IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2019, accepted October 29, 2019, date of publication December 10, 2019, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2958702

Predicting Supervise Machine Learning Performances for Sentiment Analysis Using Contextual-Based Approaches

AZWA ABDUL AZIZ[1,2] **AND ANDREW STARKEY**[1]
[1]School of Engineering, University of Aberdeen, Aberdeen AB24 3FX, U.K.
[2]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Tembila Campus, Kuala Terengganu 22200, Malaysia
Corresponding author: Azwa Abdul Aziz (azwaaziz@unisza.edu.my)

ABSTRACT Sentiment Analysis (SA) is focused on mining opinion (identification and classification) from unstructured text data such as product reviews or microblogs. It is widely used for brand reviews, political campaigns, marketing analysis or gaining feedback from customers. One of the prominent approaches for SA is using supervised machine learning (SML), an algorithm that uses datasets with defined class labels based on mathematical learning from a training dataset. While the results are promising especially with in-domain sentiments, there is no guarantee the model provides the same performance against real time data due to the diversity of new data. In addition, previous studies suggest the result of SML decrease when applied to cross-domain datasets because new features are appeared in different domains. So far, studies in SA emphasise the improvement of the sentiment result whereas there is little discussion focusing on how to detect the degradation of performance for the proposed model. Therefore, we provide a method known as Contextual Analysis (CA), a mechanism that constructs a relationship between words and sources that is constructed in a tree structure identified as Hierarchical Knowledge Tree (HKT). Then, Tree Similarity Index (TSI) and Tree Differences Index (TDI), a formula generate from tree structure are proposed to find similarity as well as changes between train and actual dataset. The regression analysis of datasets reveals that there is a highly significant positive relationship between TSI and SML accuracies. As a result, the prediction model created indicated estimation error within 2.75 to 3.94 and 2.30 for 3.51 for average absolute differences. Moreover, this method also can cluster sentiment words into positive and negative without having any linguistics resources used and at the same time capturing changes of sentiment words when a new dataset is applied.

INDEX TERMS Text analytics, sentiment analysis, contextual analysis, supervised machine learning.

I. INTRODUCTION

Sentiment analysis (SA) can be described as a computational study to assess people's attitudes, appraisals, and opinions about individuals, issues, entities, topics, events, and products as well as their attributes [1]. It aims to automatically uncover the underlying attitudes that are held towards an entity [2]. It is important to understand users' opinion which is very useful for commercial applications such as product reviews, political campaigns, product feedback, marketing analysis, and public relations. It also has the potential for being used in more critical issues such as security threats like monitoring for discussions related to terrorism. In the new industrial revolution (4IR), unstructured data like social media text has become a central issue for obtaining information as it is freely available through individual user generated content with media platforms such as Facebook or microblogs. However, the challenges to understand text is tough task which is not solid solution proposed neither by researcher nor by industrial parties. In term of Natural Language Processing (NLP), it is usually involves comprehending difference human languages (e.g. English, Chinese, Arabic) which are implicitly related with relevant human aspects such as cultures, countries or religions [3].

In this study, we focus on one of the most common techniques that is used for SA which is Supervised Machine

The associate editor coordinating the review of this manuscript and approving it for publication was Joey Tianyi Zhou.

Learning (SML) [4], [5]. Over recent years, the best results for SA are usually obtained using SML approaches. However, most of the result are from in-domain datasets where training and testing data have similar features (e.g. book reviews) [6]–[8]. The performance of ML drop when they are applied to new datasets from cross-domain sources which contain different features (words) when compared to training data. The results from (Mahalakshmi and Sivasankar [9] experiments show poor results are obtained in Cross-Domain Sentiment Analysis (CDSA) within multiple domains from an Amazon dataset (comprising Book, Kitchen, Electronic, DVD domains) when applying several SML approaches. The study looks at applying data from different domains for training and testing (e.g.: train: book and DVD domain, test: hotel reviews) with the result stated in range of 50 – 75% which is low compare to in-domain dataset. This result proves how the similarity of domains will influence the end results of the model. In our previous work, the experiments undertaken are replicates but changes the features selection method by using Term Frequency-Inverse Document Frequency (TF-IDF) instead of only word [10]. New dataset also has been tested in the study by using 25 000 IMDB records. The result indicates the accuracy in ranges between 66% to 77% depending on how similar the train data against testing data such as DVD and Movie reviews. While overall result show significant improvement when compared to the original experiments, but it still far from what in-domain dataset achieved which generate average 83% accuracy by using the same experiment setting.

While most research in this area focuses on how to improve SA results, to date, there are no studies that look to identify when the proposed SML model breaks down. Evaluating the fitness of the model when applied to real datasets (and in real time) is an important issue since it would clearly be useful to know when the performance of the model begins to deteriorate. However, to date there are no studies that focuses on this issue. The previous studies reported on how the model performance drops when more unknown features are included in the testing dataset. One of the reason because the semantic problems of the sentiments words such as the same words represent difference sentiment (unpredictable movie vs unpredictable car tyre [11]. Therefore, this paper proposes a novel Contextual Analysis (CA) method that looks to predict the performance of SML models. The CA method is an approach that focuses on the relationship of the words and groupings that words with similar contexts create based on the aggregation of their sources (i.e. where different words appear in the same context, or source document). The method is inspired by the Self-Organizing Map (SOM) [12], which is a type of Artificial Neural Network (ANN). Words and sources are embedded together in nodes within a tree-based structure on the relation and intersection of sources. Branches of the analysis are further created using parent node-child objects that allow the contextual analysis of sources to be undertaken. Figure 1 shows an example of a knowledge tree created by CA.

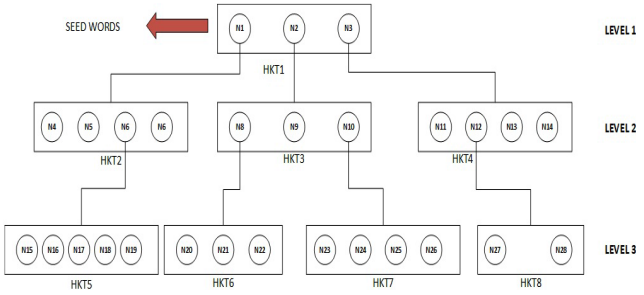


FIGURE 1. CA knowledge tree.

This paper seeks to address the following issues:

i. Proposes a method (CA) that can be used to predict the performance of ML models, and to give an early warning flag which indicates when the performance of ML models are deteriorating with new datasets
ii. Capturing and understanding changes between positive and negative words used by the SA process based on influential nodes in the tree structure

The paper has been divided into five parts. The first part deals with the introduction and aims of the research. The second part begins by laying out the theoretical dimensions of the research and looks at the current trend and results in CDSA. The third part describes the design, synthesis, characterisation, and method of CA followed by part four which gives the results of the experiments. Finally, in part five, the conclusions and discussion of potential improvements are made.

II. RELATED WORKS

As mentioned in the previous section, the two most common approaches in text analytics are using ML and NLP. ML is divided into two: supervised Machine Learning (SML) and unsupervised ML. The paper focus in SML for text analytics for comparison. Generally, SML exploits training data to train a classifier and predict unknown data in testing data. Classifiers are mostly trained using a set of features comprised of n-grams which is a contiguous sequence of n items from a given sample of text or speech.

SA or opinion mining aims at understanding a user's attitude and opinions by investigating, analysing and extracting subjective texts involving users' opinions, preferences and sentiment [13]. It is a sophisticated process that not only involve with the model training, but also a numerous additional procedure such as data processing, transformation and dimensionality reduction [14]. SA research can be categorized into four types of analysis based on Feldman [15]. There are Document Based Sentiment Analysis (DBSA), Sentence Based Sentiment Analysis (SBSA), Aspect Based Sentiment Analysis (ABSA) and Comparative Sentiment Analysis (CSA) as shown in Figure 2.

DBSA is the simplest form of sentiment analysis that assumes the document contains only one main opinion

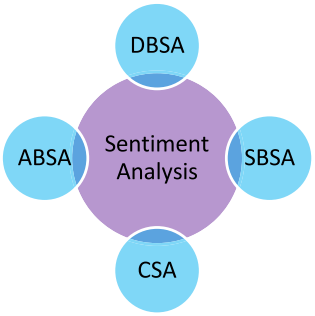


FIGURE 2. Sentiment analysis categories.

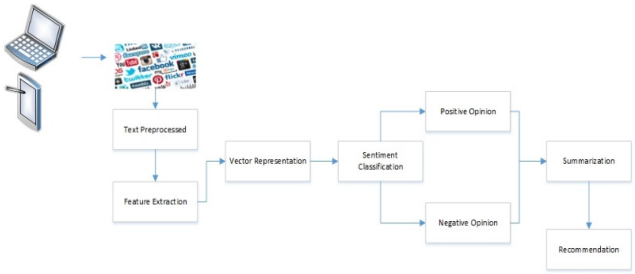


FIGURE 3. Sentiment analysis framework.

expressed by the author of the document. SBSA focuses on sentence-level sentiment for solving multiple opinions contain in particular documents. A study by Chen *et al.* [16], they used divide-and-conquer approach which first classifies sentences into different types, then performs sentiment analysis separately on sentences from each type. In CSA, the goal of the study is to identify sentences that contain comparative words such as *"better", 'more", "less", "most", "least", "outperform"* and that are important to describe the entities. Comparative opinion is considered as one of the main challenges in SA studies. ABSA refers to the contextual analysis of sentiment or which aspect (features, product features, and opinion targets) are expressed [17]. In another definition, Tubishat *et al.* [18] defined aspect or also known as feature level is a fine-grained model that deals with determining opinion intended by people to specific features of a product, service, or any entities. For example, phone reviews may involve specific aspects relating to a phone such as sound, camera, design, and price.

There are two main approaches to the problem of extracting sentiment automatically which are Lexicon-Based Approaches (LBA) and ML [19]. LBA uses dictionaries of words annotated with the word's semantic orientation, or polarity. Dictionaries for lexicon-based approaches can be created manually [19] or automatically, using seed words to expand the list of words [20]. The studies that focus on lexicon may lead to only certain languages can get direct benefit from its. The natural languages that are more ubiquitous online such as English and Chinese emerges as the best target for applying SA verified by the mass amount of SA papers and tools for these languages [21]. In contrast, ML is processed using computational intelligence to acquire knowledge through data and correctly generalise to new settings. It is divided into two major areas; supervised and unsupervised learning. This paper focuses on supervised approaches as the majority of practical machine learning for textual analysis use supervised learning for sentiment analysis studies.

The supervised learning process exploits the labeled examples in the training dataset [31]. Supervised learning creates some form of function that maps the input variables (x) to an output variable (y).

$$y = f(x)$$

The goal is to approximate and generalise the mapping function so that when new input data is presented the model is able to accurately predict the output variables for that data. In general, the features or attributes that are used to build the ML model are pre-defined by the user. Figure 3 shows the general framework for SA using supervised ML according to Zhao *et al.* [6].

The success of the SML model is dependent on the knowledge and information that exists in the training data. A considerable amount of studies has been published on how well ML approaches perform for the in-domain dataset. An early study by Ye et al. 8] in 2009 shows the best results for online travel reviews are between 86% from SVM and 80% from Naïve Bayes Model. The study indicates significant improvement when experiments repeated with more training data have been added. In 2015, a work conducted by Fang and Zhan [22] shows SVM gives 94% accuracy using product reviews datasets. However, the results only rise to 94% from 61% since training data are added from 180 records to 180 million. Vinodhini and Chandrasekaran [23] combine SVM with bagging techniques for ensemble methods that improve overall result for majority and also minority class when compared to standard SVM. However, it requires extensive experiments with benchmark and real application dataset. The success of SVM is repeated by Zhang *et al.* [24], a study that focuses on the semantic relationship between words before proceeding with the classification. The result of their work achieves over 90% accuracy using Chinese text data. Although the result is high, the approach requires word2vec methods to define the relationship between the words, which the author stated that they are having a problem in high-dimensional features vectors of word2vec for SVM. SVM deals with predictive binary classification whereby using a large set of observation with known label (training data), it finds maximum margin function that separates observation into two classes [25]. In word2vec case, the diversity of relationship obtains between each individual word from the algorithm lead to multidimensional spaces of features measurements.

Unfortunately, SML methods do not always guarantee success especially in Cross-Domain Sentiment Analysis (CDSA). The challenges of CDSA are clearly demonstrated by a review paper by Al-Moslmi *et al.* [26]. This paper provides technical reviews on CDSA research contained in 6 prominent academic databases (ACM Digital Library,

Science Direct, IEEE Xplorer, Scopus, Springer Link, Google Scholar) from the 2010-2016 period. The researcher clearly states the failure of CDSA analysis is the lack of annotated data for training purposes. While improved CDSA results can be achieved by combining with LBA such as described in work by Bach *et al.* [27], these results are still below 80%. The approaches known as Canonical Correlation Analysis (CCA) only indicates results between 72% to78% although some results indicate slightly improvement when compared to Blitzer *et al.* [28] and Xia works [29]. Most importantly, these approaches do not give the ability to detect the failure or degradation of the model performance in real time.

It is generally accepted that one of the key drivers for success with SML models is that similar training data is required for the testing dataset. Dorald [30] explained that the first reason for failure in ML is when a new object presented during testing is not similar to training data. This is also known as the generalization problem and occurs when examples in the training data are insufficient or not representative enough of the samples met during testing phases. The results of Fang and Zhan [22] and Ye *et al.* [8] studies support the statement of SML success rate depending on the fitness of training datasets and they show increased performance in the results when more labeled data are used for training. This however leads to a main problem when using a real dataset that does not have any class labels. An early indication of the performance degradation of the ML is important so that positive countermeasures for the problems can be taken such as retraining the model on new labelled data. The most common mistake among ML beginners is the illusion of success when they achieve a great result on training data [7]. Therefore, a mechanism needs to be developed to understand the underlying data relationships that are used for both training and new datasets. A suggestion from Al-Moslmi *et al.* [26] is to focus on developing methods that include distributional similarity (relatedness) in the proposed model and this is a motivation behind the techniques described in this paper.

III. CA APPROACH

A. CONSTRUCT HIERARCHICAL KNOWLEDGE TREE (HKT)

The main idea behind CA for text analytics is to construct a HKT using unlabelled data that helps to understand the subject or the body knowledge of data. There is an opportunity to use CA as an approach to predict the rate of success or failure of the other SML models by comparing the tree structure between training and testing dataset.

The first step of the CA process is to clean the dataset using textual pre-processing techniques. The text is tokenized into smaller pieces or tokens. Then, the process continues to remove stop words; (i.e. 'the', 'a') or any words that are less than 4 characters in length; (i.e. 'and', 'but', 'so'). Next, each word needs to be normalised. The normalisation approach converts text to the same case, removes punctuation, replaces characters, and so on. There are two crucial processes in text normalization: stemming and lemmatization. Stemming is the process of eliminating affixes (suffixes, prefixes, infixes) from a word in order to obtain a word stem. In contrast, lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. In order to handle negation, CA combines negation words (*not*, *never*) with the next word in the sentence (e.g. *not bad* $\rightarrow$ *notbad*, *not good* $\rightarrow$ *notgood*). The second stage of the CA process is to convert all words and sources into a vector. Each word and source is transformed into a number using a lookup table. This allows a faster comparison between words and sources to be made and allows the computational tasks to be undertaken more efficiently, and for the construction of the tree to be completed more quickly.

The third stage creates the HKT which is the most critical part of the process. This is created by using information relating to both individual sources and individual words following the above pre-processing. For a given data D containing a set of sources denoted by $D = \{S_{1,}S_{2,}\ldots.S_{n.}\}$, where n is the number of sources each source (S_i) has a set of unique words which in a sequence of m words is denoted by $S_i = \{W_{1,}\ W_{2,}\ldots.W_n\}$. Words that are repeated in the source are included only once. The first process finds the words that appear in the most sources in the dataset to create the first node (n_1) in the tree using formula below known as the word frequency (*wf*):

$$wf\ (w, d) = c(w, d)$$

where *c(w,d)* stand for the frequency count of unique word w in sources (s) of dataset D. The unique means if the words appeared more than one time in particular sources (s), it still represented as a one value. The process begins by creating the first node (N_1) that explains in the formula below:

Step 1:

$$maxCountWord \rightarrow n_1$$
$$\max CountWord\ (W_{si}) = \sum_{s=1}^{D}\sum_{i=1}^{J} W_{si}(i)$$

This is followed by comparing the sources of the second highest word (W_2). To continue the analysis for other words, a threshold (λ) is set on the minimum count of sources for the words to be used, which is set at 0.5 (for these experiments) of the total the sample highest ranking word (W_1). This is to ensure each level of the tree contains words of a similar strength (based on counted frequency). The words will become seed words that will be attached to the nodes in first level Tree (HKT_1). To finding potential words (δ) to be compared with n_1:

Step 2:

$$\forall WP = countWP > \lambda.\max(W_{si}), \quad \lambda \geq 0.5$$
$$\delta \rightarrow f(s, WP) \qquad (ii)$$
$$= \sum_{s=1}^{D}(ii)\sum_{i=1}^{J} WP_{si}, \{wp_1, wp_2, \ldots, wp_n\} \in WP \qquad (iii)$$

As a result, generation of a ranking position for all word's potential (WP) in $D = \{wp_1, wp_2, \ldots, wp_n\}$ by counting sources (s) attach to potential words (wp) are produce. To rank wp according to position i as a descendant:

Step 3:

$$\delta = \{wp_i, wp_{i+1}, wp_{i+2}, \ldots, wp_{i+n}\}, \quad i = 1 \qquad (iv)$$

Each node contains a set of sources and words attached which represented by:

$$N = \{W, S\}$$

where $\{w_1, w_2, \ldots, w_n\} \in W$ and $\{s_1, s_2, \ldots, s_n\} \in S$. All potential words will be allocated into nodes in first level cluster known as Hierarchical Knowledge Tree (HKT_1). Other nodes created by comparing dataset for each potential word (WP).

Let say the first node (n_1) having set of sources in dataset (d_{n_1}) with $s_1, s_2, \ldots, s_n \in d_{n_1}$. Word potential ($WP_i$) also having a set of datasets (d_i). The potential dataset that be allocated in HKT_1 is show below:

$$HKT_1 = \{W_{n_1} : \forall d_{n_1}, WP_i : \forall d_i, WP_{i+1} : \forall d_{i+1}, \ldots, \forall WP_{i+n} : \forall d_{i+n}\}$$

where $d_{n_1}, d_i, d_{i+1}, d_{i+2} \ldots, d_{i+n} \in D$ and $D = \{s_1, s_2, \ldots, s_n\}$. The first iteration for comparing dataset to first existing node $n_1 = \{W_{n_1}, S_{n_1}$ is depicted with the following equation:

Step 4:

$$\alpha = f(i) \rightarrow d_{n_1} \cap d_i, \quad where\ \forall x, \left(x \in d_{n_1}\right) \wedge (x \in d_i) => x \in (d_{n_1} \cap d_i)$$

Condition 1:

$$\alpha \geq \lambda . \sum_{i=1}^{N} s_i, \quad \lambda \geq 0.5 \quad \text{and } \{s_1, s_2, \ldots, s_n \in d_{n_1}\}$$

where λ is a threshold value to find similarity between dataset. Therefore, new words $\left(W'_{n_1}\right)$ and sources (S'_{n_1}) attach to n_1.

$$W'_{n_1} = \sum_{k=1}^{N} W_{n_1 k} + \sum_{k=1}^{N} WP_{ik}$$
$$S'_{n_1} = \sum_{k=1}^{N} d_{n_1 k} + \sum_{k=1}^{N} d_{ik}$$

Condition 2:

$$\alpha \leq \lambda . \sum_{i=1}^{N} s_i, \lambda \geq 0.5 \quad \text{and } \{s_1, s_2, \ldots, s_n \in d_{n_1}\}$$

New node (n') is created. The total nodes in giving tree (T) can be describe as:

$$T = \sum_{i=1}^{M} n_i$$

which $N' = \{W', S'\}$ and $\{n_2, n_3, \ldots, n_n\} \in N'$.

In general, the intersection of sources from these two words are above a predefined threshold (in this case study 0.5 has been used) then the word is added to the node. Any new sources are also added to the list of sources for this node. If the intersection of sources does not meet the threshold then a new node is created which contains this word and its associated sources.

This process continues with the next potential word (WP_i), by comparing the sources of this word against all existing nodes and the process is repeated. The next equation shows the formula to compare intersection between existing node and the next available WP_i dataset.

Step 5:

$$f(n_i) = \sum_{i=1}^{M} d_{n_i} \cap \sum_{i=1}^{M} d_i, x \in d_{n_i} \quad and\ x \in d_i \qquad (v)$$

After the process is exhausted, CA will verify that each source is mapped to at least one particular node. If not, then a new node is created to gather the remaining source

The next step in the process is to create branches for the remaining words using the nodes created in the first level of the HKT. To create a branch of analysis under a defined node, the number of words that remain in the data that map to this node once the words used by the node are removed is checked against a predefined threshold. This threshold checks the number of sources that have words remaining to be analysed. If sufficient words and sources remain, then the process begins as above but now with this reduced dataset. The formula below explained how the child node is created:

Let say we have list of parent nodes (P), were $P \in HKT_1$. To create branches, a remaining word exist in P nodes which is not appeared in HKT_1 will be populated to a child node.

$$Z \rightarrow P_i + P'_i, \quad where\ \{w_1, w_2, \ldots, w_n\} \in Z$$

To find remaining word (P'_i)

$$P'_i \rightarrow SP_i$$

$\forall P'_i$, repeat the step (i-v) process until $P'_i = \emptyset$.

For each threshold $\lambda > 0.5$ iteration for the child nodes the level/deep (μ) of tree (T) and increase to 1 that can be depicted in function below:

$$f(\mu) = \mu + 1$$

Finally, the complete structure of tree using above formula is obtained as graphically shown in Fig. 1. The relation of the tree (T) can be formalized by following mathematical equation.

The relation between HKTs and nodes

$$T = \{N\}, \quad where\ N \subseteq HKT, \{n_1, n_2, \ldots, n_m\} \in N.$$
$$HKT = \{hkt_1, hkt_2, \ldots, hkt_m\} \in T$$

For each node, it is containing words (W) and sources (S).

$$N = \{W, S\}, \{w_1, w_2, \ldots, w_m\} \in W.\{, \{s_1, s_2, \ldots, s_m\} \in S$$

The size of tree can (x) be determined by accumulating the sum of HKT or total nodes.

$$f(x) = \sum_{i=1}^{M} x_i, x = \{hkt_1, hkt_2, .., hkt_m\} | x = \{n_1, n_2, \ldots, n_m\}$$

Based on the equation, therefore, total nodes (β) also can be derived by using this equation:

$$\beta = \sum_{h=1}^{K} \sum_{i=1}^{J} N_{hi}$$

To describe relationship between child HKT (cH) and parent nodes (pN)

$$T = (pN, cH), \quad where \{n_1, n_2, \ldots, n_m\} \in pN, \{hkt_2, hkt_3, .., hkt_m\} \in cH \text{ and } hkt_1 \notin cH$$

The sample relation of tree can be described in formula below:

$$T = \{(n_1, hkt_2), (n_2, hkt_3), (n_3, hkt_3), \ldots, (n_{max}, hkt_{max})\}$$

As a result of this process, three types of relationship are obtained from the hierarchy knowledge tree that can be explored; words in the same node (identical-relation) $(w_i.k_i \in n_i)$, words in the same level but in different nodes (genealogical-relation) $(w_i, k_i \in \mu_i.w_i, k_i \notin n_i)$, words that are related by parent node-child relation (inheritance-relation) $cH_i = \{w_i, k_i\}, T = (N_i, cH_i)$.

B. TREE SIMILARITY INDEX (TSI) AND TREE DIFFERENCES INDEX (TDI)

To find interesting nodes for training classification, the HKT having ability to burst 'influential nodes' such illustrate in Figure 4 based aggregation of sources attach to the nodes. For instances, yellow nodes in the diagrams are triggered if having more than 90% class label data such as sentiments of fake news. These nodes contain crucial words that can help users to understand the data likes important sentiment words *(poor, fail, bad, great)* that may be used to improve the classification tasks.

To measure similarities between datasets, we compare the words in the first level of the tree that is created for both training and testing data. The TSI calculation is inspired from Association Rules (ARs) approaches and term-frequency (TF) calculation. In the case of the term frequency $TF(t, d)$, the simplest choice is to use the *raw count* of a term in a document (t), i.e. the number of times that term t occurs in document d. If we denote the raw count by $f_{t,d}$, then the simplest tf scheme is $TF(t, d) = f_{t,d}$. However, for CA, the counts are based on number of words in each node when compared to the overall numbers of sources in the tree.

ARs learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in

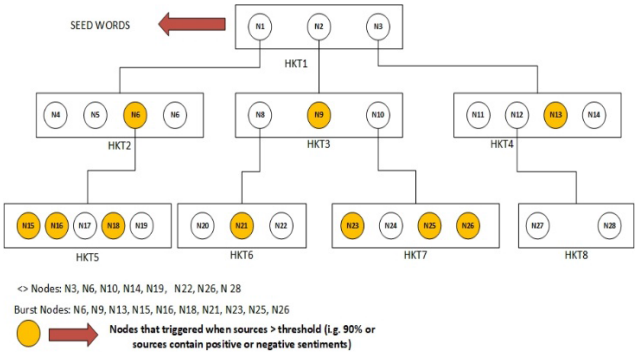


FIGURE 4. Burst nodes in tree.

databases using some measures of interestingness. AR calculates support and confidence attributes for associations between two datasets named A and B. The *support (s)* for an association rule B → A is the percentage of transactions in the database that contain B U A.

$$support\ (B \rightarrow A) = P(B\text{UA})$$

However, for CA changes are made to the calculation for support. The calculation looks at how many words appear in (B) that also appear in (A). The idea is not to find the association between A and B, but to identify changes that occur (comparison) in B (actual set) compared to A (training set). The TSI(σ) idea offers an approach to find similarity between two datasets which is an important feature for cross-domain sentiment analysis.

$$TSI = T(A \cap B)/T(B) \quad \text{where } x \in A \cap B \text{ if } x \in A \text{ AND } x \in B$$

The *confidence* or *strength* (α) for *similarity* can be calculated by the position of nodes in the tree. The position of nodes plays an important role in tree structures as the leftmost words and lead for the nodes have more frequency (words count) compared to the right nodes. The first step finding diffrences of the tree based on populated words in nodes

$$\sigma = T_{traindata} \cap T_{actualdata}$$

To find similar words (x) exist in actual dataset for there first level where weight $(\mu = 1)$ can be explored by applying the next equation:

$$\sigma = T_a \cap T_b, \quad \forall x \in T_a.x \in T_b$$

$$\sigma = \mu \left(\frac{\sum_{i=1}^{n} W_{a_i}}{\sum_{i=1}^{n} W_{b_i}} \right), \quad x_{a_i} \in T_{a_i}.x_{b_i} \in T_{b_i} \text{ and } x_{a_i} = x_{b_i}$$

The weight is based on the deep/level of the tree, which mean the actual and train tree have a strong relation parent-child if the level of tree increased. To find the overall TSI for tree is based on following equation:

$$\forall h_i = (n_i, h_i)\sigma = \sum_{v=1}^{K} \sum_{h=1}^{J} \mu_{vh} \left(\frac{\sum_{i=1}^{n} x_{a_i}}{\sum_{i=1}^{n} x_{b_i}} \right)$$

In contrast, TDI (φ), used to detect changes that occur between train and actual data. It is important features in order to capturing changes when implement in real world that lead the current model need to be revamp if the index is to high.

$$TDI = T_{traindata} \backslash T_{actualdata}$$

To find new words (x) exist in actual dataset for there first level where weight ($\mu = 1$):

$$\varphi = T_a T_b, \quad \forall x \in T_a . x \notin T_b$$

To find the overall $TDI(\varphi)$, the equation can be derived from TSI equation.

$$\varphi = \sum_{v=1}^{K} \sum_{h=1}^{J} \mu_{vh} \left(1 - \frac{(\sum_{i=1}^{n} x_{a_i})}{\sum_{i=1}^{n} x_{b_i}}\right), x_{a_i} \in T_{a_i} . x_{b_i} \in T_{b_i} \; and \; x_{a_i} = x_{b_i}$$

The paper will therefore explore whether a correlation exists between *TiS* and *ML accuracy* metrics that allow CA methods to detect the degradation in performance for the proposed ML model. In experiment, the weight (μ) are set to 1, for finding relationship only for the first level of tree. Thus, a number of experiments are designed in order to obtain a decrease in ML performance in order to test the hypotheses described below:

a. The performance of the ML model will decrease gradually when the testing dataset is combined with other domains ($\mathbf{H_1}$).
   The ability of the ML model depends on the similarity of knowledge between training and testing datasets. If the new dataset has different features than were identified in the training data, then just as in cross-domain problems, the performance of the model will decrease.
b. CA can detect the degradation performance of the model by capturing changes based on TSI model calculation ($\mathbf{H_2}$).
   CA can help to find the difference between training and testing dataset by comparing the structure of HKT for both datasets.
c. There is a correlation between a decrease or increase in ML performances with the TSI CA calculation ($\mathbf{H_3}$).
   The result should match the changes in ML accuracy. If the ML model shows a decrease in accuracy, then the TSI calculation should also follow this trend.
d. CA can capture and identify new words that appear in the HKT created from the testing dataset ($\mathbf{H_4}$).
   Comparing the tree structures will allow identification of new words that have been found in the testing dataset.

The experiment design will be based on 4 different domains contained in Amazon datasets introduced by Ghaddar and Naoum-Sawaya [25] Those domains are:

Using Electronics as bench data for the training dataset, data from other domains will gradually be introduced into the testing dataset. This is to prove ($\mathbf{H_1}$) that the performance of

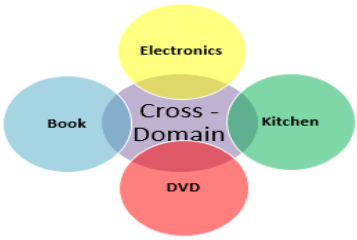


FIGURE 5. HKT cross-domain dataset.

ML models are based on knowledge of feature selection in training dataset, and to highlight the problem of cross domain analysis. The various datasets thus described can be seen in Table 1 below:

TABLE 1. Dataset setup.

Test number	Training data	Testing data
1	100% Electronic data	100% Electronic data
2	100% Electronic data	80% Electronic data, 20% kitchen
3	100% Electronic data	60% Electronic data, 40% kitchen
4	100% Electronic data	40% Electronic data, 30% kitchen, 30% DVD
5	100% Electronic data	20% Electronic data, 30% kitchen, 30% DVD, 20% book

Table 1 display five dataset are created that labels as Test 1 to Test 5. The in-domain knowledge for testing dataset will gradually reduce by adding more cross domain data. The idea to test whether the performances of SML are decreased when different feature set appeared in testing data.

IV. RESULTS

A. CORRELATION ANALYSIS (ACCURACY VS TSI)

Experiments have been conducted to measure the accuracy of ML (whether a decrease or increase) using (Random Forest Classifier (RFC) and Multinomial Naive Bayes (MNB)). There methods obtained the highest accuracy from Cross-Domain replication experiments. The results are shown in Table 2 based on average accuracy calculation (Recall).

TABLE 2. Result degradation of dataset.

Method	Test 1	Test 2	Test 3	Test 4	Test 5
RFC	81%	79%	78%	74%	72%
MNB	81%	80%	78%	75%	72%

The result confirms the $\mathbf{H_1}$ hypothesis which states the performance of ML approaches have a steady decrease when more unknown domain knowledge appears in the testing dataset. Figure 6 illustrates the drop-in performance of each test result when new domain knowledge is applied in the testing dataset.

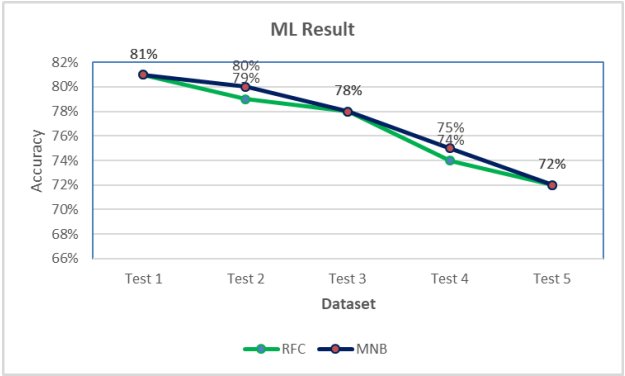


FIGURE 6. MNB and RFC performance decrease.

TABLE 3. Calculate TSI and TDI.

Test	$TSI(\sigma)$	$TDI(\varphi)$	σ percent	Confidence
Test 1	0.85	0.15	85%	100%
Test 2	0.75	0.25	75%	100%
Test 3	0.93	0.07	93%	50%
Test 4	0.67	0.33	67%	100%
Test 5	0.50	0.50	50%	100%

This result highlights an important issue when ML models are implemented in real time against real world data; with no label data provided for the real-world data, how can the performance of the deployed ML model be assessed? This question gives the main motivation to study the fitness of a given ML model when used in real time against real world data.

To begin this process, the TSI between training (A) and testing (B) data for all five datasets is calculated. The result is shown in Table 3.

The results show a clear trend of decreasing similarity across the test datasets except for Test 3. However, this result also shows a 50% confidence level. The overall result shown in the table may support the $\mathbf{H_2}$ and $\mathbf{H_3}$ hypotheses, and these are illustrated in Figure 7 below. To confirm whether a significant relationship between the similarity measurement and the ML accuracy is statistically proven, correlation analysis needs to be undertaken.

To determine the correlation, the Pearson correlation coefficient (PCC) is used, a statistic calculation that measures the linear correlation between two variables X and Y. The formula is depicted below:

$$r = \frac{N\sum xy - (\sum x)\sum(y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

where

$$\begin{aligned} N &= \text{number of pairs of scores} \\ \sum xy &= \text{sum of the products of paired scores} \\ \sum x &= \text{sum of x scores} \\ \sum y &= \text{sum of y scores} \\ \sum x^2 &= \text{sum of squared x scores} \\ \sum y^2 &= \text{sum of squared y scores} \end{aligned}$$

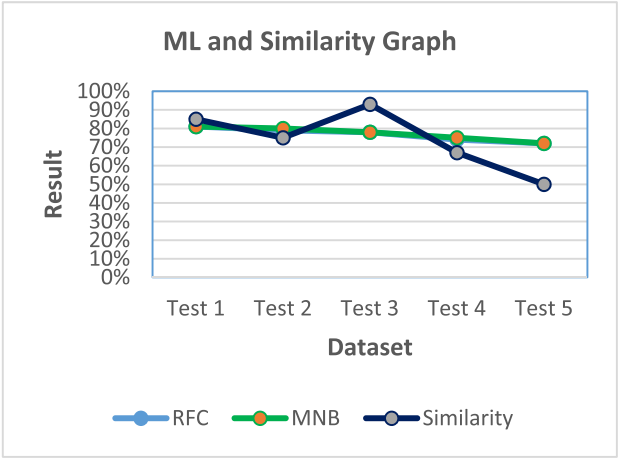


FIGURE 7. ML accuracy vs TSI result (similarity).

The correlation coefficient (R) ranges from -1 to 1, with a value of 1 indicating that a linear equation describes the relationship between X and Y perfectly, with all data points lying on a line for which Y increases as X increases. A value of -1 for R indicates that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables. In this case, the result shows a value of $R = 0.82$, which indicates that there is a high positive relationship between the decrease TSI results and ML accuracy result.

The alpha is equal to 0.05, which mean that if the value of P is more than 0.05, the result is significant and vice versa. This test shows that $P = 0.08$, which indicates that although it has a strong positive relation, it is not considered statistically significant.

The result in Test 3 may influence the significance result and so the test was run again with data that only has 100% confidence which excludes the Test 3 result. The R result shows improvement and strong positive correlation with values of $R = 0.96$ and $P = 0.04$. This indicates a statistically significant positive relationship since $P<0.05$.

The result proves that a positive correlation exists between the *Similarity* measure and ML accuracy and proves that CA can be used to predict the performance of ML models against real datasets.

B. ANALYSIS OF SML RANDOM DATASET FOR PREDICTION

To further support this conclusion, and to demonstrate the effectiveness of the CA approach, further experiments were conducted by using a random dataset from 13000 (2000 samples for each experiment) sentiment records from five domains; book, DVD, kitchen, electronics, movie as testing dataset and 2000 Eletronics data to be training dataset. A total of 26 experiments are conducted to determine the consistency of result obtained using the CA method when three types of ML approaches were implemented (MNB, RFC, and SVM). Each experiment contained 3000 records for

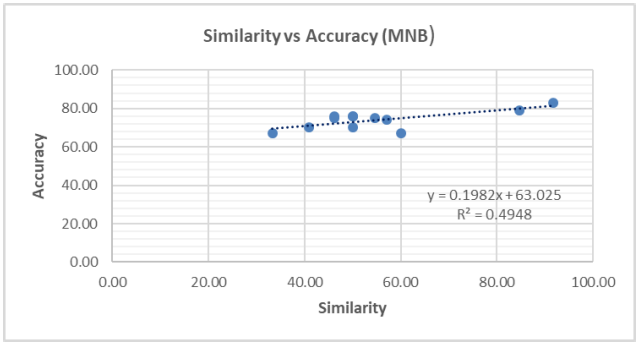


FIGURE 8. MNB linear regression.

each testing dataset. The results from 12 experiments is then used in order to teach a regression model the characteristics of the ML performance against the CA similarity measure in order to predict the remaining datasets.

Figure 8 presents the regression result for MNB learning dataset. The linear regression model created has $R^2 = 0.4948$ and linear equation of $y = 0.1982x + 63.025$.

The next step is to predict the remaining 16 experiments using the equation obtained from the regression model. In order to measure of the accuracy of predictions, the standard estimation calculation using below definition:

$$\sigma_{\text{est}} = \sqrt{\frac{\sum (Y - Y')^2}{N}}$$

where σ_{est} is the standard error of the estimate, Y is an actual score, Y' is a predicted score, and N is the number of pairs of scores. The numerator is the sum of squared differences between the actual scores and the predicted scores. The results derived from the prediction of SML accuracy can be compared as shown in Table 4.

where Y' = predict value, Y = actual value.

The result shows that the average of the absolute differences is 2.75 with estimation error of 3.16. This is an encouraging result, and furthermore the results show that the model can predict a variety of results whether it is high (e.g. Test 23), in the middle (e.g. Test 11) or at the bottom (e.g. Test 18). Figure 9 illustrates a comparison between the actual and predicted values for the MNB model.

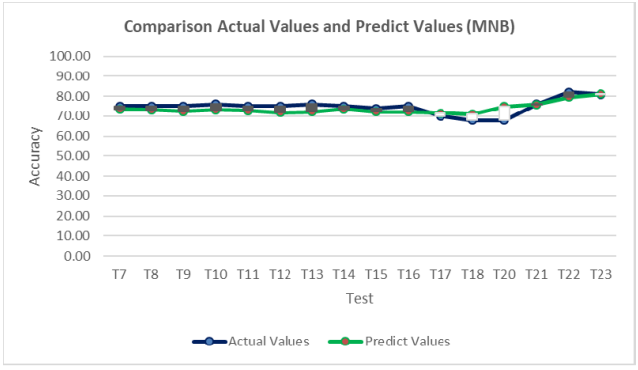


FIGURE 9. Actual values vs. predict value (MNB).

TABLE 4. Result for random dataset.

TEST	TSI (%)	(Y')	(Y)	(Y-Y')	$(Y-Y')^2$
Test 7	50.00	72.94	75.00	2.07	4.26
Test 8	48.00	72.54	75.00	2.46	6.06
Test 9	44.44	71.83	75.00	3.17	10.03
Test 10	48.00	72.54	76.00	3.46	11.98
Test 11	46.15	72.17	75.00	2.83	8.00
Test 12	41.38	71.23	75.00	3.77	14.24
Test 13	42.86	71.52	76.00	4.48	20.07
Test 14	50.00	72.94	75.00	2.07	4.26
Test 15	42.86	71.52	74.00	2.48	6.15
Test 16	46.15	72.17	75.00	2.83	8.00
Test 17	43.48	71.64	70.00	-1.64	2.70
Test 18	40.74	71.10	68.00	-3.10	9.61
Test 20	60.00	74.92	68.00	-6.92	47.84
Test 21	65.00	75.91	76.00	0.09	0.01
Test 22	83.33	79.54	82.00	2.46	6.05
Test 23	91.67	81.19	81.00	-0.19	0.04
				Total	159.30
				N	16
				$\frac{\sum(Y-Y')^2}{N}$	9.956
				Estimation Error	3.16

where Y'=predict value, Y=actual value.

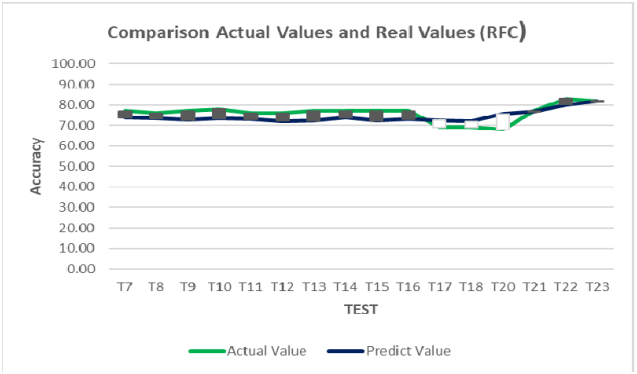


FIGURE 10. Actual values vs. predict value (RFC).

The next experiment replicates the process using RFC and SVM approaches. The predictions against the actual results for the RFC model are shown in Figure 10. The result consistent figure with the result for MNB (Figure 9). The average absolute differences for these results are 3.51 and estimation error is 3.881.

The results for the SVM model are presented in Figure 11. The SVM model represented model with the widest range of results having predictive accuracy from 57% to 75%.

The estimation prediction error for the SVM comparison of results is 3.94 with an average absolute difference of 2.30 which represents the lowest results when compared to MNB and RFC. The results are important since they have shown that the CA approach has the ability to undertake correct predictions over wide range model accuracies.

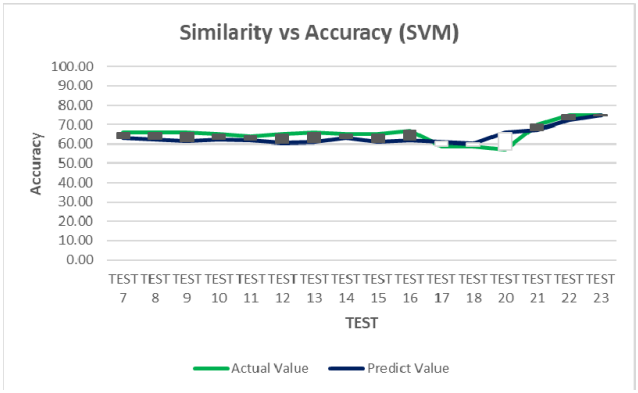


FIGURE 11. Actual values vs. predict value (SVM).

Overall, the experiment shows the consistency of CA regression model across three prominent ML approaches. The estimation error is between 2.75 to 3.94 and 2.30 for 3.51 for average absolute differences. The small margin indicates the consistency of the result.

C. ANALYSIS OF CHANGES OF IMPORTANT WORDS

The CA method has been shown to be able to predict the performance of various ML methods for unlabelled data, and it is able to do this by capturing changes in the words that new datasets introduce. The words that are considered important by the CA method are based on influential nodes which are calculated according to the negative or positive samples that map to these nodes.

To assess these important words two analysis techniques for the calculation of influential nodes are introduced. The first is based on percentage accuracy of sources that attach to the nodes, such that nodes that have more than 90% of sources that are labelled negative or positive are considered influential. The second is inspiring from the *TF* calculation and calculates the total number of sources (positive or negative) that map to the influential nodes divided by the total number of samples in the dataset. Table 5 shows how important words for negative sentiments change as the volume of the new dataset increases for each test and is followed by similar analysis for positive words in Table 6.

It is apparent from these tables that the CA method structures the words from the dataset into positive and negative sentiments without using linguistics resources such as WordNet. More importantly, the words include aspect words (e.g. size, tune, act, battery, phone) and description of aspect (e.g. larger, flexible, bright, heavy) which is represented by influential nodes in HKT. Exploring the HKT tree may reveal underlying information allowing aspect detection and classification based on these tree contexts. However, this study is beyond the scope of this paper which focuses on assessing the $\mathbf{H_4}$ hypothesis and whether the CA is able to successfully predict the changing performance in ML models through the automatic identification of the changing words of the unlabelled data.

TABLE 5. Words from influential negative nodes.

Test	Percentages Accuracy Method	TF Method	Domain
Training	refund, poor, not even, couple, fix, not buy, origin, unfortunate, die, error, worst, ridiculous, unacceptable, not worth, unsatisfactory	not buy, worst, terrible, fail, fix, similar, half, warranty, documents, post, stop, item, map	*Electronics*
Test 1	not turn, damm, monster, constant, useless, router, disconnect, not impress, bullet, RIP, costly, not recognized, failure, horrible, crash, unable, Ethernet, lie	refund, router, wrong, stop, upgrade, old	*Electronics*
Test 2	tough, reliable, suck, firmware, key, broken, adaptor, wife	radio, spent, terrible, wrong, stuck, not buy, policy, repair, trouble, machine, useless, not even, expenses, mistake, signal, failure, malfunction, noise, operate, unfortunate, low, suck, horrible, waste, worst, suffer, refurbish, services, broke, damage	*Electronics+ Kitchen*
Test 3	Refund, cannot, not buy, fail, machine, worst, not recommend, shut, stuck, terrible, not buy, broke, response, paid, access, cooker	battery, hour, defect, phone, pay, refund, fail, services, instruction, mistake, surprise, signal, noise, malfunction, manufacture, broken, suck, error, wrong, damage, poor, junk, defect	*Electronics+ Kitchen*
Test 4	sale, laugh, plot, sequence, reality, waste, situation, lead, simple, actress, horrible, rang	act, actor, person, talk, express, chance, portray, present	*Electronics+ Kitchen +DVD*
Test 5	biggest, sexual, sick, similar, condemn, extreme, shame, cliché, unfortunate, Hollywood, pass, era, dumb, ridiculous, embarrassed, sadly, pathetic, worried	type, complete, handsome, impress, content, complex, worried, not think, brand, genuine, role, waste	*Electronics+ Kitchen +DVD+ Book*

As can be seen from the above examples, a conclusion can be derived that CA is capable of capturing changes in important words between the ML models. The important words

TABLE 6. Words from influential positive nodes.

Test	Percentage Accuracy Method	TF Method	Domain
Training	highlight, size, bass, bright, movie, perfect, excellent, happy, simple, pro, clip, satisfy, subwoofer	excellent, impress, clean, signal, strong, tune, plain, flexible, family	*Electronics*
Test 1	microphone, thumbwheel, radar, desk, access, solid	strong, prove, perfect, optic, loud, extreme, thin, cordless	*Electronics*
Test 2	clean, pair, perform, stick, dollar, perfect, video, absolutely, real	size, holder, ton, aspect, fit, video, pair, heavy, desk, window, video, center	*Electronics+ Kitchen*
Test 3	cheaper, remote, family, free, perform, battery, bake, beauty, chicken, value, remove, protect, meat, vegetable, roast, taste	hand, kitchen, cook, pan, reason, expenses, size, perfect, loud, cook, amplifier, decent, larger, charge	*Electronics+ Kitchen*
Test 4	capacity, vacate, energy, perfectly, excellent, match, sweet, Italian, sun, comfort, perfect, kitchen	learn, fish, ton, fresh, meatloaf, consequence, toaster, dough	*Electronics+ Kitchen +DVD*
Test 5	knife, cookie, baked, sugar, flour, beat, genius, heavy, perfect	dive, knife, perfect, Amazon, police, marriage, ensemble, really, chef	*Electronics+ Kitchen +DVD+ Book*

for train and test 1 contain familiar words for electronics equipment. Adding 20% Kitchen dataset into Test 2 does not make any major changes for important words.

However, in Test 3 which has 40% Kitchen dataset, there are obvious changes in important words. New words appear such as *kitchen, cook, pan, taste, meat, roast* which mostly refer to the words in Kitchen dataset review. In test 4, Kitchen and DVD make up to a total of 60% of the test dataset. Now, we can see more significant words that are combined from the three domain areas (Kitchen, Electronic and DVD). Some example of new words that may come from DVD reviews are *laugh*, *plot*, *act*, *actor* and *portray*.

In test 5, the important words change again when only 20% Electronic data remain. The unique words in Electronics such as a *router, machine, malfunction, noise* and *signal* no longer appear in either *TF* or the percentage method. The results of this experiment show that important words change as the domain changes which highlights the importance of being able to capture these changes in a real-world model to ensure the fitness of the proposed ML model.

V. CONCLUSION AND FUTURE WORK

This paper has described why it is important to be able to measure the performance of SML models against real world datasets in real time. The aim of the paper was to provide a technique that can be used to discover the predictive capability and the abnormalities of SML models. In this study, a novel approach known as CA is proposed to find the relationship between words and sources which can provide a mechanism to predict SML model performance.

This study has shown that there is a significant positive correlation between our proposed similarity approaches formula (TSI) against the accuracy of SML models (MNB, RFC) based on a number of experiments. Multiple regression analysis revealed that there is a statistically significant positive relationship between TSI and the SML results, with $R = 0.96$ and $P < .05$. The evidence from these experiments lead to further 26 experiments being undertaken using a random dataset from 5 domains; Book, DVD, Kitchen, Electronics and Movie, taken from Amazon and IMDB review data. The result derived from CA experiment was found to have estimation error between 2.75 to 3.94 and between 2.30 for 3.51 for average absolute differences across three different ML models. These findings are crucial in proving the ability of CA to predict the performances of various ML model over a range of performance values.

The results presented in Table 5 and 6 shows that CA can identify the words that are changing from one dataset to another, and which cause the lower predictive capability in the ML models. It should be noted that the words are automatically clustered into negative and positive words without any linguistics resources being used, and influential nodes are calculated according to *TF* or *percentage method* calculations. However, future research should consider improving prediction result while performing real time analysis for individual changes of dataset.

In conclusion, these findings show how it is possible to measure the performances of SML models in real time data. Further work needs to be done to establish whether relationship in CA can be used to further understand the structured knowledge of the data.

REFERENCES

[1] J. Kaur, S. S. Sehra, and S. K. Sehra, "A systematic literature review of sentiment analysis," *Int. J. Comput. Sci. Eng.*, vol. 5, no. 4, pp. 22–28, 2017.

[2] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic, "A survey of multimodal sentiment analysis," *Image Vis. Comput.*, vol. 65, pp. 3–14, Sep. 2017, doi: 10.1016/j.imavis.2017.08.003.

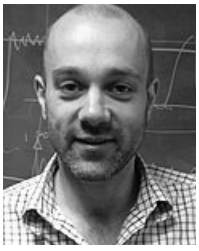
[3] D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez, "Supervised sentiment analysis in multilingual environments," *Inf. Process. Manage.*, vol. 53, no. 3, pp. 595–607, May 2017, doi: 10.1016/j.ipm.2017.01.004.

[4] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017, doi: 10.1016/j.eswa.2017.02.002.

[5] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, Jeju Island, South Korea, 2012, pp. 90–94. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390665.2390688

[6] L. Zhao, M. Huang, H. Chen, J. Cheng, and X. Zhu, "Clustering aspect-related phrases by leveraging sentiment distribution consistency," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1614–1623. [Online]. Available: http://emnlp2014.org/papers/pdf/EMNLP2014169.pdf
[7] X. Zhang and X. Zheng, "Comparison of text sentiment analysis based on machine learning," in *Proc. 15th Int. Symp. Parallel Distrib. Comput. (ISPDC)*, 2016, pp. 230–233, doi: 10.1109/ispdc.2016.39.
[8] Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6527–6535, Apr. 2009, doi: 10.1016/j.eswa.2008.07.035.
[9] S. Mahalakshmi and E. Sivasankar, "Cross domain sentiment analysis using different machine learning techniques," in *Proc. Adv. Intell. Syst. Comput. 5th Int. Conf. Fuzzy Neuro Comput. (FANCCO)*, 2015, pp. 77–87.
[10] A. A. Aziz, A. Starkey, and M. C. Bannerman, "Evaluating cross domain sentiment analysis using supervised machine learning techniques," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 689–696, doi: 10.1109/intellisys.2017.8324369.
[11] B. Liu, *Morgan & Claypool Publishers*. Morgan & Claypool Publishers, 2012.
[12] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982, doi: 10.1007/bf00337288.
[13] Z. Nanli, Z. Ping, L. Weiguo, and C. Meng, "Sentiment analysis: A literature review," in *Proc. Int. Symp. Manage. Technol. (ISMOT)*, Nov. 2012, pp. 572–576.
[14] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Syst. Appl.*, vol. 106, pp. 36–54, Sep. 2018, doi: 10.1016/j.eswa.2018.03.058.
[15] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, p. 82, Apr. 2013, doi: 10.1145/2436256.2436274.
[16] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017, doi: 10.1016/j.eswa.2016.10.065.
[17] A. Bagheri, M. Saraee, and F. de Jong, "An unsupervised aspect detection model for sentiment analysis of reviews," in *Proc. Natural Lang. Process. Inf. Syst.*, 2013, pp. 140–151.
[18] M. Tubishat, N. Idris, and M. A. Abushariah, "Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges," *Inf. Process. Manage.*, vol. 54, no. 4, pp. 545–563, Jul. 2018, doi: 10.1016/j.ipm.2018.03.008.
[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011, doi: 10.1162/coli_a_00049.
[20] V. Hatzivassiloglou and K. R. Mckeown, "Predicting the semantic orientation of adjectives," in *Proc. 8th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 1997, pp. 174–181, doi: 10.3115/979617.979640.
[21] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, and M. N. Al-Kabi, "A comprehensive survey of arabic sentiment analysis," *Inf. Process. Manage.*, vol. 56, no. 2, pp. 320–342, Mar. 2019, doi: 10.1016/j.ipm.2018.07.006.
[22] X. Fang and J. Zhan, "Sentiment analysis using product review data," *J. Big Data*, vol. 2, no. 1, p. 5, 2015, doi: 10.1186/s40537-015-0015-2.
[23] G. Vinodhini and R. Chandrasekaran, "A sampling based sentiment mining approach for e-commerce applications," *Inf. Process. Manage.*, vol. 53, no. 1, pp. 223–236, Jan. 2017, doi: 10.1016/j.ipm.2016.08.003.
[24] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, Mar. 2015, doi: 10.1016/j.eswa.2014.09.011.
[25] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018, doi: 10.1016/j.ejor.2017.08.040.
[26] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, vol. 5, pp. 16173–16192, 2017, doi: 10.1109/access.2017.2690342.
[27] N. X. Bach, V. T. Hai, and T. M. Phuong, "Cross-domain sentiment classification with word embeddings and canonical correlation analysis," in *Proc. 7th Symp. Inf. Commun. Technol. (SoICT)*, 2016, pp. 159–166.
[28] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, Prague, Czech Republic, 2007, pp. 440–447. [Online]. Available: http://aclweb.org/anthology/P07-1056
[29] R. Xia, C. Zong, X. Hu, and E. Cambria, "Feature ensemble plus sample selection: Domain adaptation for sentiment classification," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 10–18, May 2013, doi: 10.1109/mis.2013.27.
[30] L. Dorard. (2014). *When Machine Learning Fails*. [Online]. Available: http://www.louisdorard.com/blog/when-machine-learning-fails
[31] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Amsterdam, The Netherlands: Elsevier, 2012, doi: 10.1016/C2009-0-61819-5.

AZWA ABDUL AZIZ was born in Perak, Malaysia. He received the B.Sc. degree (Hons.) in information technology from Universiti Teknologi Mara (UiTM), Malaysia, in 2004, and the M.Sc. degree in science (data quality) from Universiti Malaysia Terengganu, Malaysia, in 2010. In 2004, he started working in industry as a Data Warehouse Developer, which his last position is a Business Intelligence Consultant. In 2010, he joined Universiti Sultan Zainal Abidin (UniSZA), Malaysia, as a Lecturer, where he is currently a Senior Lecturer, with specialized in machine learning. He has published more than 15 journals with a citation to date is 236 and H-index is eight (Google Scholar). He has received multiple International awards innovation and one of candidates of Malaysian Young Scientist Award, in 2014.



ANDREW STARKEY received the Ph.D. degree in the application of artificial intelligence techniques to engineering problems from the University of Aberdeen, in 2001, and the Honours degree in applied mathematics from St Andrews University, in 1993. Since then, he has been awarded as an Enterprise Fellowship from the Royal Society of Edinburgh and Scottish Enterprise. He is also responsible for Blueflow Ltd., a spinning company with the University Aberdeen that proposed a solution for a wide range of data analysis areas, such as financial, textual, and web data, such as blogs and discussion threads, and condition monitoring.