A Bayesian Sampling Method for Product Feature Extraction from Large Scale Textual Data

Sunghoon Lim
Industrial and Manufacturing Engineering,
The Pennsylvania State University,
University Park, PA 16802
e-mail: slim@psu.edu

Conrad S. Tucker [1]
Mem. ASME
Engineering Design and
Industrial and Manufacturing Engineering,
The Pennsylvania State University,
University Park, PA 16802
e-mail: ctucker4@psu.edu

ABSTRACT

The authors of this work propose an algorithm that determines optimal search keyword combinations for querying online product data sources in order to minimize identification errors during the product feature extraction process. Data-driven product design methodologies based on acquiring and mining online product-feature-related data are faced with two fundamental challenges: 1) determining optimal search keywords that result in relevant product related data being returned and 2) determining how many search keywords are sufficient to minimize identification errors during the product feature extraction process. These challenges exist because online data, which is primarily textual in nature, may violate several statistical assumptions relating to the independence and identical distribution of samples relating to a query. Existing design methodologies have predetermined search terms that are used to acquire textual data online, which makes the resulting data acquired, a function of the quality of the search term(s) themselves. Furthermore, the lack of independence and identical distribution of text data from online sources, impacts the quality of the acquired data. For example, a designer may search for a product feature using the term "screen", which may return relevant results such as "the screen size is just perfect", but may also contain irrelevant noise such as "researchers should really screen for this type of error". A text mining algorithm is introduced to determine the optimal terms without labeled training data that would maximize the veracity of the data acquired to make a valid conclusion. A case study involving real-world smartphones is used to validate the proposed methodology.

[1] Corresponding author

Keywords: product design, product feature extraction, information retrieval, online, Bayesian, text mining, training data, non-i.i.d., keyword

1 INTRODUCTION

Recently, online data has become widely used for knowledge discovery across a wide range of fields. As a low-cost and real-time information source, the role of online data in product design has become especially significant in recent years [1-6]. Online social media platforms and customer review sites have been considered a possible source of information for product design due to 1) the ease of posting comments/opinions (from a customer's perspective), 2) acquiring customers' feedback (from a designer's perspective) and 3) the size and heterogeneity of the data.

However, data-driven product design methodologies based on mining online data, may have high identification errors of product-feature-related information due to noise resulting from the differences between writing formats or because of constraints placed by online media platforms. For instance, *Twitter* has a 140-character limit. Nevertheless, existing studies on data-driven product design methodologies that utilize online data overlook the task of 1) selecting optimal search keywords and 2) determining the optimal number of search keywords needed to efficiently identify product-feature-related information. These studies could therefore face several challenges due to *a term disambiguation problem* and *a keyword recognition problem* [7], as defined below:

Term Disambiguation Problem: Messages containing highly discriminative product-feature-related words used in a non-product-feature-related context are classified as product-feature-related. A term disambiguation problem is synonymous to a false positive or a type I error. Table 1 provides an example of a term disambiguation problem. The *tweet* is not related to product features, but it can be misclassified as product-feature-related, since it contains a highly discriminative product-feature-related term ("screen").

Table 1 An example of a term disambiguation problem

tweet
*(ID: Ka*******)* researchers should really **screen** for this type of error

Keyword Recognition Problem: Messages containing highly discriminative product-feature-related words are classified as non-product-feature-related. A keyword recognition problem is synonymous to a false negative or a type II error. Table 2 provides an example of a keyword recognition problem. The *tweet* contains information about product features (the *iPhone*'s battery life). However, it can be misclassified as non-product-feature-related, since it contains the terms that can be classified as criminal-related such as "case," "battery," and "dying."

Table 2 An example of a keyword recognition problem

tweet
*(ID: Ma****)* gat…just as this court case is about to start, my iphone **battery** is dying…

The authors of this work propose an algorithm that determines optimal search keyword combinations without labeled training data. Designers aiming to utilize online product-related data will therefore be able to acquire and mine relevant product-feature-related data, while minimize noise and time challenges. This model could prove useful to product feature extraction methods that predict consumers' responses to new products by utilizing large-scale product feature data found on customer review or social media sites. In addition, the proposed methodology can also be useful for identifying online product surveys that have actual product-feature-related information or feedback. This work demonstrates how the proposed methodology can reduce product-feature-related information identification errors in non-i.i.d. (independent and identically distributed) online media data and increase significant product-feature-related knowledge. A case study involving real-world smartphones (*iPhone 4* and *iPhone 4S*) demonstrates the proposed methodology.

The rest of the paper is organized as follows. This section provides an introduction and motivation to the research. Section 2 provides the literature review for this work. Section 3 explains our proposed Bayesian-based methodology, which identifies optimal search keyword combinations in detail. Section 4 introduces an application, and Section 5 provides the experimental results and discussion. Section 6 concludes the paper.

2 LITERATURE REVIEW

The literature review section presents works related to 1) information retrieval in online media, 2) product feature extraction, and 3) sample size determination and non-i.i.d. approaches.

2.1 Information Retrieval in Online Media

The major differences between a traditional document and a short message found in online media are language formality and content length [7]. Therefore, data mining algorithms focused on mining short messages differ from traditional mining algorithms due to their ability to handle online media data containing lower data dimensions and higher variability in language structure. Several researchers have proposed mining messages in online media for domain-specific problems. For example, Phan et al. present a method based on latent topic analysis models and machine learning methods to classify short and sparse Web segments from large-scale data collections [8]. Hu et al. propose a methodology to improve the performance of short text clustering by using both the internal semantics from the original text and the external semantics from world knowledge, such as *WordNet* [9].

Two methods have been widely used to identify necessary information in online media: 1) keyword-based methods and 2) machine-learning-based methods. However, these two methods are limited in their application to data-driven product design using online media data due to the fact that 1) keyword-based methods use predetermined keywords to return product related data, hereby assuming that the domain expert always knows the entire set of words that accurately describe a product/product feature and 2) machine-learning-based methods require labeled training data, which typically is a manually intensive and costly process.

2.1.1 Keyword-based Methods for Information Retrieval. Keyword-based methods require a dictionary that contains the related words. A document is identified as *related* if it contains one or more predetermined keywords [7]. Several researchers have exploited keyword-based methods with online media in various fields. Ginsberg et al. demonstrate how a regression model classifies the query logs by detecting the presence of keywords implemented in a *Google*-based service [10]. Culotta shows that *Twitter* data yield a better prediction of the actual flu rates than query logs and proposes a method to correlate numerous flu-related *tweets*, identified by flu-related keyword detection, with the actual influenza-like-illness rates [11]. Glier et al. present a filtering method for biological keyword search results by using text mining algorithms to automatically identify which results are likely to be useful to engineering designers [12].

While previous research in keyword-based methods covers information retrieval in online media based on predetermined keywords, considerations of a method that can select optimal search keywords are limited. In this work, the proposed methodology determines optimal search keywords in consideration of the results returned.

2.1.2 Machine-learning-based Methods for Classifying Textual Data with Training Data. Machine-learning-based approaches for text classification train a learner with a collection of labeled documents and then use the trained learner to classify unlabeled documents [7]. Several machine-learning-based methods with training data have been proposed in various fields. Aramaki et al. propose a support-vector-machine-based classifier, trained with unigrams collected within the same proximity of flu-related keywords, which detect flu-related *tweets* [13]. Paul and Dredze propose a machine-learning-based classification algorithm used for identifying *tweets* associated with necessary information [14]. Stone and Choi propose a support-vector-machine-based approach for sentiment classification of *tweets* according to product attributes and attribute levels [15]. Tuarob et al. propose combinational machine learning techniques to identify health-related information with large-scale social media data [7]. Fuge et al. investigate different machine learning algorithms (content-based filtering, collaborative filtering, and hybrid models) in order to gather information about the product design's user requirements [16].

Machine-learning-based methods assume that researchers have labeled training data and then identify the necessary information based on existing training data. However, in numerous cases, it is difficult to identify the necessary information with labeled training data in online media networks. This difficulty is because manual labeling in online media is an expensive process or is not available in some cases, especially when trying to identify information about a new product [17]. This work's novelty is that the proposed methodology identifies product-feature-related information in online data without utilizing training data.

2.2 Product Feature Extraction

A product feature is defined as an attribute of a product that is of interest to customers. Automated methods of extracting product features from product-related data are an emerging area of interest in the product design community. Dong and Agogino develop an automated methodology to acquire a representation of the product design, based upon the terminological patterns in the design documents [18]. Wassenaar et al. present a discrete choice demand model to identify customer requirements, based on product design attributes [19]. Yoshimura et al. propose a machine product design optimization method for obtaining product design solutions based on product design optimization problems that then identifies optimum design solutions by selecting essential factors in the product design [20]. Zhao et al. propose a novel approach by generalizing product attributes' syntactic structures with intuitive heuristics and syntactic structure similarity [21]. Wang et al. develop a systematic methodology that elicits product attributes for design selection using web-based user-generated content [22]. Tucker and Kim propose the preference trend mining (PTM) model to guide product architecture by indicating when certain product features should be included or excluded in next generation product designs [23]. Poppa et al. present novel information retrieval approaches to assess the similarity of engineering design data using vector space query matching techniques for conceptual design [24].

Recently, online media data have been substantially used for product feature extraction. Tucker and Kim propose a methodology using publicly available customer review data for extracting product features [1]. Albornoz et al. predict a product's overall rating based on user opinions of different product features evaluated in online customer reviews [2]. Rai proposes a method that identifies key product attributes from online customer reviews and ranks each product's attributes with part-of-speech (POS) tagging [3]. Tuarob and Tucker investigate the potential uses of large-scale social media data in product design, including identifying notable features, predicting product longevity, and forecasting product sales using real-world smartphone data [4]. Chou and Shu identify several characteristics of the corpus from online consumer product reviews and develop methodologies to identify affordance-containing reviews and extract their features [5]. Zhou et al. introduce a sentiment analysis and a case analogical reasoning model for latent customer needs elicitation from online product reviews [6].

Table 3 shows a summary of existing studies and our proposed research on product feature extraction in online media. Existing studies in classifying online media data for extracting product features that consider both type I and type II errors need labeled training data [1-2, 6]. Some other existing studies in product feature extraction do not use labeled training data, but the objectives are not classifying online media data as product-feature-related or not (e.g., clustering online customer reviews [3, 5], ranking product-feature-related terms [4].) This research's originality is derived from how online media data can or cannot be classified as product-feature-related, which minimizes type I and type II identification errors through the proposed Bayesian sampling method without labeled training data.

Table 3 Summary of existing studies and our study on product feature extraction in online media

Ref	Method	Objective	Data	Type I & II errors
[1-2, 6]	supervised machine learning	classification	labeled training data	considered both
[3, 5]	unsupervised machine learning	clustering	unlabeled data	both not considered
[4]	LDA	term rankings	unlabeled data	both not considered
Ours	Bayesian sampling method	classification	unlabeled data	considered both

2.3 Sample Size Determination and Non-i.i.d. Approaches

For most research, the individuals in a population cannot be studied due to time, financial, and other resource constraints. In such situations, only a sample would be used, with the results generalized to cover the whole population. Different sample sizes from the same population would give different results [25]. Therefore, most research places importance on identifying the optimal sample size.

Sample size determination has been widely studied in various areas. Müller et al. consider the choice of an optimal sample size for multiple-comparison problems, which is the choice of the number of microarray experiments to be carried out when learning about differential gene expression [26]. Fritz and MacKinnon present the necessary sample sizes for the most common and recommended mediation tests for various combinations of parameters [27]. Byrd et al. present a methodology that uses varying sample sizes in batch-type optimization methods for large-scale machine learning problems [28].

The aforementioned literature proposes methodologies and applications for sample size determination based on an assumption that all samples are independent and identically distributed. However, it cannot be assumed that terms and documents in online media data are independent of each other. For example, if the term “camry” is contained in some document, the probability that the term “toyota” is also contained would be higher than the probability that the term “samsung” is contained in that instance. Table 4 shows an example of how the identical term “apple” can be used differently by adjacent terms in each document. The term “apple” means the *iPhone* manufacturer in the first *tweet*, but the term “apple” means a fruit in the second *tweet*.

In addition, it cannot be guaranteed that terms in online media data are identically distributed, since some terms can be used more than once in one instance. Table 5 shows an example of an *iPhone 6* user review [29]. It shows how the term “android” can be used multiple (three) times in one document.

Table 4 An example of non-independent terms

tweets
*(ID: bo****)* Brand new iPhone 4 for free, **Apple** iPhone is pretty awesome sometimes.
*(ID: pe******)* I've been trying to open this **apple** juice for 12 minutes, I'm on the verge of tears. Weight training definitely failed me & my arm strength.

Table 5 An example of non-identically distributed terms

User Review
*(ID: hu*****)* Hardware wise, I do believe it's overpriced compared to **Android** flagships but then again, leaving iOS and venturing into the **Android** world rampant with malware (not that there isn't iOS specific malware but it's a tiny pimple compared to **Android**).

Several non-i.i.d. methodologies for classification exist. Liu and Singh demonstrate how classical i.i.d. bootstrap on data that is independent but not identically distributed is frequently appropriate in the case of the sample mean [30]. Zhou et al. develop non-i.i.d. multi-instance learning methodologies with a set of labeled bags, each containing many instances [31]. Ganiz et al. propose a supervised machine-learning method for text classification in consideration of latent relations between words [32]. Görnitz et al. propose a support-vector-machine-based methodology, that captures the hidden state of label noise in the presence of systematic and non-i.i.d. label noise [33].

While several existing classification methodologies can be applied to non-i.i.d. data, the options for selecting the optimal number of search keywords in order to minimize type I and type II identification errors for non-i.i.d. data are limited. The novelty of this work is that, in contrast to traditional sample size determination problems and non-i.i.d. classification approaches, the proposed methodology can also determine the optimal number of search keywords for non-i.i.d. data.

3 METHODOLOGY

3.1 Overview and Definition

Figure 1 outlines the methodology. First, potential terms in both primary and secondary data sources are identified. Then, optimal search keywords and the optimal number of keywords are determined among potential terms. Finally, in online media networks, product-feature-related data are collected with determined search keywords and the number of search keywords.

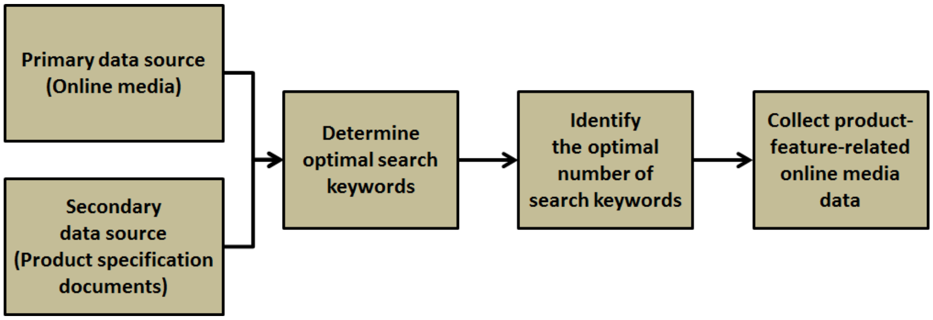

Primary data source (Online media)
Secondary data source (Product specification documents)
Determine optimal search keywords
Identify the optimal number of search keywords
Collect product-feature-related online media data

Fig. 1 Overview of the proposed methodology

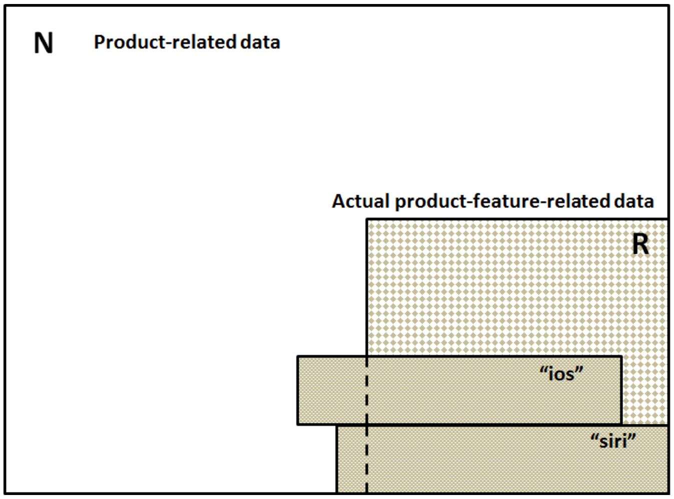


Fig. 2 N, R, and data containing "siri" or "ios"

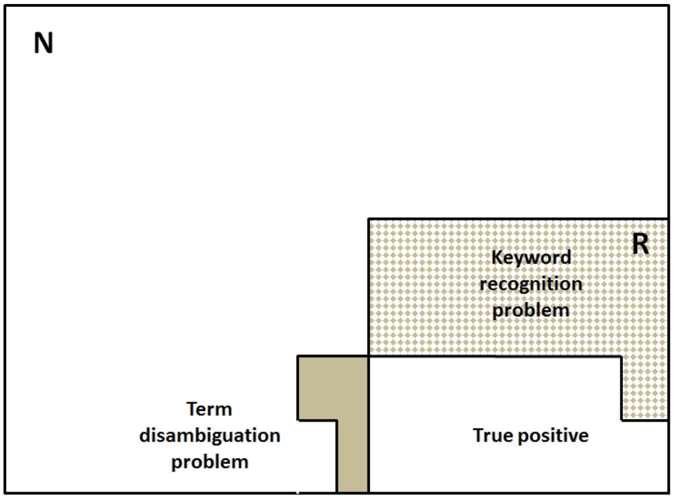


Fig. 3 Term disambiguation problem and keyword recognition problem

Figure 2 indicates the entire set of product-related online media data (*N*), the set of actual product-feature-related data (*R*), and the set of data containing the keywords “siri” or “ios.” The aim of the methodology is to identify actual product-feature-related data (*R*) using product-feature-related keywords.

Discovered product-feature-related data using both the terms “siri” and “ios” are larger than discovered product-feature-related data only using the term “siri.” In Fig. 3, the dotted gray area shows a keyword recognition problem (undiscovered product-feature-related data), based on using the keywords “siri” and “ios” to search for product-feature-related data.

However, when the terms “siri” and “ios” are both used, data containing (misidentified) non-product-feature-related data also increase relative to only using the term “siri.” In Fig. 3, a dark gray area shows a term disambiguation problem (misidentified non-product-feature-related data) by containing the keywords “siri” and “ios.” Therefore, in considering both a keyword recognition problem and a term disambiguation problem, identifying the optimal product-feature-related keywords and the number of search keywords is strongly related the accuracy of identifying product-feature-related information.

3.2 A Proposed Bayesian Sampling Algorithm for Identifying Optimal Product-feature-related Keyword Combinations

The proposed methodology aims to identify optimal search keyword combinations without labeled training data in order to reduce misidentification caused by keyword selection. The proposed methodology can be applied to identify the optimal number of search keywords in non-i.i.d. data, since the methodology does not use traditional statistics based on the i.i.d. assumption and exploits the information about relationships between terms (keywords) to determine the number of search keywords.

Suppose a term t_k is the *kth* product-feature-related term selected as a keyword among all terms in the data sources, given that $t_1, \cdots, t_{k-1}$ are already selected through the proposed methodology. Let a set $S(t_k)$ be a set of all instances (documents) that contains the term t_k. $P(S(t_k))$, calculated using Eq. (1), is defined as the probability that a randomly selected instance in the set N contains a term t_k. Since manual labeling is unnecessary for just counting the number of instances in N and the number of instances in $S(t_k)$, $P(S(t_k))$ can be calculated without labeled training data.

$$P(S(t_k)) = \frac{\#\ of\ instances\ in\ S(t_k)}{\#\ of\ instances\ in\ N} \tag{1}$$

Let $P(R)$ be the unknown probability that a randomly selected instance in the set *N* is a product-feature-related instance and let $P(R')$ be $1 - P(R)$. Suppose that $P(S(t_k)|R)$ is the conditional probability of observing a term t_k in a product-feature-related instance. The conditional probability $P(S(t_k)|R)$ is approximated in a method similar to the conditional probability used in a relevance model, which is proposed by Lavrenko and Croft as Eq. (2) [34].

$$P(S(t_k)|R) \approx P(S(t_k)|S(t_1 \cdots t_{k-1})) \tag{2}$$

Without labeled training data, $P(R)$ is unknown and the only information needed is the vector of identified terms $t_1, \cdots, t_{k-1}$, determined through the proposed methodology's previous iterations. In this condition, the possible approximation of the probability of observing a term t_k in a product-feature-related instance ($P(S(t_k)|R)$) is the probability of a co-occurrence between t_k and a set of instances containing the identified terms $t_1, \cdots, t_{k-1}$ in the previous iterations. To apply the approximation, the first term t_1 should be preassigned as the product's name (e.g., "iphone") or special feature (e.g., "siri," "ios," "facetime") that designers are interested in, because the approximation of $P(S(t_k)|R)$ is based on at least one predetermined term. In a method similar to Eq. (2), the possible estimation of the probability of observing a term t_k in a non-product-feature-related instance $P(S(t_k)|R')$ is the probability of co-occurrence between t_k and the entire set of product-related instances (*N*), except a set of instances containing the identified terms $t_1, \cdots, t_{k-1}$ in the previous iterations (Eq. (3)).

$$P(S(t_k)|R') \approx P(S(t_k)|N - S(t_1 \cdots t_{k-1})) \tag{3}$$

$P(S(t_k))$ can be expressed by Eqs. (2) and (3) and the law of total probability (Eq. (4)).

$$P(S(t_k)) = P(S(t_k)|R) \cdot P(R) + P(S(t_k)|R') \cdot P(R')$$

$$= P(S(t_k)|R) \cdot P(R) + P(S(t_k)|R') \cdot \{1 - P(R)\}$$

$$= P(S(t_k)|R') + \{P(S(t_k)|R) - P(S(t_k)|R')\} \cdot P(R) \tag{4}$$

$P(R)$ can be computed using Eq. (4). $P(R_k)$ is used as the estimation of $P(R)$ given that $t_1, \cdots, t_{k-1}$ were identified, since $P(R)$ is the unknown probability. $P(R_k)$ can be estimated by Eqs. (2) and (3) without labeled training data, since labeled training data are not necessary for calculating $P(S(t_k)), P(S(t_k)|S(t_1 \cdots t_{k-1}))$, and $P(S(t_k)|N - S(t_1 \cdots t_{k-1}))$ (Eq. (5)).

$$P(R) = \frac{P(S(t_k)) - P(S(t_k)|R')}{P(S(t_k)|R) - P(S(t_k)|R')}$$

$$\approx \frac{P\big(S(t_k)\big) - P\big(S(t_k)\big|N - S(t_1 \cdots t_{k-1})\big)}{P\big(S(t_k)\big|S(t_1 \cdots t_{k-1})\big) - P\big(S(t_k)\big|N - S(t_1 \cdots t_{k-1})\big)} = P(R_k) \tag{5}$$

Let $P(R|S(t_k))$ be the conditional probability that a randomly selected instance containing the term t_k is a product-feature-related instance. Eqs. (2) and (5) and Bayes' theorem can be applied to estimate $P(R|S(t_k))$ without labeled training data, because labeled training data are not required for calculating $P(S(t_k)), P(S(t_k)|S(t_1 \cdots t_{k-1}))$, and $P(R_k)$ (Eq. (6)).

$$P(R|S(t_k)) = \frac{P(S(t_k)|R)}{P(S(t_k))} \cdot P(R) \approx \frac{P(S(t_k)|S(t_1 \cdots t_{k-1}))}{P(S(t_k))} \cdot P(R_k) \tag{6}$$

$P\big(S(t_k)\big|S(t_1 \cdots t_{k-1})\big)$ is computed by Kolmogorov's probability theory (Eq. (7)).

$$P\big(S(t_k)\big|S(t_1 \cdots t_{k-1})\big) = \frac{P(S(t_k) \cap S(t_1 \cdots t_{k-1}))}{P\big(S(t_1 \cdots t_{k-1})\big)} = \frac{P(S(t_1 \cdots t_k))}{P(S(t_1 \cdots t_{k-1}))} \tag{7}$$

Then, $P(R|S(t_k))$ can be estimated by Eqs. (6) and (7) without labeled training data, because labeled training data are unnecessary for calculating $P(S(t_k)), P(S(t_1 \cdots t_k)), P(S(t_1 \cdots t_{k-1}))$, and $P(R_k)$ (Eq. (8)).

$$P(R|S(t_k)) \approx \frac{P(S(t_1 \cdots t_k))}{P(S(t_1 \cdots t_{k-1})) \cdot P(S(t_k))} \cdot P(R_k) \tag{8}$$

The probability of a true positive caused by containing the terms $t_1, \cdots, t_k$ $\big(P(true, k)\big)$ can be estimated as Eq. (9) without labeled training data.

$$
\begin{aligned}
P(true, k) &\approx \sum_{i=1}^{k} P(S(t_i) \cap R) = \sum_{i=1}^{k} \big\{P(R|S(t_i)) \cdot P\big(S(t_i)\big)\big\} \\
&= P(R|S(t_1)) \cdot P\big(S(t_1)\big) + \sum_{i=2}^{k} \{P(R|S(t_i)) \cdot P(S(t_i))\} \\
&\approx P(R|S(t_1)) \cdot P\big(S(t_1)\big) + \sum_{i=2}^{k} \left\{ \frac{P(S(t_1 \cdots t_i)) \cdot P(S(t_i))}{P\big(S(t_1 \cdots t_{i-1})\big) \cdot P(S(t_i))} \cdot P(R_i) \right\} \\
&= P(R|S(t_1)) \cdot P\big(S(t_1)\big) + \sum_{i=2}^{k} \left\{ \frac{P(S(t_1 \cdots t_i))}{P\big(S(t_1 \cdots t_{i-1})\big)} \cdot P(R_i) \right\}
\end{aligned} \tag{9}
$$

The probability of a type I error (a term disambiguation problem) caused by containing the terms $t_1, \cdots, t_k$ $\big(P(type\ I, k)\big)$ can also be estimated as Eq. (10) without labeled training data.

$$
\begin{aligned}
P(type\ I, k) &\approx \sum_{i=1}^{k} \{P(S(t_i)) - P(S(t_i) \cap R)\} = \sum_{i=1}^{k} \{P(S(t_i)) - P(R|S(t_i)) \cdot P(S(t_i))\} \\
&= P\big(S(t_1)\big) \cdot \{1 - P(R|S(t_1))\} + \sum_{i=2}^{k} \{P(S(t_i)) - P(R|S(t_i)) \cdot P(S(t_i))\} \\
&\approx P\big(S(t_1)\big) \cdot \{1 - P(R|S(t_1))\} + \sum_{i=2}^{k} \left\{ P(S(t_i)) - \frac{P(S(t_1 \cdots t_i)) \cdot P(S(t_i))}{P\big(S(t_1 \cdots t_{i-1})\big) \cdot P(S(t_i))} \cdot P(R_i) \right\} \\
&= P\big(S(t_1)\big) \cdot \{1 - P(R|S(t_1))\} + \sum_{i=2}^{k} \left\{ P(S(t_i)) - \frac{P(S(t_1 \cdots t_i))}{P\big(S(t_1 \cdots t_{i-1})\big)} \cdot P(R_i) \right\}
\end{aligned} \tag{10}
$$

The probability of a type II error (a keyword recognition problem) caused by the terms $t_1, \cdots, t_k$ $\big(P(type\ II, k)\big)$ can be estimated as Eq. (11), because the summation of $P(true, k)$ and $P(type\ II, k)$ is $P(R)$ as shown in Fig. 3.

$$P(type\ II, k) = P(R) - P(true, k) \tag{11}$$

An estimated F-measure ($F_{\beta}(k)$) is used as a performance measure (Eq. (12)).

$$F_{\beta}(k) = \frac{(1+\beta^2) \cdot P(true, k)}{(1+\beta^2) \cdot P(true, k) + P(type\ I, k) + \beta^2 \cdot P(type\ II, k)}$$

$$= \frac{(1+\beta^2) \cdot P(true, k)}{P(true, k) + P(type\ I, k) + \beta^2 \cdot P(R)} \tag{12}$$

The proposed methodology aims to identify a combination of the terms $t_1, \cdots, t_k$ that maximizes an estimated F-measure ($F_{\beta}(k)$). Regardless of different term combinations, the unknown probability $P(R)$ remains the same and $P(R)$ is considered constant. As a result, labeled training data for calculating $P(R)$ are unnecessary. In the same manner, $P(R|S(t_1))$ is considered constant. Labeled training data for calculating $P(R|S(t_1))$ are then not necessary, because at least one keyword is required to search for necessary information in online media data, and a term t_1 is preassigned before applying the proposed algorithm. Since the proposed methodology compares the values of $F_{\beta}(k)$, which are obtained through different term combinations in order to find the optimal term combinations, instead of finding the exact estimation of F-measure, researchers can arbitrarily set constants $P(R)$ and $P(R|S(t_1))$.

The weight of a type II error (β) can be determined differently by which type of error (a type I error or a type II error) is more significant for each application. If the significances of a type I error and a type II error are not regarded as different, β is set to 1. Primary and secondary data sources are used to provide potential search keyword candidates for the proposed methodology. Data preprocessing such as stemming, removing hyperlinks, and correcting misspellings is required to improve the accuracy of product-related information identification [35].

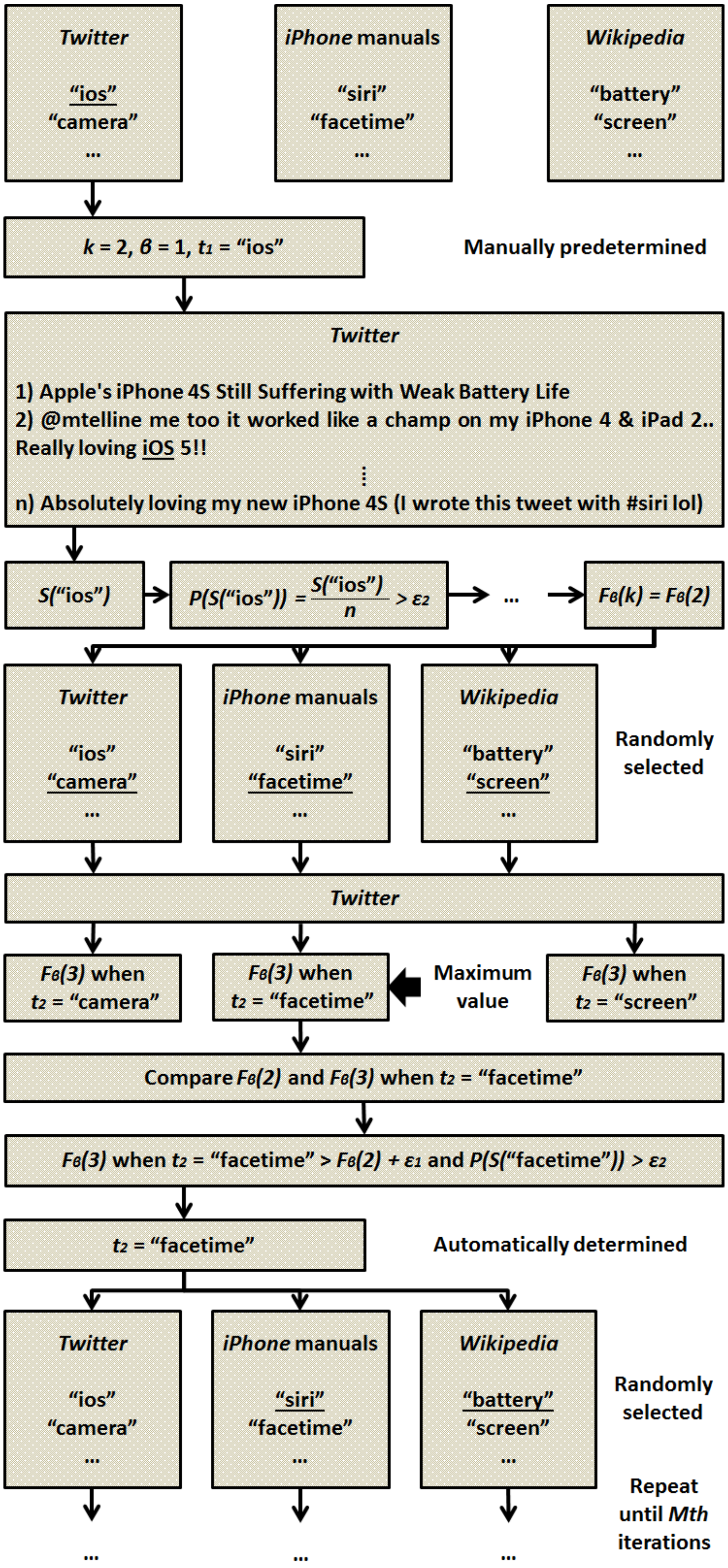
Primary Data Source Both the entire set of product-related data (*N*) and the set of instances containing selected keywords $\left(S(t_1 \cdots t_{k-1})\right)$ through the proposed algorithm's previous iterations are used as primary data sources for sampling potential terms in order to select t_k $(k \geq 2)$. *N* is not used as the sole primary data source, because the terms in online media data are not independent (Sec. 2.3). Therefore, it can be assumed that the probability that the terms from $S(t_1 \cdots t_{k-1})$ are product-feature-related is higher than the probability that the terms from *N* are product-feature-related.

Secondary Data Source Product specification documents provide existing products' actual non-biased features as a secondary data source. A secondary data source is classified between two separate branches: a non-technical data source, that has general information about products (e.g., newspaper articles, *Wikipedia* articles), and a technical data source, that has technical specifications and the ground truth features of products (e.g., the manufacturers' product technical specification manuals). It can be assumed that both a technical data source and a non-technical source have potential keyword candidates (product-feature-related terms), because product features are important factors for writing both data sources. A technical data source and a non-technical data source are used to sample potential terms for optimal search keywords.

Algorithm 1 summarizes the steps of the proposed Bayesian sampling method and Figure 4 illustrates the process example of the proposed methodology.

Algorithm 1: The Bayesian Sampling Algorithm for Identifying Optimal Keyword Combinations

```
STEP 1 Assign a predetermined product-feature-related term (e.g., the product's name or major feature) to the first term $t_1$ and set $k=2$ and $m=0$
STEP 2 If $m=M$, set $k=k$-1and go to STEP 6
Otherwise, randomly sample four terms as $\mathbf{W}= \{w_N, w_{S(t_1 \cdots t_{k-1})}, w_{tech}, w_{nontech}\}$
STEP 3 Select the term $t_k$ that has the maximum estimated F-measure $(F_\beta(k))$ among W
STEP 4 Compare $F_\beta(k)$ and $F_\beta(k-1)$
STEP 5 If $F_\beta(k) > F_\beta(k-1) + \varepsilon_1$ and $P(S(t_k)) > \varepsilon_2$, contain $t_k$ as a product-feature-related keyword, set $k=k+1$ and $m=m+1$, and go to STEP 2
Otherwise, set $m=m+1$ and go to STEP 2
STEP 6 Stop and return search keywords $t_1, \cdots, t_k$ and sets $S(t_1) \cup \cdots \cup S(t_k)$
```


Twitter
"ios"
"camera"
...
iPhone manuals
"siri"
"facetime"
...
Wikipedia
"battery"
"screen"
...
k = 2, β = 1, t_1 = "ios"
Manually predetermined
Twitter

1) Apple's iPhone 4S Still Suffering with Weak Battery Life

2) @mtelline me too it worked like a champ on my iPhone 4 & iPad 2.. Really loving iOS 5!!

⋮

n) Absolutely loving my new iPhone 4S (I wrote this tweet with #siri lol)

S("ios")
$P(S(\text{"ios"})) = \frac{S(\text{"ios"})}{n} > \varepsilon_2$
...
$F_\beta(k) = F_\beta(2)$
Twitter
"ios"
"camera"
...
iPhone manuals
"siri"
"facetime"
...
Wikipedia
"battery"
"screen"
...
Randomly selected
Twitter
$F_\beta(3)$ when t_2 = "camera"
$F_\beta(3)$ when t_2 = "facetime"
Maximum value
$F_\beta(3)$ when t_2 = "screen"
Compare $F_\beta(2)$ and $F_\beta(3)$ when t_2 = "facetime"
$F_\beta(3)$ when t_2 = "facetime" > $F_\beta(2) + \varepsilon_1$ and $P(S(\text{"facetime"})) > \varepsilon_2$
t_2 = "facetime"
Automatically determined
Twitter
"ios"
"camera"
...
iPhone manuals
"siri"
"facetime"
...
Wikipedia
"battery"
"screen"
...
Randomly selected
Repeat until *Mth* iterations
...
...
...

Fig. 4 The process of the proposed methodology

Identifying optimal product-feature-related search keyword combinations takes the following steps in Algorithm 1. Variables k and m are used for tracking the number of identified search keywords and the number of iterations, respectively. w_N, $w_{S(t_1 \cdots t_{k-1})}$, w_{tech} and $w_{nontech}$ are defined as unselected sampled terms in the previous iterations from a primary data source N, a primary data source $S(t_1 \cdots t_{k-1})$, a technical data source, and a non-technical data source, respectively. If $F_\beta(k)$ is greater than $F_\beta(k-1) + \varepsilon_1$ and $P(S(t_k))$ is greater than ε_2, then t_k is contained in the search keyword list. ε_1 is used to avoid search keywords that are selected more than necessary. For example, if an analysist thinks that a search keyword that increases an estimated F-measure no more than 1% can be considered superfluous, ε_1 is set to 0.01. ε_2 is used to avoid superfluous terms that are only contained in just a few instances and are selected as search keywords. For instance, if an analysist thinks that the selected search keyword should return at least 10% of all instances, ε_2 is set to 0.1. Stop words (e.g., "a," "the," "be") are used to avoid non-product-feature-related language-specific functional words that are selected as search keywords [36]. The algorithm terminates after M iterations. The stopping criteria (M) can be determined differently by different applications (e.g., iterate until a given F-measure is attained). Four new sampled terms **W** from each data source are required for each iteration (STEP 2) to search potential keyword candidates. A simple random sample without replacement is used for sampling four new terms for each iteration. Each term has the same probability of being selected at each iteration during the sampling process. In addition, selected terms in the previous iterations are not sampled again, since it is not necessary for the proposed algorithm to check an identical term multiple times.

Through the proposed Bayesian-based method, product-feature-related terms $t_1, \cdots, t_k$ can be selected as search keywords for the identification of product-feature-related information in online media without the need for manual labeling of training data. The optimal number of search keywords is k. In addition, a new instance can be estimated whether or not it has product-feature-related information by identified search keywords. If a new instance contains at least one keyword, it can be considered as a product-feature-related instance. Otherwise, it is considered as a non-product-feature-related instance.

3.3 The Complexity of the Proposed Bayesian Sampling Method

This section presents the algorithmic complexity of the proposed Bayesian sampling method. Big-O (O) notation, which gives the asymptotic upper bound on execution time but is not necessarily related to running time for every input combination, is used for the complexity [37].

Let P be the maximum number of terms in one document and Q be the total number of documents in N. Sampling four terms involves O(1), regardless of the size of both primary and secondary data sources, because it is not required to check every term in N for sampling. The calculation of estimated F-measures $(F_\beta(k))$ and related probabilities (Eq. (1) - Eq. (10)) requires O(PQ), because checking every terms in *N* whether each document has related terms (e.g., search keyword candidates) or not is required to calculate estimated F-measures and related probabilities. Therefore, the execution time for each iteration is directly proportional to *PQ*.

Therefore, the execution time for the proposed algorithm involves O(MPQ), since the number of iterations is *M* and *M* and *Q* are independent. The proposed algorithm is a polynomial time algorithm, and the execution time is directly proportional to *PQ*, since users can set *M* before running the algorithm. The proposed methodology has less complexity (O(PQ)) than existing machine learning techniques for one iteration (e.g., logistic regression's time complexity is $O(P^2Q+P^3)$ and support vector machines' time complexity is $O(PQ^2)$ for one iteration [38]).

4 APPLICATION

This section introduces a case study involving real-world smartphones (*iPhone 4* and *iPhone 4S*), which is used to verify the proposed methodology. The case study identifies optimal *iPhone*-feature-related keyword combinations. Experiments are conducted with Java 1.7.0. Fox presents a stop list used in automatic indexing to filter out words that would make poor index terms, and the top 278 most frequently occurring words in his list are used as stop words for the experiments [36]. The primary and secondary data sources below are used for the case study.

Primary Data Source The *Twitter* dataset, which was collected randomly using the provided *Twitter* API and is comprised of 800 million *tweets* in the United States during a period from March 2011 to September 2012 is used [4]. Among the whole dataset, 95,033 *tweets* related to the *iPhone 4* or the *iPhone 4S* are used in this *iPhone* case study. Among 95,033 *tweets (N),* 9,403 *tweets (R)* were manually labeled by 10 undergraduate students at Penn State for two weeks as *iPhone*-feature-related *tweets.* The manual labeling of the *Twitter* data is done in this case study to serve as ground truth validation of the proposed methodology. With ground truth data, the performance of the proposed methodology can be compared to existing methods for accuracy to achieve a solution. Once the performance measures for the algorithm have been established, actual implementation of the algorithm would not require manual labeling of the entire online data stream.

Secondary Data Source The *iPhone 4* and *iPhone 4S* technical specification manuals (technical data sources) from the manufacturer (*Apple*) and *Wikipedia* articles for the *iPhone 4* and the *iPhone 4S* (non-technical data sources) are used as secondary data sources. The reason why *Wikipedia* is selected as a non-technical data source is that an article on *Wikipedia* can be written and edited by more than one person, because anyone who can access *Wikipedia* can edit the article. Therefore, it can be assumed that the probability that a single article from *Wikipedia* has misinformation is lower than a single instance from other online media (e.g., *Facebook*, *Twitter*), since other people can correct an article from *Wikipedia* even if the first author wrote misinformation [39]. Only textual information is extracted from the *iPhone 4* and *iPhone 4S* manuals and the *Wikipedia* articles.

The results of the proposed methodology 1) with only primary data sources (P), 2) with only secondary data sources (S), and 3) with both primary and secondary data sources (PS) are compared with a baseline method and an expert-keyword-selection method. A random-keyword-selection method, which just selects keywords randomly without any prior information, is used as the baseline method. Let Baseline(*j*) be the baseline method with *j* keyword(s). An expert-keyword-selection method selects keywords from product-feature-related terms identified by existing research (i.e., A top 10 *iPhone* feature list presented by Tuarob and Tucker [4] in this case study) using domain experts (i.e., experts in product feature extraction at Penn State in this case study). Let Expert(*j*) be the expert-keyword-selection method with *j* keyword(s). Existing machine learning algorithms are not used in this case study, because 1) the proposed Bayesian sampling method identifies optimal keyword combinations without labeled training data and 2) the proposed methodology has less time complexity than existing machine learning algorithms (Sec. 3.3). Eleven cases (Baseline(1), Baseline(2), Baseline(3), Baseline(4), Expert(1), Expert(2), Expert(3), Expert(4), P, S, and PS) are run 30 times respectively, and manually labeled data are used as ground truth data. An F_1 score is used for verifying the proposed methodology.

To maintain consistency, the first term t_1 is identically assigned to all eleven cases above. Domain experts (i.e., experts in product feature extraction at Penn State in this case study) select the terms "siri," "ios," and "facetime," which represent the *iPhone*'s special features that designers are interested in, as the first terms used (t_1) to compare the proposed methodology's and other methods' (the baseline method and the expert-keyword-selection method) results. The term "iphone" is not used as the first keyword, since every *tweet* has the term "iphone" in this case study and the objective of the proposed method is identifying *iPhone*-feature-related *tweets* among all *iPhone*-related *tweets.* In this case study, identification errors are based on the number of misidentified *tweets* regardless of the types of error (a type I error or a type II error). The significance of a type I error and a type II error is regarded as the same in this study, hereby setting β as 1. Domain experts set ε_1, ε_2, and M to 0.01, 0.001, and 1,000, respectively.

5 RESULTS AND DISCUSSION

Table 6 and Figure 5 show the results of the number of search keywords returned by selected keyword combinations (average/mode) and the average values of the F_1 scores for 30 runs, respectively. According to Table 6 and Figure 5, the proposed method (P, S, and PS) provides better F_1 scores than the baseline method and the expert-keyword-selection method, regardless of the first keywords. It is concluded that the proposed Bayesian sampling method outperforms the baseline method and the expert-keyword-selection method on the condition of no prior information such as labeled training data. In addition, the proposed method, with both primary and secondary data sources (PS), presents a better F_1 score than the proposed method with only primary data sources (P) and secondary data sources (S) (7% and 6% higher on average, respectively.) Therefore, it can also be concluded that the effects of secondary data sources on sample keyword candidates is not negligible. The average number of keywords in Table 6, which means the average number of search keywords identified for each run, also show that the proposed Bayesian sampling method (P, S, and PS) identifies keyword combinations mostly containing 2 or 3 search keywords.

Table 6 Comparison between Baseline(*j*), Expert(*j*), P, S, and PS

1st term	Method	# of keywords		Average F_1 (%)
		Mean	Mode	
”siri”	Baseline(1)	1 (“siri”)		39
	Baseline(2)	2		36
	Baseline(3)	3		34
	Baseline(4)	4		35
	Expert(1)	1 (“siri”)		39
	Expert(2)	2		51
	Expert(3)	3		53
	Expert(4)	4		50
	P	2.30	2	57
	S	2.41	2	56
	PS	2.37	2	63
“ios”	Baseline(1)	1 (“ios”)		39
	Baseline(2)	2		36
	Baseline(3)	3		33
	Baseline(4)	4		34
	Expert(1)	1 (“ios”)		39
	Expert(2)	2		52
	Expert(3)	3		49
	Expert(4)	4		48
	P	2.43	2	54
	S	2.55	3	57
	PS	2.50	3	63
“facetime”	Baseline(1)	1 (“facetime”)		13
	Baseline(2)	2		15
	Baseline(3)	3		15
	Baseline(4)	4		12
	Expert(1)	1 (“facetime”)		13
	Expert(2)	2		38
	Expert(3)	3		40
	Expert(4)	4		37
	P	2.78	3	52
	S	2.70	3	51

	PS	2.62	3	58
Average	**Baseline(1)**	1		30
	Baseline(2)	2		29
	Baseline(3)	3		27
	Baseline(4)	4		27
	Expert(1)	1		30
	Expert(2)	2		47
	Expert(3)	3		47
	Expert(4)	4		45
	P	2.50	2	54
	S	2.55	3	55
	PS	2.50	3	61

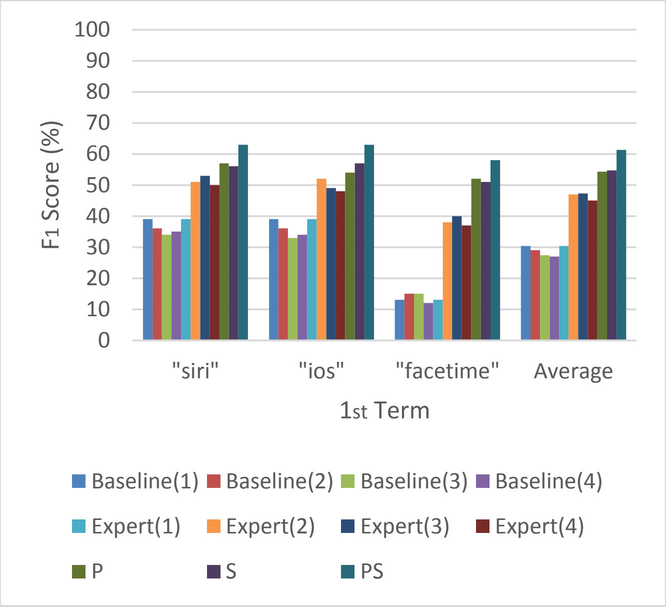


Fig. 5 Average values of the F_1 scores

Table 7 lists the top five search keyword combinations among all keyword combinations identified through this case study (P, S, and PS), along with the F_1 scores. Table 7 indicates that keyword combinations containing “siri,” “ios,” “battery,” “camera,” or “facetime” present higher F_1 scores, regardless of what the first keyword is. If other product-feature-related terms (e.g., photo, screen, case) are added to these keyword combinations, the F_1 scores decrease because of a relatively large increase in type I errors rather than a decrease in type II errors in this case study.

Table 7 Top five keyword combinations

Keyword combinations	# of keywords	F_1 (%)
“siri” + “ios” + “facetime” + “battery”	4	82
“siri” + “ios” + “facetime” + “camera”	4	81
“siri” + “ios” + “battery”	3	73
“siri” + “ios” + “camera”	3	72
“siri” + “ios” + “facetime”	3	68

Table 8 compares the results by the best keyword combinations (“siri,” “ios,” “facetime,” and “battery”) in Table 7 and the manually labeled results (ground truth) with five sampled *tweets*. Table 8 shows that the proposed method identifies *iPhone*-feature-related *tweets* correctly with the exception of the third *tweet*. The third *tweet* in Table 8 can be manually labeled as an *iPhone*-feature-related *tweet* because it contains an *iPhone* feature (“photo”), even though it does not contain keywords that the proposed method identified.

Table 8 Comparison between manually labeled results and the proposed method’s results

tweets	Manually labeled	Proposed method
Just tried Apple **facetime** for the first time on my iPhone 4 with @OhHeyForrest. It seems pretty cool! Worked great.	*iPhone*-feature-related	*iPhone*-feature-related
The new **ios** 5.1 for #iphone 4S is pretty Awesome! It's got 4G, **camera button** on lock **screen**, And **siri** in Japanese.	*iPhone*-feature-related	*iPhone*-feature-related
I am incredibly proud of my first iPhone 4 **photo**. Ron would be too. .	*iPhone*-feature-related	not related
@maliababyyy yes I have a iPhone 4 too it's really Awesome!!	not related	not related
RIP Steve Jobs.... I hated the iPhone 4s too!	not related	not related

6 CONCLUSION

The objective of the proposed work is to present a methodology to reduce product-feature-related information identification errors in online media networks and increase significant product-feature-related knowledge. The proposed Bayesian-based methodology identifies optimal search keyword combinations without labeled training data from non-i.i.d. online data. In addition, a new instance (e.g., a *tweet*) can be estimated whether or not it has product-feature-related information by selected search keywords. It is proven that the proposed algorithm is a polynomial time algorithm and has less time complexity than existing machine learning algorithms.

The methodology comprises of four main steps. First, potential search keyword candidates in both primary and secondary data sources are identified. Second, optimal search keywords are determined among potential terms using the proposed Bayesian sampling method. Third, the number of search keywords is identified with selected search keyword combinations. Finally, in online media networks, product-feature-related data are collected with determined search keyword combinations.

A case study involving real-world smartphones (*iPhone 4* and *iPhone 4S*) used to verify the proposed methodology is presented. The results show that the proposed method, with both primary and secondary data sources (PS), provides better F_1 scores than other methods. It is concluded that the proposed methodology can be useful to reduce identification errors without prior information such as labeled training data. The proposed method's top five search keyword combinations are also presented for this case study.

The authors of this work will propose a theoretical approach of how to select the first search keyword (t_1) (e.g., applying the cold start problem), because the first keyword strongly affects identifying other keywords in the proposed methodology. Future work will also include identifying what secondary data source provides better potential keywords and the most consistent results. A methodology used to identify optimal search keyword combinations in consideration of each media users' expertise (e.g., lead users [40], ordinary users) will also be proposed in future research.

ACKNOWLEDGMENT

The authors would like to acknowledge Penn State's Global Engagement Network (GEN) grant for funding this work. The authors acknowledge Zhen Sun for the programming aspects of this research. The authors thank Dr. Suppawong Tuarob for his valuable comments on this paper.

REFERENCES

[1] C. S. Tucker and H. M. Kim, "Data-Driven Decision Tree Classification for Product Portfolio Design Optimization," *Journal of Computing and Information Science in Engineering*, vol. 9, no. 4, p. 041004, 2009.

[2] J. C. de Albornoz, L. Plaza, P. Gervás, and A. Díaz, "A joint model of feature mining and sentiment analysis for product review rating," in *Advances in information retrieval*, Springer, 2011, pp. 55–66.

[3] R. Rai, "Identifying key product attributes and their importance levels from online customer reviews," in *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2012, pp. 533–540.

[4] S. Tuarob and C. Tucker, "Quantifying Product Favorability and Extracting Notable Product Features using Large Scale Social Media Data," *Journal of Computing and Information Science in Engineering*, vol. 15, no. 3, 2015.

[5] A. Chou and L. H. Shu, “Towards Extracting Affordances From Online Consumer Product Reviews,” in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2014, pp. V007T07A030–V007T07A030.

[6] F. Zhou, J. (Roger) Jiao, and J. Linsey, “Latent Customer Needs Elicitation by Use Case Analogical Reasoning from Sentiment Analysis of Online Product Reviews,” *Journal of Mechanical Design*, vol. 137, no. 7, p. 071401, 2015.

[7] S. Tuarob, C. S. Tucker, M. Salathe, and N. Ram, “An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages,” *Journal of Biomedical Informatics*, vol. 49, pp. 255–268, Jun. 2014.

[8] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, “Learning to classify short and sparse text & web with hidden topics from large-scale data collections,” in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 91–100.

[9] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, “Exploiting internal and external semantics for the clustering of short texts using world knowledge,” in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 919–928.

[10] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, “Detecting influenza epidemics using search engine query data,” *Nature*, vol. 457, no. 7232, pp. 1012–1014, Feb. 2009.

[11] A. Culotta, “Towards Detecting Influenza Epidemics by Analyzing Twitter Messages,” in *Proceedings of the First Workshop on Social Media Analytics*, New York, NY, USA, 2010, pp. 115–122.

[12] M. W. Glier, D. A. McAdams, and J. S. Linsey, “Exploring Automated Text Classification to Improve Keyword Corpus Search Results for Bioinspired Design,” *Journal of Mechanical Design*, vol. 136, no. 11, p. 111103, 2014.

[13] E. Aramaki, S. Maskawa, and M. Morita, “Twitter Catches the Flu: Detecting Influenza Epidemics Using Twitter,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA, 2011, pp. 1568–1576.

[14] M. J. Paul and M. Dredze, “A model for mining public health topics from Twitter,” *Health*, vol. 11, pp. 16–6, 2012.

[15] T. Stone and S.-K. Choi, “Extracting consumer preference from user-generated content sources using classification,” *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V03AT03A031–V03AT03A031, Aug. 2013.

[16] M. Fuge, B. Peters, and A. Agogino, “Machine Learning Algorithms for Recommending Design Methods,” *Journal of Mechanical Design*, vol. 136, no. 10, p. 101103, 2014.

[17] N. Slonim and N. Tishby, “The power of word clusters for text classification,” in *23rd European Colloquium on Information Retrieval Research*, 2001, vol. 1, p. 200.

[18] A. Dong and A. M. Agogino, “Text analysis for constructing design representations,” *Artificial Intelligence in Engineering*, vol. 11, no. 2, pp. 65–75, 1997.

[19] H. J. Wassenaar, W. Chen, J. Cheng, and A. Sudjianto, “Enhancing discrete choice demand modeling for decision-based design,” *Journal of Mechanical Design*, vol. 127, no. 4, pp. 514–523, 2005.

[20] M. Yoshimura, M. Taniguchi, K. Izui, and S. Nishiwaki, “Hierarchical Arrangement of Characteristics in Product Design Optimization,” *Journal of Mechanical Design*, vol. 128, no. 4, pp. 701–709, Jul. 2006.

[21] Y. Zhao, B. Qin, S. Hu, and T. Liu, “Generalizing syntactic structures for product attribute candidate extraction,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 377–380.

[22] L. Wang, B. D. Youn, S. Azarm, and P. K. Kannan, “Customer-driven product design selection using web based user-generated content,” in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2011, pp. 405–419.

[23] C. S. Tucker and H. M. Kim, “Trend Mining for Predictive Product Design,” *Journal of Mechanical Design*, vol. 133, no. 11, p. 111008, Nov. 2011.

[24] K. Poppa, R. Arlitt, and R. Stone, “An Approach to Automated Concept Generation Through Latent Semantic Indexing,” in *IIE Annual Conference. Proceedings*, 2013, p. 151.

[25] S. Lemeshow, D. W. Hosmer, J. Klar, S. K. Lwanga, and W. H. Organization, *Adequacy of sample size in health studies*. 1990.

[26] P. Müller, G. Parmigiani, C. Robert, and J. Rousseau, “Optimal sample size for multiple testing: the case of gene expression microarrays,” *Journal of the American Statistical Association*, vol. 99, no. 468, pp. 990–1001, Dec. 2004.

[27] M. S. Fritz and D. P. MacKinnon, “Required sample size to detect the mediated effect,” *Psychological science*, vol. 18, no. 3, pp. 233–239, Mar. 2007.

[28] Richard H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, “Sample size selection in optimization methods for machine learning,” *Mathematical programming*, vol. 134, no. 1, pp. 127–155, Aug. 2012.

[29] “Customer Review,” *Amazon*. [Online]. Available: http://www.amazon.com/review/R1ZZ4LU5RWTHXZ/ref=cm_cr_dp_cmt#wasThisHelpful. [Accessed: 24-Jan-2016].

[30] R. Y. Liu and K. Singh, “Using iid bootstrap inference for general non-iid models,” *Journal of statistical planning and inference*, vol. 43, no. 1, pp. 67–75, Jan. 1995.

[31] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, “Multi-instance learning by treating instances as non-iid samples,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1249–1256.

[32] M. C. Ganiz, C. George, and W. M. Pottenger, “Higher order Naive Bayes: A novel non-IID approach to text classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1022–1034, Jul. 2011.

[33] N. Görnitz, A. K. Porbadnigk, A. Binder, C. Sannelli, M. Braun, K.-R. Müller, and M. Kloft, “Learning and Evaluation in Presence of Non-iid Label Noise,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014, pp. 293–302.

[34] V. Lavrenko and W. B. Croft, “Relevance based language models,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 120–127.

[35] K. Zhang, Y. Cheng, Y. Xie, D. Honbo, A. Agrawal, D. Palsetia, K. Lee, W. Liao, and A. Choudhary, “SES: Sentiment Elicitation System for Social Media Data,” in *2011 IEEE 11th International Conference on*, 2011, pp. 129–136.

[36] C. Fox, “A stop list for general text,” in *ACM SIGIR Forum*, 1989, vol. 24, pp. 19–21.

[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, vol. 6. Cambridge, Mass.: MIT Press, 2001.

[38] C. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, “Map-reduce for machine learning on multicore,” *Advances in neural information processing systems*, vol. 19, p. 281, Dec. 2007.

[39] “Wikipedia,” *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Wikipedia. [Accessed: 24-Jan-2016].

[40] P. R. Berthon, L. F. Pitt, I. McCarthy, and S. M. Kates, “When customers get clever: Managerial approaches to dealing with creative consumers,” *Business Horizons*, vol. 50, no. 1, pp. 39–47, Feb. 2007.

LIST OF TABLES

LIST OF FIGURES