each rater's evaluation process was performed separately. By comparing the evaluation results and the app's original categorization, the inter-rater reliability for three raters with Fleiss' kappa can be identified.

The programming language "Python" was used to deal with the evaluation process for Fleiss' Kappa. Python is an interpreted high-level programming language which has been widely used in machine learning or data mining process.

3.2 Stage-2 : App Data Collection and Preparation for Data Mining Process

The second research stage was about app data collection process from Apple iTunes Store. The description of an app plays a very important role for its discoverability and categorization. Generally a successful app or a branded app has a higher profile based on its name, description, and the category, no matter whether it has other information or factors. Based on the official ranking list for popular apps on April and May 2018, limited by less open sources, we manually collected total 5,460 app's specifics from 21 primary categories. After screening, 4,905 apps left and their specifics were saved as a CSV file which consisted of several attributes, such as :

- App Name
- Subtitle
- Description
- Update information

Although some prior researchers believed that rating and review attributes and comments attributes contained keywords information about the relevant app, we ignored such information which was not composed directly by app developers or App Store. Another reason is that numerous app remarks or comments are subjective and have too much irrelevant information. As we mentioned before, some researchers also argued about its accuracy used in categorization process. The dataset was used to build up a classifier for our categorization improvement process.

3.3 Stage-3 : Data Mining for Functionality Stems Extraction

3.3.1 Stems Extraction by Mixed Method

In term frequency examination process, there are several methods to calculate term frequency and extract stems, such as using some commands in Microsoft Word or Excel, using some words frequency calculator programs, or using cloud computing. If we use those traditional methods, the amount of calculation is too large and the data extracting process are very difficult. However, traditional methods sometimes can be more reasonable especially for describing representative functionality stems for a classification. As we have mentioned before, some keywords for a category are totally irrelevant with its representative meanings, such as “dynamics” and “android” for Business category, etc. The purpose of the extracting process was to find out some stems that may represent the features of each category. In English morphemes, English words can be divided into simple words and composite words. Composite word also includes compound words and derivative words. Therefore, we tried to introduce the intersection and union theories of Algebra of sets to extract the representative stems for each category as well as their frequencies. It can be represented as a very simple law as follows.

$$(A \cap B) \in (A \cup B) \in U$$

where A and B means two sets, U means a universe.

We tried to develop several methods. The first method was a regular statistical method that we searched some simple words manually to locate all of the other composite words consisted of those simple words. In this word union, some representative stems can be observed by comparing their frequencies. For example, in Book category, all of the words that included “book” will be collected as a union. This “book” stem union may consist of many book-related composite words. We analyzed and screened those words in order to find the stems in the form of simple words closest to the features of Book category. A reasonable and accurate result may become the advantage of this method, but the analyzing time period will be very long.

The second method was to use computer programming. By making a Python program for intersection and union, it could be very easy and quick to locate and form the key simple words and those related words as a union. By using machine learning language, the intersection and union process for extracting words was called

Stemming and lemmatization. It belongs to natural language processing (NLP). Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form (a written word form). Lemmatization in linguistics is the process of grouping tog ether the different inflected forms of a word so they can be analyzed as a single item. Based on greedy strategy design, MaxMatch algorithm is often used to segment words in natural language processing. However, the disadvantage existed in the stems extracting process. It was hard for computers to associate the most appropriate stems with the category features. Therefore, a mixed method combined with two methods was conducted and we extracted some high-frequency stems except stop words by following the frequency ranking list.

3.3.2 Verifying Stems Extraction Results by TF-IDF

Automatic Keyphrase Extraction involves many computer frontier fields, such as data mining, text processing, information retrieval, etc. As a very simple and classic algorithm, Term-Frequency-Inverse Document Frequency (TF-IDF) algorithm can also give quite satisfactory results. Generally, if a word is important in a category, it should appear at least more than once by ignoring those "stop words" that are not helpful in finding results and must be filtered out, such as "is", "in", "the", etc.

$$\omega_n = TF_n \times \log(IDF_n)$$

The higher the importance of a word to a category is, the greater its TF-IDF value should be. So, the first few words should be the functionality stems of this category.

In this study, by using mixed method, the features of each category were extracted. Based on TF-IDF, a data mining processing for verifying the representative functionality stems of each category was conducted. And further inter-rater reliability with Cohen kappa evaluation process was conducted to make sure those stems represented the functionality descriptions of each category. Two app design experts were asked to classify those stems into different categories.

4. Data Analysis and Result

4.1 Fleiss' Kappa for Original Categorization

In the first research stage, we performed the first round of inter-rater reliability with Fleiss' kappa and evaluated the original categorization on Apple iTunes Store. Fleiss' Kappa with five raters (m = 5) was performed to classify total 210 apps (n = 210) into 24 categories. A calculating program on Python 2.7 with a library of NumPy 1.11.1 was conducted for Fleiss' kappa calculation. We picked up the main part of the program for calculating on Python and showed as follows.

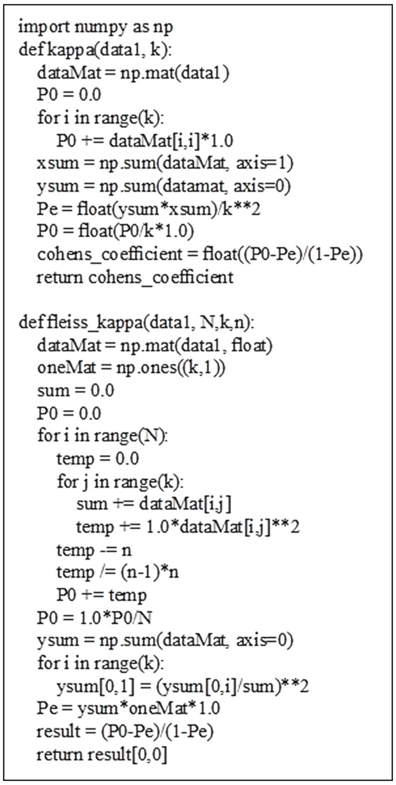
$$\text{Fless' Kappa} = = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}}$$

$$\overline{P} = \frac{1}{Nn(n-1)}\left(\sum_{i=1}^{n}\sum_{j=1}^{k} n_{ij}^2 - Nn\right)$$

$$\overline{P_e} = \sum_{j=1}^{k} p_j^2$$

<Table 1> Summarized Results

Results of Fleiss' Kappa	
Kappa	0.70079
S.E	0.00571
Z	122.697
P-value	0.00000
Lower	0.68960
Upper	0.71198



```
import numpy as np
def kappa(data1, k):
    dataMat = np.mat(data1)
    P0 = 0.0
    for i in range(k):
        P0 += dataMat[i,i]*1.0
    xsum = np.sum(dataMat, axis=1)
    ysum = np.sum(datamat, axis=0)
    Pe = float(ysum*xsum)/k**2
    P0 = float(P0/k*1.0)
    cohens_coefficient = float((P0-Pe)/(1-Pe))
    return cohens_coefficient

def fleiss_kappa(data1, N,k,n):
    dataMat = np.mat(data1, float)
    oneMat = np.ones((k,1))
    sum = 0.0
    P0 = 0.0
    for i in range(N):
        temp = 0.0
        for j in range(k):
            sum += dataMat[i,j]
            temp += 1.0*dataMat[i,j]**2
        temp -= n
        temp /= (n-1)*n
        P0 += temp
    P0 = 1.0*P0/N
    ysum = np.sum(dataMat, axis=0)
    for i in range(k):
        ysum[0,1] = (ysum[0,i]/sum)**2
    Pe = ysum*oneMat*1.0
    result = (P0-Pe)/(1-Pe)
    return result[0,0]
```

<Figure 1> Fleiss' kappa Calculation Process on Python

The result was collated and shown in <Table 1>. It showed that the Fleiss' kappa value was 0.70079. The p-value (P-value = 0.000) of z-test (Z-Statistics = 122.697, Standard Error = 0.00571, alpha = 0.05, two tails) was significant which represented the value of kappa as not equal to zero. The population kappa's confidence interval was between 0.68960 (Lower) and 0.71198 (Upper). The interpreting Fleiss' Kappa value was 0.71198 which represented the strength of agreement as "Good" (Altman, 1991) or "Substantial agreement" (Landis and Koch, 1977). Social scientists commonly relied on data with reliabilities when α ≥ .800, considered data with 0.800 > α ≥ 0.667 only to draw tentative conclusions, and discarded data whose agreement measures α < 0.667. Based on the results, it was concluded that the original categorization on Apple iTunes Store could be considered as substantial reliable but not perfect.

4.2 Stems Extracting Process by Mixed Method and TF-IDF

At the first round, we used a frequency checking software to calculate the term frequency for app's attributes (such as name, subtitle, description, feature, update information) from each category. After screening some apps with non-English descriptions, the total numbers of evaluated apps in each category and top three frequency stems in the form of simple words that may represent the category features were extracted.

In order to verify the accuracy of results, TF-IDF method was also performed. Based on TF-IDF formula, we developed a computer program under Python 2.7 environment with Scikit-Learn package 0.19.1 to extract stems from total 4905 apps. Furthermore, library Count Vectorizer and library Tfidf Transformer were used for the calculation of words segmentation weight. Those stems in the form of simple words should represent the main features of each category.

The advantages of the TF-IDF algorithm are simple and rapid, and the results are more in line with the actual situation. The disadvantage is that simply measuring the word frequency by word frequency is not enough. Sometimes important words may not appear many times. That is, for some important words that represent the main features of a classification may not be extracted by TF-IDF. Moreover, this algorithm cannot reflect the position of the word. Therefore, in this study, TF-IDF was only considered as a test tool or test process for the mixed method. The results of top three stems were collated and shown in <Figure 2>, <Figure 3> (part one and two).

Primary Category	Number of Apps	Total Words	Top Three Stems with Frequency and Leaf	
			“Stem,” (frequency)	“Leaf“ or keywords example”
Books	192	54,605	“book,” (1037)	“eBook,” “bookmark,” “bookshelf”
			“read,” (690)	“reader,” “good reads,” “read”
			“story,” (166)	“storybook,” “history,” “storytelling”
Business	237	65,761	“business,” (279)	“business quality,” “business”
			“job“ (226)	“job,” “jobsite”
			“work“ (350)	“co-worker,” “workspace,” “workforce”
Education	239	81,200	“learn,” (690)	“learner,” “learning”
			“kids,” (420)	“kids academy,” “kids creen,” “kids cafe”
			“student,” (261)	“students,”
Entertainment	239	61,281	“movie,” (499)	“movie,” “movie”
			“TV,” (528)	“MTV,” “HGTV,” “TV”
			“show,” (372)	“show,” “showtime”
Finance	243	71,184	“account,” (334)	“accounting,” “accuont”
			“money,” (503)	“money,” “moneycard,” “papermoney”
			“bank,” (706)	“banking,” “firstbank,” “usbank”
Food & Drink	227	49,295	“food,” (327)	“food”
			“meal,” (218)	“meals,” “mealboard,” “mealtime”
			“restaurant,” (295)	“restaurant”
Health & Fitness	237	95,401	“fitness,” (471)	“fitness”
			“weight,” (391)	“bodyweight,” “weight-loss,” “overweight”
			“health,” (770)	“healthier,” “healthkit,” “healthcare”
Lifestyle	237	65,840	“appartment,” (135)	“apartment”
			“life,” (114)	“lifestyle,” “reallife,” “lifetime”
			“home,” (433)	“home,” “townhome,” “homegoods”
Medical	240	76,366	“medical,” (405)	“medical”
			“health,” (596)	“healthcare,” “healthy,” “healthier”
			“drug,” (245)	“drugguide,” “drugstore,” “drug”
Music	238	72,950	“mucis,” (1159)	“musical,” “music,” “musician”
			“song,” (630)	“song,” “songbook”
			“radio,” (263)	“radio”

<Figure 2> Top Three Stems for Each Category _ Part 1

<Table 2> Summarized Results

Results of Cohen Kappa	
Kappa	0.900
S.E	0.039
N	63
P-value	0.00000

The result of Cohen kappa was shown in <Table 2>. And further inter-rater reliability with Cohen kappa showed that the inter-rater reliability between two experts was almost perfect agreement, which means those stems significantly represented the functionality features of relevant categories.

Primary Category	Number of Apps	Total Words	Top Three Stems with Frequency and Leaf	
			“Stem,” (frequency)	“Leaf“ or keywords example
Navigation	228	70,700	“map,” (813)	“map,” “streenmap”
			GPS, (353)	“GPS”
			“location,” (345)	“locate,” “location”
News	234	55,737	“news,” (1227)	“newscasts,” “newspaper”
			“live,” (324)	“live,” “alive,” “deliever”
			“subscription,” (316)	“subscription,” “subscriber”
Photo & Video	238	74,632	“photo,” (1740)	“photoshop,” “photography,” “photo”
			“video,” (1013)	“video”
			“camera,” (316)	“camera,” “camera 360”
Productivity	240	81,018	“automatic,” (193)	“auto,” “automatic360
			“note,” (433)	“note,” “keynote,” “notebook”
			“device,” (319)	“device”
Reference	215	55,299	“bible,” (467)	“bible,” “biblebook”
			“translate,” (225)	“translate,” “translation,” “translator”
			“dictionary,” (120)	“dictionary”
Shopping	239	56,688	“shopping,” (317)	“shopping”
			“product,” (236)	“product”
			“store,” (472)	“store,” “drugstore,” “onlinestore”
Social Networking	233	63,287	“friend,” (442)	“friend,” “friendship,” “friendly”
			“message,” (294)	“messenger,” “iMessage,” “InMessage”
			“chat,” (402)	“snapchat,” “chatting,” “chat-to-meet”
Sports	239	65,758	“sport,” (520)	“sportsgame,” “foxsports,” “sportsengine”
			“ball,” (445)	“football,” “baseball,” “basketball”
			“league,” (275)	“leaguemate,” “colleague”
Travel	237	65,942	“travel,” (531)	“traveler,” “traveling,”
			“trip,” (326)	“trip,” “tripadvisor”
			“hotel,” (365)	“hotel”
Utility	238	59,721	“device,” (264)	“device”
			“calls,” (414)	“call,” “calling,” “caller”
			“support,” (200)	“support”
Weather	231	56,530	“weather,” (1785)	“weather”
			“temperature,” (212)	“temperature”
			“forecast,” (609)	“forecast”

<Figure 3> Top Three Stems for Each Category _ Part 2

5. Discussion

5.1 Discussion for Original Categorization on Apple iTunes Store

Based on the result of inter-rater reliability with Fleiss' kappa, the original categorization on Apple iTunes Store was considered as sustainable accepted but not perfect. By observing five rater's decision-making process, several problems about original categorization on Apple iTunes Store were discovered.

First, we selected top ten apps from Food & Drink category which should have represented the category features of Food and Drink. However, nine of ten apps were classified as shopping category by five raters with significant agreement. We observed those nine apps and we found all of them were about food ordering, online payment, and delivery. Only one app was consistent with people's perception of food and drink category in introducing related information about foods and drinks, restaurants, cooking knowledge, etc. Therefore, creating a secondary category for food delivery apps under the primary category of Shopping or Food & Drink may increase their discoverability.

Second, Medical category and Health & Fitness category have a significant correlation. Especially, for word "Health" and "Medical", it is hard for all of five raters to perceive difference when they evaluated those apps for two categories. Only those apps with significant professional information about medical treatment, curing, and recovery treatment were clearly identified as Medical category. Therefore, merging the Medical category and Health & Fitness Category into one primary category, or setting up the Health and the Fitness category as two secondary categories of the primary Medical category would be a better choice.

Third, category News and category Newspaper & Magazine also have a significant correlation.

By observing the top ten apps, most of those branded television broadcasting company have apps from both of two categories. Therefore, our suggestion is to merge them into one category.

Fourth, numerous apps related to transportation, such as flight, airline company, subway or subway map, train, train tickets, etc. were not in appropriate categories and they were not classified separately, especially for those car-related mobile apps. A prior study has classified those car-related mobile apps into eight different categories, and the classification method has been proved to be reliable (Zhang et al., 2017). Establishing a new primary category for transportation may be urgently needed.

5.2 Discussion for Representative Functionality Stems

Based on the extracted results, some "stems" in the form of simple words or some "keywords" were extracted and verified. In <Table 2>, the frequency of each stems or keywords were also listed. It should be noted that the size of the word frequency is not an important factor, What's important is the degree of closeness between the word frequency and the representative features of its category. By comparing the extracted word frequency list and the TF-IDF results, some stems in the form of simple words or some keywords were selected as an index for categorization on Apple iTunes Store. The further inter-rater reliability with Cohen kappa showed a significant almost perfect agreement (Cohen kappa = 0.900) that identified those stems or keywords were strongly relevant to the category features. However, for category Productivity, Utility, Lifestyle, and Reference, it was really hard to extract a stem or a word which can perfectly explain the category features. Actually, for the primary names of those four categories, raters and users may feel very confused about their representative features. Especially, in terms of their names, the meanings represented by these nouns are really hard for understanding as well as measuring or quantifying. Therefore, all of those four categories need to be improved, segmented or replaced in order to have understandable meanings.

Furthermore, the evaluation method needs further verified. Recently, with the development of machine learning and AI technology, more useful programming libraries or tools were developed. Some methods or models, such as Latent Semantic Analysis (LSA) can also be used to extract the stems which are related to the category features. By using Python Latent Semantic Indexing (LSI), representative stems or keywords may be extracted in a more effective way. By developing and using such programs in extracting and verifying our results, more accurate and useful stems may be discovered and identified for future categorization improvement.

6. Conclusion and Future Research

The purpose of this paper focused on using inter-rater reliability with Fleiss' kappa to evaluate original categorization on Apple iTunes Stores. Although many problems have been found during the evaluation process, such as app misclassification, no significant description or definition for some primary categories, and serious cross-attribution for some app's classification, the agreement of five raters for classifying total 210 apps into 24 categories still remained to be substantial reliable. Further stems extracting process showed that every three representative stems were chosen for total 21 categories and inter-rater reliability with Cohen kappa shows a significant reliable. In this paper, for every research stage, we developed at least two methods to mutually verify the consistency of the results. In the first research stage, we used statistics methods to calculate the Fleiss' kappa, which was also verified by a developed Python program from machine learning perspective. In the second research stage, we tried to develop a mixed method which combined a regular words extracting method and a Natural Language Processing method by using machine learning technology to extract representative stems and keywords for each category on Apple iTunes Store. In the following process, we conducted TF-IDF to verify our extracted results. Because we tried to extract some functionality key stems or keywords that may not represent the highest term frequency, a further inter-rater reliability with Cohen kappa was conducted and performed again to verify the reliability of our mixed method developed in the second stage. We tried to avoid the disadvantages from separately using any of those methods in order to discover more accurate and reliable results. This paper may be able to provide some fusion research experience and methodological suggestions in categorization research field. Furthermore, the results of this study may be used to help current App Stores improve their categorization and help app developers increase the discoverability of their apps on App Stores. In our following research, we are trying to develop Python programs for Multinomial Naïve Bayes Algorithm and Latent Semantic Analysis. And we are trying to combine machine learning or AI technology with social science research methods so that we can find a better way to improve our research in the near future.

References

Al-Aidaroo, K.M., A.A. Bakar, and Z. Othman, "Medical Data Classification with Naive Bayes Approach", *Information Technology Journal*, Vol.11, No.9, 2012, 1166-1174.

Gorla, A., I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions", *in Proc. Int. Conf. Softw. Eng.*, 2014, 1025-1035.

Baeza-Yates, R. and B. Ribeiro-Neto, "Modern information retrieval", *Packt Publishing Ltd*, Vol.9, 1999.

Kreyenhagen, C.D., T.I. Aleshin, J.E. Bouchard, A.M.I. Wise, and R.K. Zalegowski, "Using supervised learning to classify clothing brand styles", *in 2014 Systems and Information Engineering Design Symposium (SIEDS),* Charlottesville, VA, USA : IEEE, apr 2014, 239-243.

Eck, M., S. Vogel, and A. Waibel, "Low Cost Portability for Statistical Machine Translation based on N-gram Frequency and TF-IDF", *in International Workshop on Spoken Language Translation, IWSLT 2005, Pittsburgh*, PA, USA, 2005, 61-67.

Fleiss, J.L., B. Levin, and M.C. Paik, Statistical methods for rates and proportions, New Jersey : John Wiley & Sons Hoboken, 2003.

Freitag, D., "Machine learning for information extraction in informal domains", *Machine Learning*, Vol.39, No.2-3, 2000, 169-202.

Berardi, G., A. Esuli, T. Fagni, and F. Sebastiani, "Multi-store metadata-based supervised mobile app classification", *in Proc. 30th Annu. ACM Symp. Appl. Comput.*, 2015, 585-588.

Guo, Q., "An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification", *in 2010 Second International Conference on Computer Research and Development*, Kuala Lumpur, Malaysia : IEEE, No.1, 2010, 699-701.

Gu, Z. and N. Cercone, "Naive Bayes Modeling with Proper Smoothing for Information Extraction", *in 2006 IEEE International Conference on Fuzzy Systems. Vancouver*, BC, Canada : IEEE, 2006, 393-400.

Islam, M.R., "Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews", *in 2014 International Conference on Electrical Engineering and Information & Communication Technology*, Dhaka, Bangladesh : IEEE, Apr 2014, 1-4.

Lulu, L.B. and T. Kuflik, "Wise mobile icons organization : Apps taxonomy classification using functionality mining to ease apps finding", *Mobile Inf. Syst.*, 2016.

Maalej, W. and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews", *in 2015 IEEE 23rd International Requirements Engineering Conference (RE)*, Ottawa, ON : IEEE, aug 2015, 116-125.

McMillan, C., M.L. Vásquez, D. Poshyvanyk, and M. Grechanik, "Categorizing Software Applications for Maintenance", *27th IEEE International Conference on Software Maintenance (ICSM)*, 2011.

Nithya, R. and D. Maheswari, "Sentiment Analysis on Unstructured Review", *in 2014 International Conference on Intelligent Computing Applications*, Coimbatore, India : IEEE, mar 2014, 367-371.

Olabenjo, B., "Applying Naïve Bayes Classification to Google Play Apps Categorization", arXiv : 1608.08574v1[cs.LG], 30 Aug, 2016.

Surian, D., S. Seneviratne, A. Seneviratne, and S. Chawla, "App Miscategorization Detection : A Case Study on Google Play", *IEEE Transactions on Knowledge and Data Engineering*, Vol.29, No.8, 2017, 1591-1604.

Yang, Z., X. Nie, and W. Xu, "An Approach to Spam Detection by Naive Bayes Ensemble Based on Decision Induction", *in Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China : IEEE, Vol.2, 2006, 861-866.

Zhang C., L. Wan, and D. Min, "A Classification of Car-related Mobile Apps : For App Development from a Convergence Perspective", *Journal of Digital Convergence*, Vol. 15, No.3, 2017, 77-86.

Zhang, C., L. Wan, and D. Min, "Car App's Persuasive Design Principles and Behavior Change", *Proceedings of the International Conference on Internet Technologies and Society* 2016, ITS 2016 Melbourne, Australia, 2016, 73-82.

Zhang, C., L. Wan, and D. Min, "Persuasive Design Principles of Car Apps", Proceedings of Business Information Systems, 19th International Conference in BIS 2016, Leipzig, Germany, 2016, 397-410.

Zhu, H., E. Chen, H. Xiong, H. Cao, and J. Tian, "Exploiting enriched contextual information for mobile app classification", *in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, 1617-1621.

Zhu, H., E. Chen, H. Xiong, H. Cao, and J. Tian, "Mobile app classification with enriched contextual information", *IEEE Trans. Mobile Comput.*, Vol.13, No.7, 2014, 1550-1563.

Zhang, H. and S. Sheng, "Learning Weighted Naive Bayes with Accurate Ranking", *in Fourth IEEE International Conference on Data Mining (ICDM'04)*, IEEE, 2004, 567-570.

About the Authors

Chao Zhang (alan.zhangchao@gmail.com)

Chao Zhang is a professor of College of Business at Hankuk University of Foreign Studies. He received his Ph.D. in E-Commerce System from Korea University (KU) in 2016. His current research interests include Human-Computer Interaction (HCI), behavioral psychology, individual information cognitive processing, data mining, in-car application design, and eye movement in system design.

Lili Wan (shelleyone@naver.com)

Lili Wan is a professor of College of Business at Hankuk University of Foreign Studies. She received her Ph.D. in E-Business from Korea University (KU) in 2012. Her current research interests include Human-Computer Interaction (HCI), behavioral psychology, Experience Design, data mining, persuasive technology, and wearable system design.