Automated Discovery of Product Feature Inferences within Large Scale Implicit Social Media Data

Suppawong Tuarob
Faculty of Information and
Communication Technology
Mahidol University, Thailand
Email: suppawong.tua@mahidol.edu

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

Conrad S. Tucker
Engineering Design and Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802
Email: ctucker4@psu.edu

Recently, social media has emerged as an alternative, viable source to extract large-scale, heterogeneous product features in a time and cost efficient manner. One of the challenges of utilizing social media data to inform product design decisions is the existence of implicit data such as sarcasm, which accounts for 22.75% of social media data, and can potentially create bias in the predictive models that learn from such data sources. For example, if a customer says "I just love waiting all day while this song downloads", an automated product feature extraction model may incorrectly associate a positive sentiment of "love" to the cell phone's ability to download. While traditional text mining techniques are designed to handle well-formed text where product features are explicitly inferred from the combination of words, these tools would fail to process these social messages that include implicit product feature information. In this paper, we propose a method that enables designers to utilize implicit social media data by translating each implicit message into its equivalent explicit form, using the word concurrence network. A case study of Twitter messages that discuss smartphone features is used to validate the proposed method. The results from the experiment not only show that the proposed method improves the interpretability of implicit messages, but also sheds light on potential applications in the design domains where this work could be extended.

1 Introduction

The rigorous competition in the market space drives designers to create products that better satisfy the majority of customers in a resource efficient manner. Oftentimes, it is crucial that designers are familiar with target customers' needs and preferences, in order to incorporate preferable features and remove weak elements from a design artifact. Recently, literature has shown that information generated by social media users could prove critical to product designers in learning relevant preferences towards products/product features [1–6].

Technological advancements in digital communication has allowed many social media platforms to emerge as an alternative means for communication and information exchange in a timely and seamless manner. Literature in various fields of study has shown successful applications that rely on information extracted from large scale social media data, such as mining healthcare-related information for disease prediction [7–9], detecting earthquake warnings and emergence needs due to natural disasters [10, 11], predicting financial market movement [12, 13], etc.

In the design informatics domain, despite the traditional methods that extract customers' preferences from online product reviews, recent findings have illustrated that social networks could also serve as a viable source of information for mining customers' opinions towards products/product features, due to its fast publication, wide range of users, accessibility, and heterogeneity of contents that provides an opportunity for customers to express opinions about products outside the review sites [2]. A data driven methodology has been proposed to automatically discover notable product features mentioned in social networks [5]. Later, such notable product feature information is incorporated into a decision support framework that helps designers to develop next-generation products [2]. Furthermore, large scale social media data has been established as a viable platform to automatically discover innovative users in social networks [1, 4]. Such innovative users could prove critical to product design and development as they help designers to discover relevant product feature preferences months or even years before they are desired by general customers.

Implicit speech is a form of language usage in which the actual meaning is intended to be comprehended, but not directly stated. A majority manifestation of implicit speech includes *sarcasm*, which has become not only abundant, but also a norm in social networks. Maynard and Greenwood found that roughly 22.75% of social media data is sarcastic [14]. While it is evident that knowledge extracted from social media data is useful to product designers, the applicability of such data pertains to the portion expressed in *explicit* forms, due to the limitation of the underlying natural language processing algorithms that assume the explicit, well-formed textual input. As a result, implicit information would be either treated as noises or misinterpreted, resulting in inaccurate recommendation of product design decision support systems that process the information from large scale social media data. Hence, the ability to automatically understand and correctly interpret such implicit information in social networks would not only reduce the errors caused by methods that are not specifically designed to handle implicit information, but also allow the methodologies to make use of additional implicit data that would have traditionally been disregarded due to being treated as noise.

Examples of explicit and implicit social media messages are given below:

Explicit "`My old 7 inch` *`Samsung Galaxy Tab`* `is my #1 travel companion -` *`perfect size`* `&` *`functionality`*`.`"

Implicit "`I` *`love`* `when my` *`blackberry bold screen`* `freezes, the iphone 4 is definitely on my list of #13thingsiwant right now`"

The first example is considered explicit because it can be directly inferred from both keywords and the grammatical structure that the user may be satisfied with the *perfect size* and *functionality* of his/her *Samsung Galaxy Tab*. On the contrary, the second example does not give any direct information about the *screen* feature of his/her *Blackberry Bold*, and hence is implicit, though it may be inferred that this particular user may feel dissatisfied with his/her *Blackberry Bold* due to its frozen *screen*. If these implicit social media messages remain untreated, two problems could occur:

1. Many data mining algorithms are extraction-based that would classify a social media message whether it is useful or not. Such methods would disregard such implicit data where explicit knowledge could not be extracted, resulting in low utilization of useful data.
2. Sarcastic social media messages may either *exaggerate* (i.e. "`Apparently the new iphone 5 helps you lose weight, you buy it and you can't afford food for a month.`") or *oppose* (i.e. "`HOLY SH**!... The iPhone 5 can now have 5 rows of icons. Too amazing. #sarcasm`") the original meaning. The traditional text mining techniques are incapable of correctly interpreting the true meaning of these untreated social media messages.

Regardless of all the useful applications that emerge from social media data, being able to automatically explicate the implicit social media data would not only increase the performance of the existing natural language processing techniques, but would also enable discovery of real important product features that exist in the implicit data.

Processing social media data has been one of the biggest challenges for researchers. Traditional natural language processing techniques that have been shown to work well on traditional documents are reported to fail or under-perform when applied on social media data, whose natures differ from traditional documents in the following ways:

1. **Social media data is high-dimensional, but sparse.** A unit of social media document (aka message) is short, containing only one or two sentences. Some social media services, such as Twitter, enforce the length of a message, urging the users to be creative and use their own combination of word forms to express their opinions within limited context. Traditional techniques for interpreting semantics from documents would fail on social media data due to insufficiency in textual content [7]. Furthermore, the high-dimensionality caused by using creative word forms would prevent such traditional techniques from finding semantic similarity among the pool of social media messages.
2. **Social media data is noisy.** Noise in social media data comes in multiple forms such as grammatical errors (e.g., `''In the middle of the day and takes off running''`), intentional/unintentional typographical errors (e.g., `''iphone 4s sooo COOOOLLLL!''`), and symbolic word forms (e.g., `'':-/''`, `''LOL''`). Since traditional text processing techniques assume documents to be well-formed and grammatically corrected [15], they would fail to operate on social media data.

Existing attempts to interpret the semantic meaning behind implicit social media and relevant kinds of data (i.e. product reviews) include machine learning based implicit sentence detection algorithms proposed by Tsur et al. [16, 17]. However, their methods only identify whether a piece of textual information is sarcastic or not. The work presented in this paper extends the previous literature by further extracting true meaning from social media messages whose context related to products/product features are implicit.

This paper presents a mathematical model based on the heterogeneous *co-word network* patterns in order to *translate* implicit context towards a particular product or product feature into the explicit equivalence. A co-word network (or word co-occurrence network) is a graph where each node represents a unique word, and an undirected edge represents the frequency of co-occurrence of the two words. In this work, the network is augmented to incorporate parts of speech into each word. The intuition behind using the co-word network is that even though a message may be implicit, the similar combination of the words may have been used by other users who express their messages more explicitly. For example, given an implicit message "`wow I have to` *`squint`* `to read this on the` *`screen`*", other users may have used the terms *squint* and *screen* in a more explicit context such as "`Don't make me squint @user - your mobile banner needs work on my` *`tiny screen`* `iPhone 5S.`" If the combination of the words *squint* and *screen* occurs in the messages that contain the word *tiny* frequently enough, then the system would be able to relate the original message to a more explicit set of terms. Particularly, the system would be able to interpret that the user thinks that the *screen* feature of this particular product is *small*.

Specifically, this paper has the following main contributions:

1. The authors adopt the usage of the co-word network in a product design context. The co-word network has shown to be useful in multiple semantic extraction applications in information retrieval literature [7, 18]. To the best of our knowledge, this technique has first been used in the design literature.
2. The authors propose a probabilistic mathematical model in order to map implicit product-related information in social media data into the equivalent explicit context.
3. The authors illustrate the efficacy of the proposed methodology using a case study of real world smartphone data and Twitter data.

2 Related Works

While the use of implicit language such as indirect speech and sarcasm has been well explored in multiple psycholinguistic studies [19–21], automatic semantic interpretation of implicit information in social networks is still in an infancy stage. This section first surveys the use of social media data pertaining to the product design applications, and then discusses existing natural language processing techniques that have been used to extract semantics from social media data.

2.1 Applications of Large Scale Social Media Data in Product Design Domain

Knowledge extracted from product-related, user-generated information has proved valuable in product design applications. Archak et al. proposed a set of algorithms, both fully automated and semi-automated, to extract opinionated product features from online reviews. The extracted information was successfully used to predict product demand [22]. While their findings were promising, the algorithms were applied on online product reviews whose nature is different from social media data, in terms of noise, amount of indirect language (i.e. sarcasm), and language creativity that do not conform to the standard English grammar. This research primarily aims to interpret semantics of a subset of social media data whose language is presented with sarcasm, that traditional natural language processing techniques would fail to handle effectively. Social media has recently been established as a viable source for product design and development. Previous studies claimed that knowledge extracted from social media data could be more beneficial than traditional product design knowledge sources such as product reviews (from popular online electronic commerce website such as Amazon.com, BestBuy.com, Walmart.com) and user study campaigns [2, 4, 5]. Asur et al. was able to use Twitter data collected during a 3 month period to predict the demand of theatre movies [23]. They claimed that the prediction results are more accurate than those of the Hollywood Stock Exchange. Their study also found that sentiments in tweets can improve the prediction after a movie has been released. Tuarob and Tucker found that social media data could be a potential data source for extracting user preferences towards particular products or product features [2, 5]. In a later work, they presented a methodology for automatic discovery of innovative users (aka. *lead users*) in online communities, using a set of mathematical models to extract latent features (product features not yet implemented in the market space), then identify lead users based on the volume of innovative features that they express in social media [1, 4]. Lim and Tucker proposed a Bayesian-based statistical sampling algorithm that identifies product-feature-related keywords from social media data, without human-labeled training data [6]. Recently, Stone and Choi presented a visualization tool which allows designers to extract useful insights from online product reviews [24].

Since all the above techniques rely on the assumption that social media data is explicit, these techniques would fail to correctly process implicit social media messages which could result in error or inaccurate results. With these emerging product design applications that rely on social media as a knowledge source, it is crucial that the algorithms behind these applications are able to correctly interpret the true meaning of the data.

2.2 Natural Language Technology for Semantic Interpretation in Social Media

In this subsection, technologies used to process social media data that go beyond just keyword detection (which works only on explicit data) are reviewed. Multiple studies in the Information Retrieval field have agreed that it is necessary to develop special text processing techniques for social media messages, since they are different from traditional documents due to smaller textual content, heterogeneous language standards, and higher level of noise [25–29].

Social media holds sentiments expressed by its users (primarily in the form of textual data). Sentiment analysis in social media refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in social media. Thelwall et al. found that important events lead to increases in average negative sentiment strength in tweets during the same period [30]. The authors concluded that negative sentiment may be the key to popular trends in Twitter. Kucuktunc et al. studied the influence of several factors such as gender, age, education level, discussion topic, and time of day on sentiment variation in Yahoo! Answers [31]. Their findings shed light towards an application on attitude prediction in online question-answering forums. Weber et al. proposed a machine learning based algorithm to mine *tips*, short, self-contained, concise texts describing non-obvious advice [32]. Lim et al. applied unsupervised sentiment analysis in social media to identify the patient's potential symptoms and latent infectious diseases [9]. Sentiment of each short text is extracted and used as part of the features. Even though sentiment analysis could prove to be useful when designers would like to know how customers feel about a particular product or product feature, most sentiment extraction techniques only output sentiment level in two dimension (i.e. Positive and Negative). Hence, more advanced techniques are needed in order to narrow down what actually the customers want to say.

Besides sentiment analysis, multiple studies have found that topical analysis could be useful when dealing with noisy textual data such as social media. Even though social media is high in noise due to the heterogeneity of the writing styles, formality, and creativity, such noise bears undiscovered wisdom of the crowd. Paul and Dredze utilized a modified Latent Dirichlet Allocation [33] model to identify 15 ailments along with descriptions and symptoms in Twitter data [34, 35]. Tuarob et al. proposed a methodology for discovering health related content in social media data by quantifying topical similarity between documents as a feature type [7,8]. Furthermore, a number of studies have devoted to using topical models to detect emerging trends in social networks [36–38]. In the design informatics field, Tuarob and Tucker proposed a set of methods that extract product related information from large scale social media data, such as customer demands, notable product features, and innovative product ideas [1, 2, 39]. The techniques mentioned above rely on explicit content of social media data and would likely fail or not produce correct results when applied on documents whose meanings are implicit.

Implicit document processing has posed challenges to computational linguists. Researchers have studied on the nature of implicit uses of language; however, none have successfully developed a computational model to translate implicit content into the equivalent explicit form. In dealing with implicit context in social media data, multiple algorithms have been proposed to detect the presence of implicit content in social media [16,40,41]; however, these algorithms do not further attempt to map the implicit content to the equivalent explicit forms. To the best of our knowledge, we are the first to explore the problem of identifying explicit customer preferences towards a product/product feature from large scale social media data.

3 Methodology

The method proposed in this paper mines language usages in the form of word co-occurrence patterns, in order to map implicit context commonly found in social media data to equivalent explicit ones. Figure 1 outlines the overview of the proposed methodology.

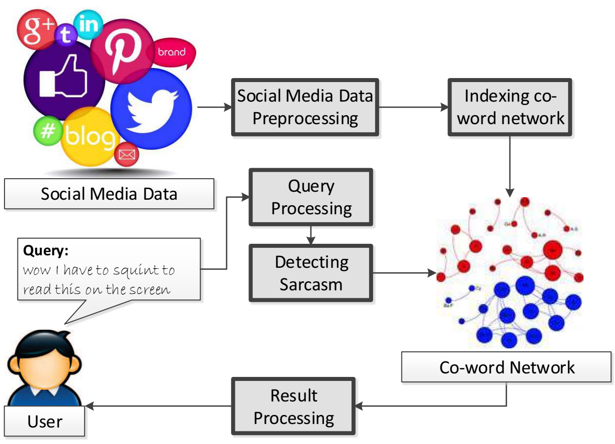

Fig. 1. Overview of the proposed system.

First, social media data is collected and preprocessed (Section 3.1). The textual content is then fed to the indexer in order to generate the co-word network (Section 3.2). Once the network is generated and indexed, the user could give the system an implicit message as the query. The query is processed and the results are returned to the user as a ranked list of relevant keywords classified by parts of speech (Section 3.4). In this system, the user could be a human designer, or an automated program that mines product related information from social media messages.

A practical usage of the proposed implicit message inference system would be to aid designers in synthesizing product features, mined from customers' feedback in large scale social media data, into the next generation products. A framework was presented in [2], where designers iteratively identify notably good and bad features from existing products, and incorporate/remove them from the next generation products. The method proposed in this paper could be incorporated into such a framework to improve the notable product feature extraction process. The following subsections will discuss each component in Figure 1 in detail.

3.1 Social Media Data Preprocessing

Social media provides a means for people to interact, share, and exchange information and opinions in virtual communities and networks [42]. For generalization, the proposed methodology minimizes the assumption about functionalities of social media data, and only assumes that a unit of social media is a tuple of unstructured textual content, a user ID, and a timestamp. Such a unit is referred to as a *message* throughout the paper. This minimal assumption would allow the proposed methodology to generalize across multiple heterogeneous pools of social media such as Twitter, Facebook, Google+, etc., as each of these social media platforms has this common data structure. Social media messages, corresponding to each product domain, are retrieved by a query of the product's name (and its variants) within the large stream of social media data.

3.1.1 Data Cleaning

Most social media crawling APIs provide additional information with each social media message such as user identification, geographical information, and other statistics [1]. Though this additional information could be useful, it is disregarded and removed not only to save storage space and improve computational speed, but also to preserve the minimal assumption about the social media data mentioned earlier.

Raw social media messages are full of noise that could prevent further steps from achieving the expected performance. In order to remove such noise, the data cleaning process does the following:

1. Lowercasing the textual content
2. Removing hashtags, usernames, and hyperlinks
3. Removing stop words[2]

Note that misspelled words (e.g. `hahaha`, `lovin`, etc.) and *emoticons* (e.g. `:-)`, `(")(-_-)(")`, etc.) are intentionally preserved. Even though they are not well-formed and do not exist in traditional dictionaries, they have been shown to carry useful information that infers semantic meaning behind the messages [8, 43]. Furthermore, unlike traditional preprocessing techniques for reducing noise in documents, the social media data is not *stemmed*, since previous studies have shown that stemming could excessively reduce the dimensionality of the data (especially in short messages, each of which contains roughly 14 words on average [44, 45]), and would likely result in poorer performance [7].

3.1.2 Sentiment Extraction

The technique developed by Thelwall et al. is employed to quantify the emotion in a message [43]. The algorithm takes a short text as an input, and outputs two values, each of which ranges from 1 to 5. The first value represents the *positive* sentiment level, and the other represents the *negative* sentiment level. The reason for having the two sentiment scores instead of just one (with $-/+$ sign representing negative/positive sentiment) is because research findings have determined that positive and negative sentiments can coexist [46]. However, in this research, we only focus on the net sentiment level; hence, the positive and negative scores are combined to produce an emotion strength score using the following equation:

$$Emotion\ Strength(ES) = PositiveScore - NegativeScore \quad (1)$$

A message is then classified into one of the 3 categories based on the sign of the Emotion Strength score (i.e. **positive (+ve)**, **neutral (0ve)**, **negative (-ve)**). The *EmotionStrength* scores will later be used to identify whether a particular message conveys a positive or negative attitude towards a particular product or product feature.

3.1.3 Feature Extraction.

Algorithm 1: The feature extraction algorithm from a collection of documents

Input: D: Set of free-text documents to extract product features.
Output: E: Set of extractions. Each $e \in E$ is a tuple of $\langle feature, frequency \rangle$, for example $e = \langle \text{'}onscreen\ keyboard', 5 \rangle$

preprocessing;
for $d \in D$ **do**
    Clean d ;
    POS tag d ;
    Extract multi-word features ;
end
initialization;
E = $\oslash$;
T = $\oslash$;
F = Seed Features ;
while *E can still grow* **do**
    Learn templates from seed features;
    Add new template to T;
    foreach $d \in D$ **do**
        foreach *Sentence* $s \in d$ **do**
            e $\leftarrow$ Extract potential feature-opinion pair using T;
            Add e to E ;
        end
    end
    Update F;
end
E $\leftarrow$ Clustering and normalizing features ;
return *E*;

Product features are extracted from each social media message. In this paper, the feature extraction algorithm used in [4] is employed. The pseudo-code of the algorithm is outlined in Algorithm 1. At a high level, the algorithm takes a collection of social messages corresponding to a product as input, and outputs a tuple of $\langle feature, frequency \rangle$ such as $\langle \text{'}onscreen\ keyboard', 5 \rangle$, which infers that the *on-screen keyboard* feature of this specific product was mentioned *5* times

[1] https://dev.twitter.com/rest/public

[2] Stop words are words that are filtered out before processing of textual information. Such words are typically too common to infer meaningful semantics. Examples of stop words include *the*, *is*, *at*, *which*, and *on*.

within the given corpus of social media messages. Interested readers are encouraged to consult [4] for additional details about the feature extraction algorithm.

The features are extracted because the proposed methodology infers explicit opinions towards a particular product feature, hence it is imperative that product features can automatically be identified.

3.1.4 Part of Speech Tagging.

Table 1. Node types and their descriptions.

Node Type	Description
PRODUCT	Smartphone model name
N	common noun
A	adjective
V	verb including copula, auxiliaries
^	proper noun
!	interjection
P	pre- or postposition, or subordinating conjunction
G	other abbreviations, foreign words, possessive endings, symbols, garbage
R	adverb
L	nominal + verbal (e.g. *i'm*), verbal + nominal (*let's*)
D	determiner
~	discourse marker, indications of continuation across multiple tweets
E	emoticon
O	pronoun (personal/WH; not possessive)
$	numeral
,	punctuation
&	coordinating conjunction
Z	proper noun + possessive
T	verb particle
S	nominal + possessive
X	existential *there* , predeterminers

The final step of the social media data preprocess is to tag each word in a social message with a part of speech (POS). In this paper, Carnegie Mellon ARK Twitter POS tagger[3] is used for this purpose. This particular POS tagger has not only been developed specially for social media data, but has also been successfully used in the product design domain [2].

The part of speech information is needed in order to disambiguate words with multiple meanings (i.e. homonyms) [47], which can be commonly found in social media. For example, the word "*cold*" in "`Who waits for an iphone5 in this` **`cold`** `weather?`" and "`I've got a` **`cold`** `this morning. will skip class.`" may have different meanings.

Each POS tag will become a node type in the co-word network. Besides standard linguistic POS tags offered by the POS tagger tool, a special node type *PRODUCT* is also introduced to distinguish a word that represents a product name (e.g. *iPhone 4*, *Samsung Galaxy SII*, *Nokia N9*, etc) from other words. Table 1 lists the node types used in this research, along with their descriptions.

3.2 Generating and Indexing Co-word Network

A co-word network is the collective interconnection of terms based on their paired presence within a specified unit of text. Traditional co-word networks represent a node with only textual representation of a word. Variants of co-occurrence networks have been used extensively in the Information Retrieval field in a wide range of applications that involve semantic analysis such as concept/trend emergence detection [48, 49], discovering new words, finding/clustering relevant items [50, 51], semantic interpretation [7, 52], and document annotation [53, 54].

In this paper, a node also incorporates part of speech information for word-sense disambiguation purposes. Concretely, a co-word network is an undirected, weighted graph where each node is a pair of $\langle Word, POS\ Tag\rangle$ (e.g., $\langle squint, V\rangle$ and $\langle iPhone\ 4, PRODUCT\rangle$) that represents a POS tagged word, and each edge weight is the frequency of co-occurrence. Let D be the set of all social media messages, and T be the vocabulary extracted from D. Formally, the co-word network G is defined as follows:

G= $\langle$V,E$\rangle$
V=$\{\langle Word, POS\ Tag\rangle \in$ T $\}$
E=$\{$(a,b)$|$ a,b $\in$ T$\}$
Weight(a,b)=$|\{$d $|$ d $\in$ D, (a,b)$\in$ E, d contains both a and b$\}|$

A *compound* is defined as a set of nodes. A social media message is converted to a compound by converting each word in the message into a node. The nodes are then combined. Replicated nodes are removed. Algorithm 2 explains how the co-word network is generated from a corpus of social media messages. First, the set of nodes, V, and the set of edges, E, are initialized to empty sets. For each social media message d in the corpus D, all the words are tagged with appropriated POS tags, and then converted into nodes which are then combined into a compound c. For each node n in the compound c, update V by including n. Then for each possible combination pair of nodes in c, the weight of the edge that links these two nodes is incremented by 1. The co-word network generation is finished once all the messages are processed. In this paper, the open-source graph database *Neo4J*[4] is used to store and index the network. Neo4J is used in this task due to its scalability that allows a network with millions of edges to be efficiently stored and indexed.

[3] http://www.ark.cs.cmu.edu/TweetNLP/

[4] http://neo4j.com/

Algorithm 2: The co-word generation algorithm from a collection of social media messages.

```
Input: D: Set of free-text social media messages.
Output: G: Co-word network.
initialization;
V = ∅ ;
E = ∅ ;
foreach Document d ∈ D do
    /*Extracting nodes from message*/ ;
    Compound c = ∅ ;
    foreach Word w ∈ d do
        Node n = ⟨w.text, w.pos⟩ ;
        Add n to c ;
    end
    /* Update V */ ;
    foreach Node n ∈ c do
        if n ∉ V then
            Add n to V ;
        end
    end
    /* Update E */ ;
    foreach Possible combination of word pair ⟨a,b⟩ in D
      do
        Edge e = (a,b) ;
        if e ∉ E then
            Add e to E ;
        end
        Increment e.weight by 1 ;
    end
end
return G = ⟨V,E⟩;
```

3.3 Sarcasm Detection

A majority of implicit social media data is manifested in the form of sarcastic messages. Maynard and Greenwood reported that roughly 22.75% of social media data is sarcastic [14]. Hence, this work focuses on improving the ability to interpret sarcastic product related social media messages. In the proposed framework, sarcastic messages are automatically discovered using a machine learning based sarcasm detection algorithm, implemented in [55]. The algorithm produces a sarcastic message detection model using the features extracted from the training data. These feature sets include:

N-grams: This feature set extracts individual words (uni-grams) and two consecutive words (bi-grams) from a given message. These n-gram features are used extensively to train classification models for text classification tasks. Three and more consecutive words are not used since research has shown the combination of uni-grams and bi-grams are sufficient and optimal, that yields the best results while consuming reasonable amounts of computing resources and memory [56].

Sentiment: It is a hypothesis that sarcastic messages are more negative than non-sarcastic ones. Mathematically,

$$h0 : sentiment_{neg}(sarcastic) > sentiment_{neg}(non\ sarcastic) \qquad (2)$$

Moreover, studies show that sarcastic messages tend to exhibit the co-existence of positive and negative sentiments [46]. The sentiment features include 1.) a positive and a negative sentiment score to each word in the message using the SentiWordNet[5] dictionary, and 2.) the sentiment score produced by the python library TextBlob[6].

Topics: The topical features are extracted using the Latent Dirichlet Allocation algorithm [33] implemented in gensim[7].

The training dataset includes 20,000 sarcastic tweets and 100,000 non-sarcastic tweets over a period of three weeks in June-July 2014. Once the features are extracted from the training data, they are used to train a support vector machine (SVM) classification model. The trained model is then used to identify a message whether it is sarcastic or non-sarcastic.

3.4 Query and Result Processing

A *query* is a free text message with implicit content. Example queries include "`I can't express how much I love the price of iPhone 5 on black Friday`" and "`I have to squint the screen to read this on Nokia N9`". This section describes how a user query is transformed into the network-compatible format, or a compound Q, for further processing. In particular, in order to process a free text query Q_{Text}, the following steps are performed:

1. Preprocess the query Q_{Text} using the mechanism described in Section 3.1, in order to clean the raw message, extract features, and assign POS tags.
2. Form the query compound Q, by converting each POS tagged word into a node, and combining them into a set.
3. Remove the nodes in Q that do not exist in the co-word network.

The resulting query compound Q is then fed into the system for further processing.

The implicit message translation problem in transformed into a *node ranking* problem so that traditional Information Retrieval techniques can be applied. In this context, a node in the co-word network is equivalent to a combination of a word and its POS. Given the set of products in the same domain (product space) $\mathbb{S}$, the set of all features (feature space) $\mathbb{F}$, the co-word network $G = \langle V, E \rangle$, and query compound Q. The node ranking algorithm takes the following steps:

STEP1 For each node $t \in V$, compute $P(t|Q, f, s)$, the likelihood (relevant to product features) of the node t given the

[5] http://sentiwordnet.isti.cnr.it/
[6] https://textblob.readthedocs.io/en/dev/
[7] https://radimrehurek.com/gensim/

query compound Q, target product feature $f \in \mathbb{F}$, and the product $s \in \mathbb{S}$.

STEP2 Rank the nodes by their likelihood.

STEP3 Top nodes are returned.

$P(t|Q,f,s)$ represents the likelihood that the node t is relevant to the feature f of the product s, given the query compound Q. The relevance of a node is quantified by its *relatedness* and *explicitness* to the query compound Q. Hence, mathematically $P(t|Q,f,s)$ is defined as follows:

$$P(t|Q,f,s) = \sum_{q \in Q} w_q \cdot Relatedness(t,q) \cdot Explicitness(t|q) \tag{3}$$

Where,

$$Relatedness(t,q) = \frac{weight(t,q)}{\sum_{n \in Adj(q)} weight(n,q)} \tag{4}$$

$$Explicitness(t|q) = \frac{degree(t)}{\sum_{n \in Adj(q)} degree(n)} \tag{5}$$

w_q is the weight for the node $q \in Q$, and $\sum_{q \in Q} w_q = 1$. $Adj(q)$ is the set of adjacent (neighbor) nodes to q. In the implementation, feature (i.e. f) and product nodes (i.e. s) are given twice the weight of other nodes in the compound. This is because, by giving higher weight to the target feature and product, the likelihood given to each node will be more relevant towards the feature of the product of interest. $weight(t,q)$ is the weight of the edge linking t and q, which is the co-occurrence frequency of the two nodes. Note that if t and q have never been mentioned together, then the $Relatedness(t,q)$ is evaluated to zero.

$Relatedness(t,q)$ hence quantifies how frequently t and q are mentioned together. The score is normalized to range between [0,1] for consistency when combined with other components.

$Explicitness(t|q)$ quantifies *explicitness* of the term represented by the node t when presented in the same context as the term represented by the node q, and is measured by the normalized degree of the node t. A term is explicit if it makes the context clearer or easier to understand to the readers. An intuitive assumption is made that terms that have explicit meanings tend to be commonly used and mentioned frequently in multiple contexts. Such properties are captured by the degree of the node representing the term, since the higher degree a node has, the more diverse it is co-mentioned with other words. Table 2 provides examples of 10 highest degree nodes and 10 lowest degree nodes, classified by parts of speech. From the example, it can be seen that words with high degrees have explicit meanings and would make the context simpler and more clarified. On the other hand, the words with low degrees tend to be spurious words that do not directly associate with the product domain. These words tend to make the context implicit, especially when talking about a product or product feature.

Finally, $P(t|Q,f,s)$ is then a weighted sum of the relevance between the node $t \in V$ and each node in the query compound Q. $P(t|Q,f,s)$ ranges between [0,1], using to approximate the probability of the node t being relevant to the query compound q. Once $P(t|Q,f,s)$ is computed for all the nodes in the co-word network, they can then be ranked using this score. The final output of the system would then be the top words classified by their parts of speech.

4 Case Study, Results, and Discussion

This section introduces a case study used to verify the proposed methodology and discusses the results.

A case study of 27 smartphone products is presented that uses social media data (Twitter data) to mine relevant product design information. Data pertaining to product specifications from the smartphone domain is then used to validate the proposed methodology. The selected smartphone models include *BlackBerry Bold 9900*, *Dell Venue Pro*, *HP Veer*, *HTC ThunderBolt*, *iPhone 3G*, *iPhone 3GS*, *iPhone 4*, *iPhone 4S*, *iPhone 5*, *iPhone 5C*, *iPhone 5S*, *Kyocera Echo*, *LG Cosmos Touch*, *LG Enlighten*, *Motorola Droid RAZR*, *Motorola DROID X2*, *Nokia E7*, *Nokia N9*, *Samsung Dart*, *Samsung Exhibit 4G*, *Samsung Galaxy Nexus*, *Samsung Galaxy S 4G*, *Samsung Galaxy S II*, *Samsung Galaxy Tab*, *Samsung Infuse 4G*, *Sony Ericsson Xperia Play*, and *T-Mobile G2x*.

Smartphones are used as a case study in this paper because of the large volume of discussion about this product domain in social media. Previous work also illustrated that social media data (i.e. Twitter) contains crucial information about product features of other more mundane products such as automobiles [2,39]. The proposed algorithms may not work well for products which are not prevalently discussed (in terms of quantity of messages related to the product) in social media as the corresponding sets of social media messages may be too small to extract useful knowledge from.

4.1 Social Media Data Collection

Twitter[8] is a microblog service that allows its users to send and read text messages of up to 140 characters, known as *tweets*. The Twitter dataset used in this research was collected randomly using the provided Twitter API, and comprises 2,117,415,962 (˜ 2.1 billion) tweets in the United States during the period of 31 months, from March 2011 to September 2013.

Tweets related to a product are collected by detecting the presence of the product name (and variants), and preprocessed by cleaning and mapping sentiment level as discussed in Section 3.1. Table 3 lists the number of tweets, number of unique Twitter users, and number of extracted features.

4.2 Co-word Network Generation

The co-word network is generated using the procedure outlined in Algorithm 2, using all the social media data associated with the 27 smartphone models. The resulting network contains 95,999 nodes and 2,288,723 edges. A node has a degree of 47.7 and is used 160 times on average. Table 4 lists the

[8] https://twitter.com/

Table 2. (Left) Top 10 nodes (words) with highest degree, classified by parts of speech. (Right) Bottom 10 nodes (words) with lowest degree, classified by parts of speech.

Highest degree nodes (words)				Lowest degree nodes (words)			
N	V	A	!	N	V	A	!
phone	got	good	lol	synergy	obstruct	contentious	heeh
case	need	free	haha	granddaddy	expel	uncomplicated	ofan
today	buy	great	ya	seeds	configure	sowable	wordddd
time	wait	cool	yeah	average	cook	disconnected	soz
day	love	bad	lmao	pervert	violate	democratic	lolololol
screen	gonna	nice	wow	hugs	deploy	doubtful	eiiishhh
people	make	long	hey	orphan	redirect	inappropriate	yayayayay
app	upgrade	happy	damn	swimsuit	bleed	practicamente	wujuuu
charger	sell	big	yo	chauffeur	impersonate	unrecoverable	oooou
camera	feel	fast	omg	paradigm	reign	heartbroken	naaaaaw

Table 3. Statistics of the Twitter data used in this paper, classified by smartphone products. See Appendix A for explanation of each statistic.

Model	Num Tweets	Num Users	Num Features	Feature Utilization	Feature Intensity	Average Feature Diversity
BlackBerry Bold 9900	308	252	126	1.7460	0.7143	0.0219
Dell Venue Pro	96	64	50	1.8800	0.9792	0.0380
HP Veer	143	110	76	1.7632	0.9371	0.0265
HTC ThunderBolt	1157	851	335	2.5522	0.7390	0.0071
iPhone 3G	2154	1874	532	2.8459	0.7029	0.0050
iPhone 3GS	3803	3119	775	3.0361	0.6187	0.0041
iPhone 4	68860	43957	6057	5.2196	0.4591	0.0010
iPhone 4S	63500	39145	5922	6.0750	0.5666	0.0010
iPhone 5	211311	124461	13493	7.8739	0.5028	0.0006
iPhone 5C	5533	4475	833	5.1477	0.7750	0.0046
iPhone 5S	15808	12417	1962	5.9210	0.7349	0.0023
Kyocera Echo	52	42	22	1.3636	0.5769	0.0877
LG Cosmos Touch	23	20	11	1.4545	0.6957	0.1313
LG Enlighten	18	17	5	1.6000	0.4444	0.2000
Motorola Droid RAZR	2535	1981	593	3.4840	0.8150	0.0056
Motorola DROID X2	471	378	162	2.1790	0.7495	0.0134
Nokia E7	26	18	14	1.2143	0.6538	0.0879
Nokia N9	208	153	83	1.7470	0.6971	0.0224
Samsung Dart	29	28	10	1.5000	0.5172	0.1071
Samsung Exhibit 4G	23	22	10	1.2000	0.5217	0.1333
Samsung Galaxy Nexus	5218	2988	1147	3.2476	0.7139	0.0031
Samsung Galaxy S 4G	188	152	62	2.0161	0.6649	0.0293
Samsung Galaxy S II	4599	3517	801	3.1436	0.5475	0.0042
Samsung Galaxy Tab	3989	2578	884	3.1912	0.7072	0.0033
Samsung Infuse 4G	284	215	85	2.2000	0.6585	0.0192
Sony Ericsson Xperia Play	481	325	132	1.9394	0.5322	0.0148
T-Mobile G2x	83	69	39	1.4359	0.6747	0.0351

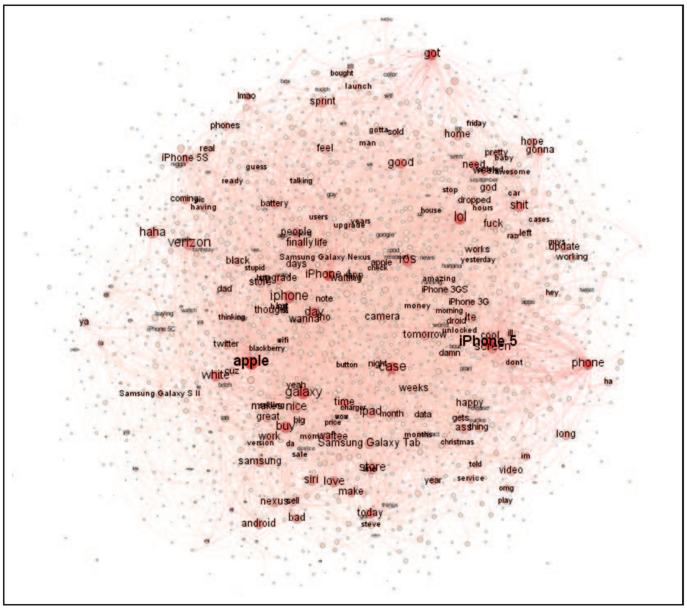

Fig. 2. Graphical visualization of the generated co-word network.

Table 4. Statistics of the co-word network generated using the Twitter data associate with the 27 smartphone products. The number of nodes and average degrees are categorized by the types of nodes.

Node Type	# of Nodes	Avg Degree	Node Type	# of Nodes	Avg Degree
PRODUCT	27	4589.17	~	92	22.63
N	24931	56.79	E	88	35.73
A	6169	64.93	O	452	50.16
V	13508	62.11	$	25	79.56
^	32562	30.62	,	35	20.43
!	4566	39.46	&	62	55.50
P	840	57.13	Z	90	15.47
G	9354	38.30	T	42	11.00
R	2325	48.09	S	25	10.28
L	432	66.48	X	13	16.38
D	361	79.07			

numbers and average degrees of nodes categorized by parts of speech.

Figure 2 illustrates a graphical visualization of the generated co-word network using the large-scale graph layout generation algorithm *OpenORD* [57].

4.3 Query and Result Processing

This section reports notable results from the proposed methodology.

Given a textual query with implicit content, the system first transforms it into a compound, by removing stop words and converting each remaining distinct word into a node. For example, a textual query "`I have to squint the screen to read this on Nokia N9`" would be translated into the compound { $\langle read, V\rangle$, $\langle squint, V\rangle$,

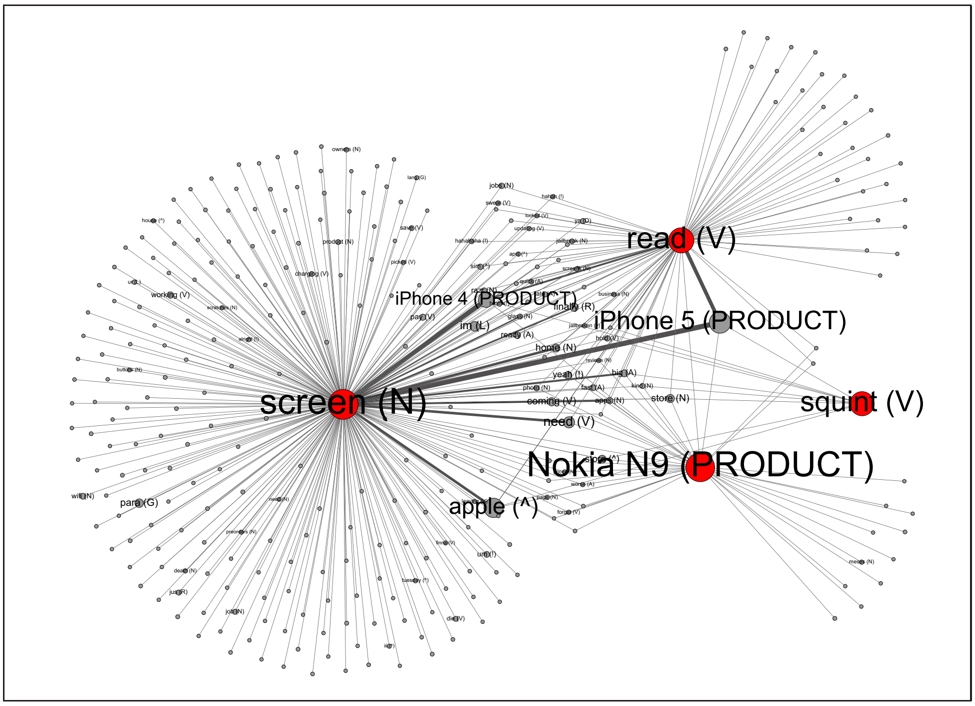

Fig. 3. Graphical example of the words co-occurring with the query compound.

$\langle screen, N\rangle$, $\langle Nokia\ N9, PRODUCT\rangle$ }. Note that not all the words in the query are converted into nodes since they could be stop words (e.g., `I`, `have`, `the`, `this`, and `on`). Figure 3 shows part of the generated co-word network where all the nodes co-occur with the queries nodes (red nodes). The thickness of the edges are proportionate to the actual edge weight. Similarly, the size of each node represents its relative degree.

4.3.1 Experiment Procedure

Product	# Negative	# Neutral	# Positive	All
HTC ThunderBolt	5	4	4	13
Motorola Droid	4	20	11	35
Samsung Galaxy	22	46	20	88
iPhone 3	7	15	3	25
iPhone 5	34	95	47	176
iPhone 4	35	38	52	125
Avg	17.83	36.33	22.83	77

Table 5. Number of tweets, categorized by hand-labelled sentiment (Negative, Neutral, Positive), associated with each selected smartphone model.

Six smartphone models are selected for evaluation of the proposed co-word implicit message translation model, including *HTC ThunderBolt*, *Motorola Droid*, *Samsung Galaxy*, *iPhone 3*, *iPhone 5*, *iPhone 4*. The sarcasm detection algorithm described in Section 3.3 is used to select sarcastic messages associated with each select smartphone model. To establish ground-truth validation data, each message (and the focus *product feature*) is manually tagged with *actual sentiment* (negative, neutral, or positive) of the message poster towards such a feature. For example, "`I love how everyone with an iPhone 5 says 'look! My camera is 8 megapixels.' No. F*** off. Both of my Evo's have had 8 megapixel camera's.`" is associated with *iPhone 5*, and is tagged with $\langle camera, Negative\rangle$, meaning that the poster may actually feel negative (i.e. unsurprised) about the *camera* feature of the *iPhone 5*. Table 5 list the number of sarcastic messages, manually classified by its *actual sentiment*, associated with each selected smartphone model.

The evaluation is designed to compare the performance between the proposed co-word implicit message translation model (Co-word) against a baseline (Baseline). The baseline method returns the original sarcastic message without any modification (hence, the message is not semantically processed with the co-word network shown in Figure 1). Such comparison would allow us to see if the proposed Co-word model could translate a given sarcastic message into its explicit form. To compare the efficacy of both methods, this problem is transformed into a classification problem, where both the Co-word and Baseline translated versions are classified based on the sentiment (Negative, Neutral, Positive) using the sentiment extraction algorithm described in Section 3.1.2, and compared with the ground truth actual sentiment. Standard information retrieval evaluation metrics are used, including precision, recall, and F-measure. These metrics have been used extensively to validate the quality of the results of classification algorithms

[58].

For each sentiment class $c \in \{$Negative, Neutral, Positive$\}$, let $CC(c)$ denote the number of sarcastic messages correctly classified as c, $CA(c)$ denote the number of sarcastic messages classified as c, and $N(c)$ denote the number of sarcastic messages labelled as class c, these metrics are defined as follows:

$$precision(c) = \frac{CC(c)}{CA(c)} \quad (6)$$

$$recall(c) = \frac{CC(c)}{N(c)} \quad (7)$$

$$F-measure(c) = 2 \cdot \frac{precision(c) * recall(c)}{precision(c) + recall(c)} \quad (8)$$

Recall is the ratio of a number of messages the classifier can correctly recall to a number of all messages in that class. If there are 10 messages that belong to the class c, and a classifier can recall all 10 messages correctly, then the recall of the classification with respect to class c is 1.0 (100%). If the classifier can recall 7 messages correctly, then the recall ratio is 0.7 (70%). Precision is the ratio of a number of messages the classifier correctly recalls to a number of all messages it recalls (mix of correct and wrong recalls). In other words, precision quantifies how precise of the recalled results. F-measure combines precision and recall into one number with equal weights. Note that, precision, recall, and F-measure range from [0,1].

4.3.2 Experiment Results

Table 6 reports the sentiment classification results from the messages translated by the co-word method and the baseline for each select smartphone model. The classification results for each class (Positive, Neutral, Negative) of both the methods are displayed. The **bold** figures denotes the better result between the co-word and the baseline methods. Figure 4 summarizes the classification performance of the six select smartphone models, grouped by precision, recall, and F-measure of the three sentiment classes.

Figure 5 emphasizes the comparison between the F-measure of the classification results of sarcastic messages translated by the co-word and the baseline methods. The messages translated by the proposed co-word method improves the sentiment extraction algorithm to identify the true *negative* sentiment for four out of six products, namely *Motorola Droid*, *iPhone 3*, *iPhone 5*, *iPhone 4*. The reason why the co-word method performs worse than the baseline could be explained by Figure 4. For negative class, though the overall recall of the co-word method surpasses that of the baseline by +39.46%, the precision suffers from the deterioration of -21.21%. This phenomenon suggests that the co-word method still misinterprets some of the actual positive messages as negative (since the recall for the positive class also drops by -34% according to Figure 4), and hence introduce false positives to the sentiment classifier. Regardless, the overall performance in terms of F-measure is improved by 14% for the negative class.

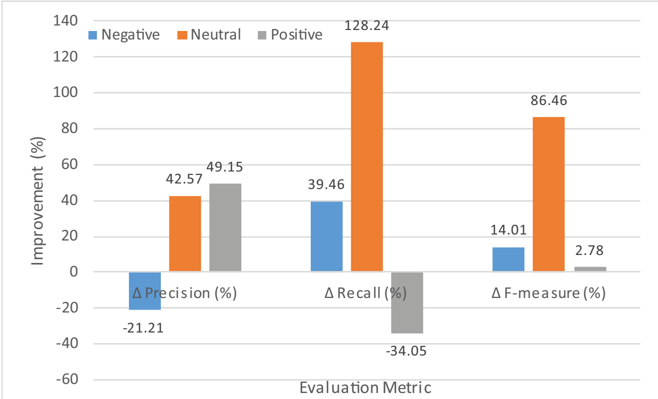

Fig. 4. Improvement of the sentiment classification results, grouped by precision, recall, and F-measure, of each sentiment class (Negative, Neutral, Positive) when translated the sarcastic messages with the co-word method.

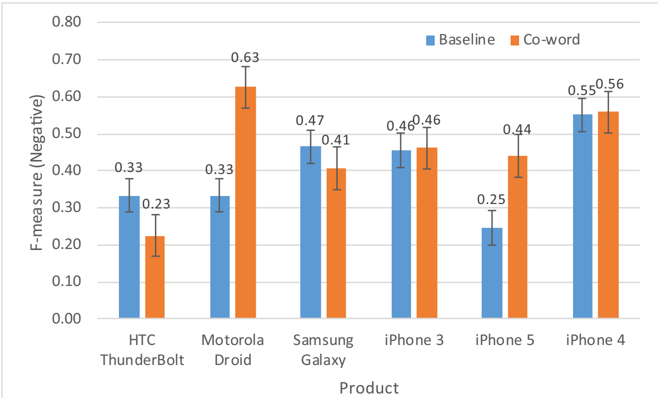

Fig. 5. Comparison of F-measure evaluation of the class *Negative*, for each selected smartphone model.

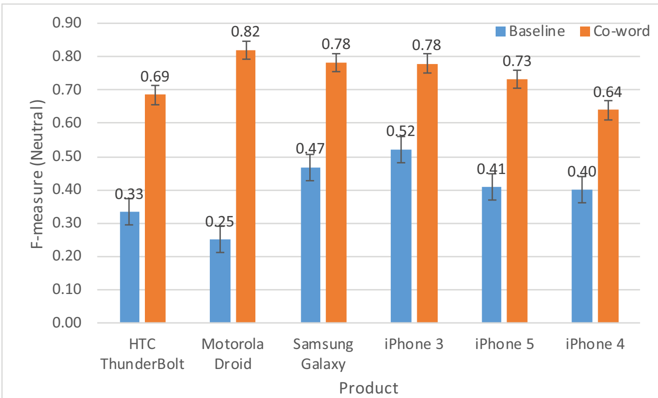

Fig. 6. Comparison of F-measure evaluation of the class *Neutral*, for each selected smartphone model.

Figure 6 compares the F-measure of the co-word and baseline of the neutral class. Evidently, the proposed co-word method allows the sentiment classifier to correctly interpret the actual neutral sentiment of a sarcastic message in

Method	Baseline									Co-word								
Class	Negative (-)			Neutral (0)			Positive (+)			Negative (-)			Neutral (0)			Positive (+)		
Product	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
HTC ThunderBolt	1.00	0.20	**0.33**	0.25	0.50	0.33	**0.50**	**0.50**	**0.50**	**0.23**	**0.22**	0.23	**0.52**	**1.00**	**0.69**	0.30	0.34	0.32
Motorola Droid	0.25	0.50	0.33	0.33	0.20	0.25	0.20	0.27	0.23	**0.46**	**1.00**	**0.63**	**0.84**	**0.81**	**0.82**	**0.34**	**0.32**	**0.33**
Samsung Galaxy	**0.48**	**0.45**	**0.47**	0.54	0.41	0.47	0.31	**0.50**	0.38	0.38	0.44	0.41	**0.73**	**0.84**	**0.78**	**0.56**	0.38	**0.45**
iPhone 3	**0.50**	0.42	0.46	0.75	0.40	0.52	0.18	**0.67**	0.29	0.41	**0.53**	**0.46**	**0.78**	**0.78**	**0.78**	**0.34**	0.33	**0.34**
iPhone 5	0.23	0.26	0.25	0.54	0.33	0.41	0.31	**0.53**	0.39	**0.37**	**0.55**	**0.44**	**0.75**	**0.72**	**0.73**	**0.72**	0.37	**0.49**
iPhone 4	**0.56**	0.54	0.55	0.48	0.34	0.40	0.59	**0.73**	**0.66**	0.53	**0.59**	**0.56**	**0.52**	**0.83**	**0.64**	**0.87**	0.38	0.53
Avg	**0.50**	0.40	0.40	0.48	0.36	0.40	0.35	**0.53**	0.41	0.40	**0.55**	**0.45**	**0.69**	**0.83**	**0.74**	**0.52**	0.35	**0.41**

Table 6. Comparison of the classification performance between the proposed co-word based method and the baseline (no translation process), for each sentiment class. P denotes Precision, R denotes recall, and F denotes F-measure.

all the select six smartphone models. Figure 4 further elaborates this phenomenon by showing the improvement in precision (by +42.57%), recall (by +128.24%), and F-measure (by +86.46%).

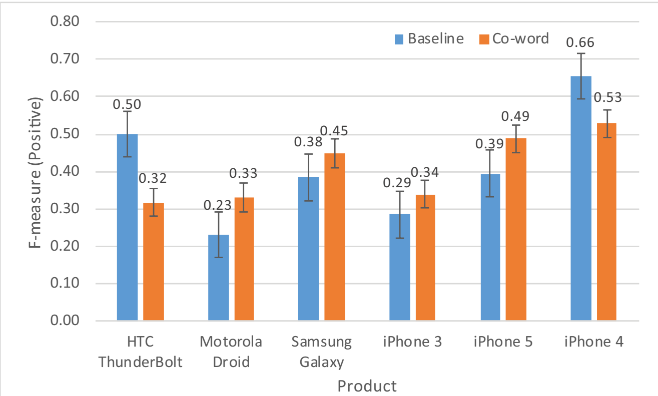

Fig. 7. Comparison of F-measure evaluation of the class *Positive*, for each selected smartphone model.

Figure 7 compares the F-measure of both the co-word and baseline of the sentiment classification on the positive class. The co-word method improve the sentiment classification in four out of six smartphone models including *Motorola Droid*, *Samsung Galaxy*, *iPhone 3*, and *iPhone 5*. The reason why the co-word method is not the clear winner for all the select products for the positive sentiment messages is because the co-word technique still tends to misinterpret some of the positive sentiment messages as negative ones. As a result, the amount of the translated messages that appear to be positive is limited, causing a drop in recall of -34% on average (according to Figure 4). Regardless, since the co-word method is selective about the positive class, the precision of the positive class is boosted by +49.15%, leading the F-measure of the positive class to improve by +2.78%.

Overall, the classification results in terms of F-measure are improved on average for all the three sentiment classes (+14% for positive class, +86.46% for neutral class, and +2.78% for positive class). The experiment results presented in this section not only illustrate that the co-word technique has the potential to facilitate the translation of implicit social media messages, so that they can be further processed by traditional natural language processing techniques, but also shed light on room to improve and extend this proposed framework for design applications that rely on knowledge extracted from large scale social media data such as [1–3].

Table 7 illustrates the actual results from the proposed methodology on 10 sample social media messages whose preferences associated with the target product features are implicit (i.e. in the form of sarcasm). The table lists actual Twitter messages, target features, manual interpretation (by the authors) and the resulting top 3 relevant keywords (out of 10 keywords returned by the system), classified by parts of speech. Only nouns (N), verbs (V), and adjectives (A) are showed since the combination of these words are mostly sufficient in order to interpret the explicit semantic behind each message.

From the sample results, it is evident that the combination of the top words returned by the system could potentially provide explicit meaning of the implicit message. For example, the meaning behind "`I can't express how much I love the price of iPhone 5 on black Friday`" may infer that the user would like to *buy* and iPhone 5 *today* (which may be a Black Friday) because the price is *cheap*. Similarly, the user who posts "`eh DroidRazr HD resolution? I don't think so.`" may convey that the *display* of his/her Droid Razr is *bad*, and needs to be *upgraded*.

Most traditional semantic interpretation techniques including sentiment analysis assume that documents are explicit, and would fail when dealing with these implicit social media messages. The Column "*Sentiment Level (From Implicit Context)*" shows quantified sentiment level using the algorithm described in Section 3.1.2 on the original tweets. The actual *Emotional Strength* scores are in parentheses. The Column "*Manual Sentiment Evaluation*" lists the manual evaluation by the authors on the actual sentiment that each sample tweet infers towards the target product features (either *Positive* or *Negative*). The Column "*Sentiment Level (From Translated Explicit Context)*" shows the sentiment level using the same sentiment extraction algorithm, but on the translated explicit content generated by concatenating the top 20 keywords returned by the system into a single text (disregarding parts of speech). The sentiment levels computed on the translated text agree with the

Table 7. Sample results of 10 sarcastic product-related tweets.

Sample Number	Tweet	Target Product Feature	Manual Context Interpretation	Manual Sentiment Evaluation	Sentiment Level (From Implicit Context)	Explicit Translation			Sentiment Level (From Translated Explicit Context)
						Nouns	Verbs	Adjectives	
1	So the new #iPhone5s is getting a new "Home" button? Hmm...something #Android phones have had for ages. Way to go #Apple! Way to innovate!	Home Button	The home button feature is not impressive.	Negative	Neutral (0)	home button shit	need lost sold	slow black wrong	**Negative (-1)**
2	Every time I see the Droid Razr commercial, all I can think is, "You know what a great product doesn't need? Gimmicks." #apple #droidrazr	Commercial	The comercial is not relevant.	Negative	Positive (2)	life battery gimmicks	make have hate	bad real funny	**Negative (-2)**
3	Heard this from a Best Buy Mobile employee to a customer: "The only difference between the iPhone 4 and 4S is Siri." #impartial #informed	Siri	Adding Siri to iPhone 4S is not impressive.	Negative	Positive (1)	day case shit	wait need buy	big great bad	**Negative (-2)**
4	Here's A Phone I Wouldn't Mind Getting-- Nokia N9	Nokia N9	I want Nokia N9.	Positive	Neutral (0)	case screen camera	need want buy	good great awesome	**Positive (1)**
5	That's so true! "if it looks the same how will people know I upgraded?" (iPhone 4 vs iPhone 4s)	Looks	iPhone 4s should have different appearance from its predecessor.	Negative	Positive (1)	people week shit	wait need hate	good white free	**Negative (-1)**
6	I can't express how much I love the price of iPhone 5 on black Friday	price	Price is cheap.	Positive	Neutral (0)	today sale price	need wait buy	good free cheap	**Positive (3)**
7	I LOVE when my iPhone 5 charger stops working #deepsarcasm Have to use my moms iPhone	Charger	Charger is easily broken.	Negative	Positive (2)	case iphone time	hate borrow wait	stupid fast bad	**Negative (-2)**
8	eh DroidRazr HD resolution? I don't think so.	Resolution	Resolution is not HD.	Negative	Neutral (0)	screen camera case	check sucks buying	free sweet slow	**Negative (-1)**
9	"Samsung Galaxy S II Sprint Epic 4G Touch." Yes that's a real name for a phone. Incredible.	Name	The name is not suitable.	Negative	Neutral (0)	galaxy day shit	need make sucks	bad fast true	**Negative (-1)**
10	New iCloud? sounds a lot like the SkyDrive that I used long before the iPhone 4s came out. Yay for Apple "innovation"	iCloud	iCloud is not an innovative feature.	Negative	Positive (1)	lot photo time	sounds need wait	cool bad crazy	**Negative (-1)**

manual evaluation in all the samples shown in Table 7.

Not surprisingly, the sentiment level extracted from the original texts are all incorrect, since the sentiment extraction technique is designed to detect *explicit* sentiment, and hence would not give correct results when dealing with sarcasm or vague context. It is also interesting to note that the sentiment computed for the implicit sample messages tend to be neutral (Sentiment Level ≈ 0), regardless of the fact that they are composed with emotion-inspired words (i.e., *love*, *can't*, *shit*, *beautifully*, *incredible*, etc.). This agrees with prior findings that messages with implicit sentiment (i.e. sarcasm) would be sentimentally neutralized since such messages tend to have equally high volumes of both *Positive* and *Negative* scores, causing the *Emotion Strength* score to converge to 0 [59].

5 Conclusions and Future Works

This paper proposes a knowledge based methodology for inferring explicit sense from social media messages whose connotations related to products/product features are implicit. The methodology first generates a co-word network from the corpus of social media messages, which is used as the knowledge source that captures the relationship among all the words expressed in the stream of large scale social media data. A set of mathematical formulations are proposed in order to identify a combination of keywords that would best infer explicit connotation to a given implicit message query. A case study of real-world 27 smartphone models with 31 months' worth of Twitter data is presented. The results of selected smartphone models show great promises that the proposed methodology is effective in translating implicit product preferences to their explicit equivalent connotation that could be readily used in further knowledge extraction applications such as synthesizing product features [2], predicting future product demands and long-term product longevity [5], and identifying innovative users in online communities [4]. Future works could strengthen the evaluation process by involving user studies, and verify the generalizability of the proposed methodology by examining diverse case studies of different product domains and social media services. Machine learning approaches that process psychological information such as [60] will also be explored to predict behaviors of customers from their sarcasm and other forms of language usages.

Acknowledgements

This research project is supported by Mahidol University. Suppawong Tuarob is the corresponding author. We are also grateful for help with implementation and experimentation from Lawrence Lee.

References

[1] Tuarob, S., and Tucker, C. S., 2015. "Automated discovery of lead users and latent product features by mining

large scale social media networks". *Journal of Mechanical Design,* **137**(7), p. 071402.

[2] Tuarob, S., and Tucker, C. S., 2015. "Quantifying product favorability and extracting notable product features using large scale social media data". *Journal of Computing and Information Science in Engineering,* **15**(3), p. 031003.

[3] Tuarob, S., and Tucker, C. S., 2015. "A product feature inference model for mining implicit customer preferences within large scale social media networks". In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V01BT02A002–V01BT02A002.

[4] Tuarob, S., and Tucker, C. S., 2014. "Discovering next generation product innovations by identifying lead user preferences expressed through large scale social media data". In Proceedings of ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference 2014, ASME.

[5] Tuarob, S., and Tucker, C. S., 2013. "Fad or here to stay: Predicting product market adoption and longevity using large scale, social media data". In Proc. ASME 2013 Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.(IDETC/CIE2013).

[6] Lim, S., and Tucker, C. S., 2016. "A bayesian sampling method for product feature extraction from large-scale textual data". *Journal of Mechanical Design,* **138**(6), p. 061403.

[7] Tuarob, S., Tucker, C. S., Salathe, M., and Ram, N., 2014. "An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages". *Journal of Biomedical Informatics*.

[8] Tuarob, S., Tucker, C. S., Salathe, M., and Ram, N., 2013. "Discovering health-related knowledge in social media using ensembles of heterogeneous features". In Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management, CIKM '13, ACM, pp. 1685–1690.

[9] Lim, S., Tucker, C. S., and Kumara, S., 2016. "An unsupervised machine learning model for discovering latent infectious diseases using social media data". *Journal of Biomedical Informatics*.

[10] Sakaki, T., Okazaki, M., and Matsuo, Y., 2010. "Earthquake shakes twitter users: real-time event detection by social sensors". In Proceedings of the 19th international conference on World wide web, WWW '10, ACM, pp. 851–860.

[11] Caragea, C., McNeese, N., Jaiswal, A., Traylor, G., Kim, H., Mitra, P., Wu, D., Tapia, A., Giles, L., Jansen, B., et al., 2011. "Classifying text messages for the haiti earthquake". In Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM2011).

[12] Bollen, J., Mao, H., and Zeng, X., 2011. "Twitter mood predicts the stock market". *Journal of Computational Science,* **2**(1), pp. 1–8.

[13] Zhang, X., Fuehres, H., and Gloor, P., 2012. "Predicting asset value through twitter buzz". *Advances in Collective Intelligence 2011*, pp. 23–34.

[14] Maynard, D., and Greenwood, M. A., 2014. "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis". In Proceedings of LREC.

[15] Dey, L., and Haque, S., 2009. "Studying the effects of noisy text on text mining applications". In Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data, ACM, pp. 107–114.

[16] Tsur, O., Davidov, D., and Rappoport, A., 2010. "Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.". In ICWSM.

[17] Davidov, D., Tsur, O., and Rappoport, A., 2010. "Semi-supervised recognition of sarcastic sentences in twitter and amazon". In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, pp. 107–116.

[18] Navigli, R., and Velardi, P., 2005. "Structural semantic interconnections: a knowledge-based approach to word sense disambiguation". *IEEE transactions on pattern analysis and machine intelligence,* **27**(7), pp. 1075–1086.

[19] Muecke, D. C., 1982. *Irony and the Ironic*. Methuen.

[20] Gibbs, R. W., 1986. "On the psycholinguistics of sarcasm.". *Journal of Experimental Psychology: General,* **115**(1), p. 3.

[21] Gibbs, R. W., and Colston, H. L., 2007. *Irony in language and thought: A cognitive science reader*. Psychology Press.

[22] Archak, N., Ghose, A., and Ipeirotis, P. G., 2011. "Deriving the pricing power of product features by mining consumer reviews". *Management science,* **57**(8), pp. 1485–1509.

[23] Asur, S., and Huberman, B. A., 2010. "Predicting the future with social media". In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, Vol. 1, IEEE, pp. 492–499.

[24] Stone, T., and Choi, S.-K., 2014. "Visualization tool for interpreting user needs from user-generated content via text mining and classification". In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A009–V02AT03A009.

[25] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X., 2011. "Comparing twitter and traditional media using topic models". In *Advances in Information Retrieval*. Springer, pp. 338–349.

[26] YaJuan, D., WEIF uRu, C. Z., Heung, Z. M., and SHUM, Y., 2012. "Twitter topic summarization by ranking tweets using social influence and content quality". In Proceedings of the 24th International Conference on Computational Linguistics, pp. 763–780.

[27] Wang, Y., Wu, H., and Fang, H., 2014. "An exploration of tie-breaking for microblog retrieval". In *Advances in*

Information Retrieval. Springer, pp. 713–719.

[28] Tuarob, S., Tucker, C. S., Salathe, M., and Ram, N., 2015. “Modeling individual-level infection dynamics using social network information”. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, pp. 1501–1510.

[29] Tuarob, S., and Mitrpanont, J. L., 2017. “Automatic discovery of abusive thai language usages in social networks”. In International Conference on Asian Digital Libraries, Springer, pp. 267–278.

[30] Thelwall, M., Buckley, K., and Paltoglou, G., 2011. “Sentiment in twitter events”. *J. Am. Soc. Inf. Sci. Technol.*, pp. 406–418.

[31] Kucuktunc, O., Cambazoglu, B. B., Weber, I., and Ferhatosmanoglu, H., 2012. “A large-scale sentiment analysis for yahoo! answers”. In Proceedings of the fifth ACM international conference on Web search and data mining, WSDM ’12, ACM, pp. 633–642.

[32] Weber, I., Ukkonen, A., and Gionis, A., 2012. “Answers, not links: extracting tips from yahoo! answers to address how-to web queries”. In Proceedings of the fifth ACM international conference on Web search and data mining, WSDM ’12, ACM, pp. 613–622.

[33] Blei, D. M., Ng, A. Y., and Jordan, M. I., 2003. “Latent dirichlet allocation”. *J. Mach. Learn. Res.*, **3**, Mar., pp. 993–1022.

[34] Paul, M. J., and Dredze, M., 2011. A model for mining public health topics from twitter. Tech. rep.

[35] Paul, M. J., and Dredze, M., 2011. “You are what you tweet: Analyzing twitter for public health.”. In ICWSM, pp. 265–272.

[36] Ramage, D., Dumais, S. T., and Liebling, D. J., 2010. “Characterizing microblogs with topic models.”. *ICWSM*, **10**, pp. 1–1.

[37] Prier, K. W., Smith, M. S., Giraud-Carrier, C., and Hanson, C. L., 2011. “Identifying health-related topics on twitter”. In *Social computing, behavioral-cultural modeling and prediction*. Springer, pp. 18–25.

[38] Jin, O., Liu, N. N., Zhao, K., Yu, Y., and Yang, Q., 2011. “Transferring topical knowledge from auxiliary long texts for short text clustering”. In Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, pp. 775–784.

[39] Tuarob, S., and Tucker, C. S., 2016. “Automated discovery of product preferences in ubiquitous social media data: A case study of automobile market”. In Computer Science and Engineering Conference (ICSEC), 2016 International, IEEE, pp. 1–6.

[40] González-Ibáñez, R., Muresan, S., and Wacholder, N., 2011. “Identifying sarcasm in twitter: a closer look”. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, Association for Computational Linguistics, pp. 581–586.

[41] Reyes, A., Rosso, P., and Veale, T., 2013. “A multidimensional approach for detecting irony in twitter”. *Language Resources and Evaluation*, **47**(1), pp. 239–268.

[42] Ahlqvist, T., 2008. *Social media roadmaps: exploring the futures triggered by social media*. VTT.

[43] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A., 2010. “Sentiment in short strength detection informal text”. *J. Am. Soc. Inf. Sci. Technol.*, **61**(12), Dec., pp. 2544–2558.

[44] Guo, W., Li, H., Ji, H., and Diab, M. T., 2013. “Linking tweets to news: A framework to enrich short text data in social media.”. In ACL (1), Citeseer, pp. 239–249.

[45] Ramaswamy, S. “Comparing the efficiency of two clustering techniques”.

[46] Fox, E., 2008. *Emotion science: cognitive and neuroscientific approaches to understanding human emotions*. Palgrave Macmillan.

[47] Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P., 1992. “A practical part-of-speech tagger”. In Proceedings of the Third Conference on Applied Natural Language Processing, ANLC ’92, Association for Computational Linguistics, pp. 133–140.

[48] Özgür, A., Cetin, B., and Bingol, H., 2008. “Co-occurrence network of reuters news”. *International Journal of Modern Physics C*, **19**(05), pp. 689–702.

[49] Jia, S., Yang, C., Liu, J., and Zhang, Z., 2012. “An improved information filtering technology”. In *Future Computing, Communication, Control and Management*. Springer, pp. 507–512.

[50] Tuarob, S., Mitra, P., and Giles, C. L., 2012. “Improving algorithm search using the algorithm co-citation network”. In Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL ’12, ACM, pp. 277–280.

[51] Tuarob, S., Bhatia, S., Mitra, P., and Giles, C., 2013. “Automatic detection of pseudocodes in scholarly documents using machine learning”. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, pp. 738–742.

[52] Evans, D. A., Handerson, S. K., Monarch, I. A., Pereiro, J., Delon, L., and Hersh, W. R., 1998. *Mapping vocabularies using latent semantics*. Springer.

[53] Tuarob, S., Pouchard, L. C., and Giles, C. L., 2013. “Automatic tag recommendation for metadata annotation using probabilistic topic modeling”. In Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL ’13, ACM, pp. 239–248.

[54] Tuarob, S., Pouchard, L., Mitra, P., and Giles, C., 2015. “A generalized topic modeling approach for automatic document annotation”. *International Journal on Digital Libraries*, pp. 1–18.

[55] Cliche, M., 2014 (accessed February 19, 2017). *The sarcasm detector: Learning sarcasm from tweets!*

[56] Liu, F., Liu, F., and Liu, Y., 2008. “Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion”. In Spoken Language Technology Workshop, 2008. SLT 2008. IEEE, IEEE, pp. 181–184.

[57] Martin, S., Brown, W. M., Klavans, R., and Boyack, K. W., 2011. "Openord: An open-source toolbox for large graph layout". In IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, pp. 786806–786806.

[58] Manning, C. D., Raghavan, P., and Schütze, H., 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[59] Thelwall, M., 2013. "Heart and soul: Sentiment strength detection in the social web with sentistrength". *Cyberemotions*, pp. 1–14.

[60] Tuarob, S., Tucker, C. S., Kumara, S., Giles, C. L., Pincus, A. L., Conroy, D. E., and Ram, N., 2017. "How are you feeling?: A personalized methodology for predicting mental states from temporally observable physical and behavioral information". *Journal of Biomedical Informatics*, **68**, pp. 1 – 19.

[61] Tuarob, S., Pouchard, L. C., Noy, N., Horsburgh, J. S., and Palanisamy, G. "Onemercury: Towards automatic annotation of environmental science metadata". In Proceedings of the 2nd International Workshop on Linked Science 2012.

A Statistics for Feature Characteristics

The statistics used to describe the characteristics of the product features extracted from the social media data are described in this section. Given a product $s \in \mathbb{S}$, social media message corpus M_s, and the set of extracted features $F(M_s)$, *Feature Utilization*, *Feature Intensity*, and *Feature Diversity* are defined below:

A.1 Feature Utilization

For a given product $s \in \mathbb{S}$, the *feature utilization* is defined as:

$$F - Utilization(s) = \frac{\sum_{f \in F(M_s)} |\{m \in M_s : f \in m\}|}{|F(M_s)|} \tag{9}$$

The feature utilization quantifies how frequently on average a product feature is mentioned. The notion was first used in [53, 61] as the *Tag Utilization* metric, and was used to measure how *solid* and *standardized* a tag collection is. Similarly, the *Feature Utilization* measures, overall, how standardized the features of a specific product are.

From Table 3, the products with highest feature utilization are *iPhone 5*, *iPhone 4S*, *iPhone 5S*, *iPhone 4*, *iPhone 5C*, *Motorola Droid RAZR*, and *Samsung Galaxy Nexus* respectively. It does not come to a surprise to see the *iPhone* product line having high feature utilization since the product line has been in the market space for a long time and most features are inherited from the very first generation (such as *tough screen*, *home button*, *color (black/white)*, etc.). After generations, these features may have become standardized as opposed to products with newly emerging features such as *Kyocera Echo* (F-Utilization = 1.36) which distinctly offers two screens and the ability to use two applications at once.

A.2 Feature Intensity

Given a product s, the *feature intensity* is defined as:

$$F - Intensity(s) = \frac{|\bigcup_{f \in F(M_s)} \{m \in M_s : f \in m\}|}{|M_s|} \tag{10}$$

While feature utilization quantifies the overall quality of the features of a product, the feature intensity quantifies the volume of discussion in social media about the product features. It is measured by the proportion of messages related to the features of the product s over all the messages related to s. The feature intensity can infer how many of the consumers care to discuss about the product that they are using.

Interestingly, most of the *iPhone* products (except newly launched *iPhone 5S* and *iPhone 5C*) are among the smartphone products with lowest *feature intensity* scores. This might be because such products may ave been perceived by the consumers as generally *good* by word of mouth, which induce other consumers to purchase such products without much consideration about the features before making the purchasing decisions.

A.3 Feature Diversity

The feature diversity tells how diverse the consumers' opinions are towards a particular feature. For a feature f of product $s \in \mathbb{S}$:

$$F - Diversity(f,s) = \frac{|Opinion(f,s)|}{\left|\bigcup_{s' \in \mathbb{S}, f' \in F(M_s)} Opinion(f',s')\right|} \quad (11)$$

$$Avg - F - Diversity(s) = \frac{\sum_{f \in F(M_s)} F - Diversity(f,s)}{|F(M_s)|} \quad (12)$$

Where $Opinion(f,s)$ is the set of distinct *opinions* towards the feature f. Recall that the feature extraction algorithm (Algorithm 1) also extracts opinions for each extracted feature. The average feature diversity then quantifies the opinion diversity in features of a particular product. The products with highest diversity include *LG Enlighten*, *Samsung Exhibit 4G*, *LG Cosmos Touch*, *Samsung Dart*, *Kyocera Echo* and *iPhone 5C*. Note that one could observe that these products are either having highly controversial features (i.e. *Kyocera Echo* which offers dual screens with predictive text input and *Samsung Exhibit 4G* which offers dual cameras with surprisingly cheap prices.) or newly launched (i.e. *iPhone 5C*), all of which could incite diverse opinion-related discussion about the product features.