Expert Systems With Applications 106 (2018) 197–216

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach

M. Ghiassi[a,*], S. Lee[b]

[a] Santa Clara University, Santa Clara, CA 95053, United States
[b] Stella Technology, San Jose, CA 95119, United States

ARTICLE INFO

Article history:
Received 23 May 2017
Revised 4 April 2018
Accepted 4 April 2018
Available online 7 April 2018

Keywords:
Twitter sentiment analysis
Domain transferability
n-gram analysis
Machine learning
Dynamic artificial neural networks (DAN2)

ABSTRACT

The Twitter messaging service has become a platform for customers and news consumers to express sentiments. Accurately capturing these sentiments has been challenging for researchers. The traditional approaches to Twitter Sentiment Analysis (TSA) include dictionary-based and use supervised machine learning tools for sentiment classification. This research follows the supervised machine learning approach. A major challenge for the machine learning approach is feature selection, which is often domain dependent. We address this specific challenge and present a novel approach to identify a lexicon set unique to TSA. We show that this Twitter Specific Lexicon Set (TSLS) is small, and most importantly, is domain transferable. This identification process generates a collection of vectorized tweets for input to machine learning tools. In traditional approaches, this vectorization often results in a highly sparse input matrix which produces low accuracy measures. In this research, we hierarchically reduce the feature set to a small set of seven "meta features" to reduce sparsity. We show that TSA based on these features can produce highly accurate results using a dynamic architecture for neural networks (DAN2) and SVM (machine learning tools) as measured by recall, precision, and F_1 metrics (the harmonic average of precision and recall). Our results show that a Twitter Generic Feature Set (TGFS) derived from two datasets (@JustinBieber and @Starbucks) is domain transferable and when combined with only a few Twitter Domain Specific Features (TDSF) (less than 3%), can produce excellent sentiment classification values. We evaluate the effectiveness and transferability of the TGFS across three new and distinct domains (@GovChristie, @SouthwestAir, and @VerizonWireless).

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Social media and the wide adoption of the Web introduce user feedback, reviews, and comments as consumable web content. Organizations can benefit from analyzing these user inputs to offer better services, refine product designs, improve user experience, and manage the overall organization's performance. User input is often presented online, and in the case of Twitter, the opinions are expressed in real or near real time with the potential of reaching a very wide audience in a matter of seconds. Overall as stated by Poria, Cambria, and Gelbukh (2016), "the opportunity to capture the opinion of general public … has raised increasing interest of both scientific community and the business world."

To analyze user input, organizations use sentiment analysis or opinion mining tools. Sentiment analysis is defined as "the task of finding the opinions of authors about specific entities" (Feldman, 2013). Sentiment analysis can be based on and assessed at the document, sentence, or word level. For Twitter, we use the whole tweet as the basis for our analysis and assume that the whole tweet contains an opinion or a sentiment.

For firms, taking advantage of Twitter data, however, requires them to collect, store, and analyze an immense amount of data produced by Twitter each day. In 2016, there were more than 319 million active users sending more than 500 million tweets per day (http://statista.com/statistics/282087/number-of-monthly-tive-twitter-users/). Some of the most prolific accounts on Twitter receive hundreds of thousands of Twitter messages a day (e.g. Xbox Support has more than 400,000 followers and receives more than 1.5 million tweets daily; Justin Bieber receives more than 300,000 tweets daily). During the 2014 World Cup, there were more than 672 million tweets related to World Cup (Rogers, 2014).

Examples of commercial sentiment services that offer applications to process datasets of these sizes include Lexalytics, Converseon, and Summize (Jansen, Zhang, Sobel, & Chowdury, 2009).

Abbreviations: TSA, Twitter Sentiment Analysis; TGFS, Twitter Generic Feature Set; TDSF, Twitter Domain Specific Features; TSLS, Twitter Specific Lexicon Set.

* Corresponding author.
E-mail address: mghiassi@scu.edu (M. Ghiassi).

https://doi.org/10.1016/j.eswa.2018.04.006
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

However, state-of-the-art approaches to TSA fail to offer applications with acceptable performance. Literature review of such applications lists accuracy measures between 40% and 70% (Abbasi, Hassan, & Dhar, 2014; Ghiassi, Zimbra, & Lee, 2016; Zimbra et al., 2016). For TSA to become a more effective tool for corporations in assessing their customer feedback, a better and more accurate set of TSA tools is required.

The poor performance values may be attributed to several properties of tweets that make TSA particularly challenging. Tweets are characterized by diverse, evolving language with frequent use of slang, abbreviations, and emoticons. The brevity of tweets offers relatively few terms to evaluate with a sentiment lexicon, resulting in sparsely populated tweet feature representations. Sparsity is a shortcoming of traditional feature representations and typically diminishes performance of sentiment analysis methods (Saif, He, & Alani, 2012a,b).

This study shows that development and utilization of a Twitter specific lexicon set can improve TSA accuracies, and combined with DAN2 (a machine learning tool), these accuracies can be improved noticeably. Specifically, this research makes the following contributions to existing research: we develop a reduced Twitter lexicon set specific to TSA that lowers problem complexity and increases the density of the feature matrix, which mitigates the classic feature sparsity problem.

Most TSA studies only use a three-class (positive/negative/neural) scale for their analysis. To provide actionable suggestions, we expand the number of sentiment classes from three to five, through the addition of the mildly positive and mildly negative classes. Mild sentiment expressions are likely to be of particular interest to firms and brand management practitioners.

We also address the important and intractable domain transferability problem that is present in TSA. As noted by Andreevskaia and Bergler (2008), "The development of domain-independent sentiment determination systems poses a substantial challenge for researchers in NLP and artificial intelligence." The solution presented in this research addresses this specific challenge and significantly improves domain transferability. We offer a general and reusable feature set for TSA that can be applied across domains. We show that researchers simply need to augment this feature set with a very small number of domain-specific features to generate a highly effective tweet feature representation for TSA in a new application. Our experimentation shows that 97% of the features utilized to generate the tweet feature representations are applicable across varying cases. This contribution addresses the domain transferability challenge for Twitter sentiment analysis. The study also offers DAN2 as a machine learning model for tweet sentiment classification. DAN2 produces excellent overall accuracy and has the sensitivity required to distinguish mild sentiment expressions and cope with the unbalanced class distribution, typical of Twitter datasets, which is intensified by the five-class sentiment division.

Specifically, this research contributes to TSA literature by (a) introducing a feature engineering approach to determine a TSLS using two diverse datasets, (b) uses the TSLS with DAN2 and SVM to produce excellent results, and (c) by utilizing three additional datasets from diverse domains, we evaluate and show the domain transferability of the TSLS using supervised machine learning tools.

The remainder of this paper begins by conducting the literature review in Section 2, followed by discussion of the Twitter specific feature engineering approach in Sections 3 and 4 presents data collection and preparation, and the results of our experiments are presented in Section 5. Section 6 presents our conclusion. The appendix offers supporting material for this paper.

2. Literature review

Literature for TSA offers three different methodologies: lexicon-based, machine learning, and hybrid approaches. These approaches are summarized by several researchers (Feldman, 2013; Pang, Lee, & Vaithyanathan, 2002; Thakkar & Patel, 2013; Zhang, Ghosh, Dekhil, Hsu, & Liu, 2011). In this section, we first present a summary of the strategies for sentiment analysis and follow that with literature review on feature engineering, target class specification, and domain transferability. We conclude this section with a discussion on the choice of machine learning algorithms and feature engineering approaches used in this research.

2.1. Lexicon-based vs. machine-based approaches

Lexicon-based sentiment analysis assumes that "individual words have prior polarity that are independent of context; and that said semantic orientation can be expressed as a numerical value," (Taboada, Brook, Tofiloski, Voll, & Stede, 2011). This approach results in a large set of features, and for sentence level documents, such as microblogs, will result in a very sparse representation of text. Thus, a tweet often will be represented by very few features from the lexicon set that often reaches several thousand. In general, lexicon-based approaches offer more portable solutions across domains which do not require supervision, but they often are less accurate (Oliveira, Cortez, & Areal, 2014).

Authors in Moreno-Ortiz and Hernández (2013) evaluate the use of a lexicon-based approach for sentiment analysis of Twitter messages in Spanish. They note that in "sentiment analysis, it has become a commonplace assertion that successful results depend to a large extent on developing systems that have been specifically developed for a particular subject domain."

In the machine learning approach, classifiers are mostly trained using a set of features comprised of n-grams. Researchers have shown that machine learning classifiers perform better than lexicon-based solutions. Yet, this superiority is often limited to a single domain (Kennedy & Inkpen, 2006) and is not generally transferable across applications. This limitation is one reason researchers have introduced "hybrid approaches" (Andreevskaia & Bergler, 2008). Taboada et al. (2011) report that "a lexicon-based system could outperform pure or hybrid machine-learning methods in cross-domain situations, though further research would be necessary to establish this point conclusively."

Most TSA studies, however, use the machine learning approach. Its popularity is due to its adaptability and accuracy. Machine learning approaches can be categorized into supervised, semi-supervised, and unsupervised approaches. In this research, we use the supervised machine learning approach for TSA. A significant task in this approach is labeling (scoring) of the datasets, which is often manual and may require domain experts. Although collecting a large dataset for sentiment analysis has become easier, labeling of these records continues to be challenging and this has limited the actual size of sentiment analysis models. For example, Pak and Paroubek (2010) analyzed 300,000 records, but employed a training set of only 216 records, and Go, Bhayani, and Huang (2009) manually marked only 359 tweets for their sentiment analysis model. Andreevskaia and Bergler (2008) use 200 sentences that were manually annotated by two independent judges (100 positive and 100 negative).

To address this challenge, researchers have introduced alternative approaches to manual labeling, such as using emoticons as noisy class labels (Barbosa & Feng, 2010; Chung & Mustafaraj, 2011; Ghiassi, Skinner, & Zimbra, 2013; Loughran & McDonald, 2011). They manually classify emoticons based upon their interpreted sentiment expression, then collect and classify tweets containing the emoticons. The emoticon-based sentiment classifications are

then used as noisy class labels to train a machine-learned classifier. This type of machine learning has been described as distant supervision (Chung & Mustafaraj, 2011). Distant supervision may also include the use of star ratings or hashtags (Davidov, Tsur, & Rappoport, 2010; Go et al., 2009; Silva, Coletta, & Hruschka, 2016). Although distant supervised learning might be useful for some applications, direct labeling remains the superior alternative, albeit a challenging endeavor. For this research, we create relatively large datasets and use three independent judges to manually label each tweet. The process and the details of generating the datasets are fully described in follow up sections.

Supervised classification algorithms commonly applied to TSA include Naïve Bayes, kNN, Logistic Regression, artificial neural networks (ANN), SVM and DAN2. Computational linguistic tools such as bag of words, part-of-speech (POS) information, term frequencies, etc. are used to represent tweets as information vectors. However, the choice of vocabulary (features) in text document modeling is large, and Twitter-based documents offer additional challenges due to their limited size (140 characters) and special use of language: everyday expressions, excessive use of abbreviation, acronyms, alternative spellings (love vs. luv), emoticons and emojis, use of single or double letters to imply words (u for you; ur for you are. etc.), as well as misspelling of words.

Literature on performance of supervised classification algorithms show SVM as an excellent solution for TSA (Pang et al., 2002; Salvetti, Reichenbach, & Lewis, 2006). These solutions are often trained using n-gams. However, "although such classifiers perform very well in the domain that they are trained on, their performance drops precipitously when the same classifier is used in a different domain (Aue & Gamon, 2005)." Supervised classification algorithms also rely on a large set of labeled data. Andreevskaia and Bergler (2008) observe that "the lack of sufficient data for training appears to be the main reason for the virtual absence of experiments with statistical classifiers in sentiment tagging at the sentence level."

Authors in Kontopoulos, Berberidis, Dergiades, and Bassiliades (2013) offer a semantic web-based method as an alternative formulation and attempts to identify an effective set of features for sentiment analysis. In this research, we strive to extend this approach and to identify an effective set of features that are also transferable across domains.

2.2. Feature engineering analysis

Most approaches to sentiment analysis use feature engineering to reduce the size of the vocabulary used for analysis. For TSA, the reduced lexicon set, referred to as the "sentiment lexicon," is considered "the most crucial resource for most sentiment algorithms," including those employed in this research (Feldman, 2013). Authors in Zhang et al. (2011) emphasize that "without a comprehensive lexicon, the sentiment analysis results will suffer." Determination of a reduced, robust and reusable lexicon is one of the main objectives of this paper. Three approaches are used to define sentiment lexicons: (1) the manual approach that uses selection of lexicon by hand, (2) a dictionary-based approach in which a set of words from resources such as "WordNet" is utilized, and (3) a corpus-based approach using seed words from a specific domain (Feldman, 2013). The manual approach is often domain and case specific and is time consuming. The dictionary-based approach uses seed words and their synonyms and antonyms from a source, such as WordNet. This approach is widely used and has the advantage of being domain independent but often lacks the ability to capture the embedded nuances of a field. Finally, the corpus-based approach uses seed words from a domain along with complementary adjectives and linguistic connectors (Feldman, 2013). These approaches are primarily unigram based. Some researchers (Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011) use the unigram approach as their baseline model and use linguistic analysis to introduce a feature set for use in alternative models to show that models with only 100 selected features can perform as well or even better than a unigram model containing over 10,000 features.

Additionally, the use of a unigram approach can result in inaccurate polarity representation since some words, or features, may have different polarity values within a specific context. Authors in Ding, Liu, and Yu (2008) offer examples of how a word such as "small" can indicate a positive or a negative opinion of a product feature depending on the product and the context. Similarly, Oliveira et al. (2014) state that "[using unigrams] is often at the expense of lower accuracy… and these resources may not be appropriate for specific domain contexts and types of messages used. For example, the word "long" can have many sentiment orientations within the financial domain (e.g., 'long debt list', 'long Google stocks')." Another difficulty in using a word-based approach is the integration of multiple conflicting opinion words in a sentence or a document.

To remedy the shortcomings of the unigram approach, researchers have examined semantic, syntactic, and stylistic aspects of text. Some researchers address the semantic consideration which attempts to generate semantic lexicons of specific terms to express opinion polarity (Tetlock, 2007; Turney, 2002). Others use syntactical aspects of text by introducing word n-grams and part-of-speech n-grams in their models (Pang et al., 2002; Dave, Lawrence, & Pennock, 2003). Finally, the stylistic aspect of the text has also been included in sentiment analysis (Abbasi, Chen, & Salem, 2008). These approaches often increase the feature size of the sentiment analysis model, which in turn will require additional feature reduction efforts to identify the discriminating expressions (Abbasi et al., 2008; Gamon, 2004).

Several researchers have built their feature sets using POS features, unigrams, n-grams or combinations of them to introduce many categories. Authors in Agarwal et al. (2011) use this approach to design a tree representation of tweets to combine many features. They report that combining only certain significant features with POS tags were helpful in their experiments.

Other researchers have applied feature engineering tools to reduce the size of their feature set to around 2000 (Gamon, 2004). Authors in Kennedy and Inkpen (2006) report using various numbers of features ranging from 3066 features (a simple collection of unigrams from the *General Inquirer*) to 34,718 features (unigrams and bigrams). They report that Pang et al. (2002) "found that by adding all bigrams as features, the SVM classifier performed worse than when using only unigrams," however, they state that "by selecting specific bigrams it is possible to improve over just unigrams." We show that the intelligent selection of unigrams, bigrams and even 3, 4, and 5-grams can indeed improve sentiment analysis further while reducing the modeling complexity.

For TSA, noting the short length of tweets, Andreevskaia and Bergler (2008) conclude that "using sentiment clues can negatively affect the accuracy and recall if that single sentiment clue encountered in the sentence was not learned by the system." Their study suggests that sentiment analysis of documents at the sentence level using unigrams has shown better results than higher order n-grams (bigrams and trigrams) (Andreevskaia & Bergler, 2008). Our findings are in contrast to theirs. However, they offer increased feature vector sparsity as the likely explanation for this observation. We agree with this premise. At the sentence level, the frequency of bi and tri-grams per sentence are even lower than unigrams. To see the full benefit of higher order n-grams, a large labeled dataset is required, and the sparsity concerns need to be addressed. We address these concerns in the feature engineering process we introduce in this research.

2.3. Target class specification

Most existing models use a three-class approach to classify tweets as positive, negative, or neutral. Using the three-class approach, Jansen et al. (2009) conclude that in most cases, the "majority of tweets either lack specific sentiment or are neutral." Authors in Moreno-Ortiz and Hernández (2013) state that although "most work on the field, [sentiment analysis], has focused on the *thumbs up or thumbs down* approach … a further step involves an attempt to compute not just a binary classification of documents, but a numerical rating on a scale." Alternatively, some researchers use either a continuous sentiment scale (of up to 100) or a wide range (from 0 to 10) (Moreno-Ortiz & Hernández, 2013). The problem with using a wide scale is the inability of human judges to agree on a specific sentiment score for the training data, which renders any validation of the results by human judges impractical.

We address the sentiment class issue by extending the traditional scale to one that ranges from strongly negative to mildly negative, neutral, mildly positive, and strongly positive. Brand-related tweets regularly express opinions, and these tend to range from strong sentiments in one direction or the other (Ghiassi et al., 2013, 2016).

When brand-related tweet sentiment classes are further divided into strong and mild intensities, the tweets of interest, expressing mild sentiments, are typically far outnumbered by strong/negative sentiments. Most machine learning models lack the sensitivity to distinguish strong/negative sentiment expressions from mild ones.

Even for three-class sentiment analysis, most studies use a "balanced" dataset (Andreevskaia & Bergler, 2008; Pang & Lee, 2004; Taboada et al., 2011). However, for Twitter datasets, the unbalanced sentiment distribution is prevalent. Although, some researchers have noted that the number of positive sentiments in a corpus dominate the negative ones, especially when defining a five-class scale, we can validate the skewness of the sentiment distributions, however, depending on the type of subject matter, the skewness could be either positive or negative.

In Ghiassi et al. (2016), they report results for two distinctive brand-related Twitter datasets (Starbucks and Governor Christie) using both three-class and five-class models. The models are tested using DAN2 and two state-of-the-art TSA systems from the academic and commercial domains, Sentiment140 (www.sentiement140.com) and Repustate (www.repustate.com). They show that the performance of Sentiment 140 and Repustate degrades when the number of classes increase from three to five. However, DAN2 outperforms these systems by a wide margin for both the three and five-class models.

2.4. Domain transferability

As stated earlier, the supervised machine learning approach to sentiment analysis can be accurate and effective. However, it often lacks portability across diverse domains. In general, most sentiment analysis applications are domain specific and do not perform well when applied to other domains. This problem is known as the domain transfer problem (Aue & Gamon, 2005; Blitzer, Dredze, & Pereira, 2007; Dredze, Blitzer, & Pereira, 2007; Tan et al., 2007) and is a significant challenge in sentiment analysis research.

In general, the domain transferability experiments conducted by Aue and Gamon (2005) were not encouraging. Some researchers (Andreevskaia & Bergler, 2008), acknowledging these conclusions, have offered a solution that partitions the problem domain into a "general" and "domain specific" solutions. Yet others (Dredze & Crammer, 2008; Dredze et al., 2007; Samdani & Yih, 2011) use the "structural correspondence learning" approach for "domain adaptation" for sentiment analysis of product reviews. They show that for "domains that are reasonably similar," domain adaptation can be achieved. Similarly, Tan, Wu, Tang, and Cheng (2007) augment out-of-domain labeled examples with unlabeled ones from the target domain to address the domain transfer problem. Their approach relies on "Similarity Ranking and Relative Similarity Ranking algorithms" to report a 15% gain using SVM. Andreevskaia and Bergler (2008) conclude that "Overall, the development of semi-supervised approaches to sentiment tagging is a promising direction … but so far, the performance of such methods is inferior to the supervised approaches with in-domain training and to the methods that use general word lists. It also strongly depends on the similarity between the domains." They use three different domains to train their SVM-based classifiers and use a fourth domain as an evaluation set.

Developing a transferable lexicon for a specific domain may benefit from the concept of "similarity." Oliveira et al. (2014) introduce an "adaptation" method that combines existing linguistic measures with a newly proposed "Weighted Class Probability," to show that their model produces better results when compared with a set of six reference lexicons.

Another approach to the domain transferability problem is *transfer learning*, which leverages existing knowledge from known domains to aid in tasks, such as feature selection, in new domains. The motivation for transfer learning in the context of machine learning comes from the human learning experience. "Humans encounter a continual stream of learning tasks. They do not just learn concepts or motor skills, they also learn *bias*… As a result, humans are often able to generalize correctly from extremely few examples" (Thrun & Pratt, 2012). In the context of sentiment analysis, datasets from different domains will often contain properties such as different sentiment distributions and domain specific language. The ability to use the knowledge gained from previous analysis and apply it to new unlabeled data is very valuable as the lack of sufficient labeled data is the most prohibitive factor in Twitter-based studies on sentiment analysis.

Some researchers have followed the "ensemble" or "all-in-one" approach to address transferability concerns. Dredze and Crammer (2008) offer the Multi-Domain Regularization framework, Samdani and Yih (2011) introduce the Ensemble Learner, Mansour, Refaei, Gamon, Abdul-Hamid, and Sami (2013) the all-in-one approach, while Poria et al. (2016) and Glorot, Bordes, and Bengio (2011) suggest using deep learning to address transfer learning.

Pan and Yang (2010) categorize transfer learning under three sub-settings: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. Of these, the transductive transfer learning use-case is most applicable to text classification; the source and target tasks are the same, while the source and target domains are different. This technique has been applied successfully to text classification problems in Arnold, Nallapati, and Cohen (2007) and Daume III and Marcu (2006). Arnold et al. (2007) provide a comparative study of maximum entropy, support vector machines and naïve bayes models using conventional inductive and transductive approaches for domain adaptation. These authors observe that the benefit of labeled data from the target domain cannot be overcome by large amounts of labeled data from the source domain. We note that both studies use datasets which are characterized by highly specialized language which are not expected to transfer well between domains: news data from the Wall Street Journal, broadcasts from CNN, NPR and ABC Primetime and a dataset comprised of abstracts from biological journals.

The Twitter domain transferability approach developed in this research is generalized, and although it depends on addition of few domain specific lexicons, it is neither restricted to a specific sector (Oliveira et al., 2014) nor does it rely on similarity metrics as used in other studies (Andreevskaia & Bergler, 2008; Tan et al., 2007).

We note that the majority of existing TSA solutions are tailored for specific domains and while many offer reasonably good accuracy, these solutions are not transferrable.

In this research, we use Twitter sphere as our data source, where we can expect a high degree of generality in the language used, regardless of the topic domain of the target Twitter account. As such, we view the Twitter sphere as a single large dataset for which we can define a lexicon which captures generic features across all of Twitter. We posit that although the language used may differ between domains due to factors such as user demographics, domain, and topic, users express common notions which are domain transferable. Capturing these notions is very challenging, as the language that can be used to express any single notion is highly variant. To address this challenge, we include multiple datasets from varying domains from which to build our feature set. The goal is to collect sufficient source data with enough variation such that our lexicon is relatable to any other domain we encounter on Twitter.

In this paper, we introduce a hierarchical model that produces a domain agnostic lexicon set complimented by a small sets of domain specific features which are particular to the unique characteristics of the Twitter language.

2.5. Properties for the feature engineering approach

In general text processing and classification analysis using supervised learning methods, researchers use feature engineering to represent text as a vector of features.

The text expressed in social media in general, and in Twitter in particular, offer a different writing style that is characterized by slang, alternative spelling, and abbreviations. Researchers examining sentiment expressed in tweets have also noticed that there is often a strong positive or negative slants toward a subject. These differences between a twitter corpus and a general textual corpus require special considerations for lexicon construction for TSA. Authors in Taboada et al. (2011), Ghiassi et al. (2013), 2016), Lexalytics (2016), and Zimbra, Ghiassi, and Lee (2016) have addressed these concerns. Ghiassi et al. (2013), 2016) have offered a set of properties to address these Twitter specific properties.

2.5.1. Twitter specific corpus properties

The nature and size of tweets differ from standard texts. These differences necessitate a feature engineering approach tailored for TSA. For example, one popular method used to select features for standard text processing research is the TF/IDF weighting scheme (Ghiassi, Olschimke, Moon, & Arnaudo, 2012; Joachims, 1999; Manning, Raghavan, & Schütze, 2008). This approach places value on highly discriminating terms in the document corpus.

Ghiassi et al. (2013) acknowledge these unique characteristics of tweets and introduce a set of four TSA properties (representing the underlying assumptions for Twitter texts) for feature selection. We highlight these properties and use them in this research as well. We also introduce one additional property. The four properties from Ghiassi et al. (2013) are briefly presented next.

Property 1: Strong Sentiment Terms are Pervasive. This property addresses the use of TF/IDF weighting approach and highlights the conflict it may present. For instance, in twitter, the word "Love" and "Luv" or the acronyms "lol," representing the phrase "laugh out loud", are used interchangeably. Using TF/IDF approach, will result in underweighting "Love" while the feature "Luv" would be more strongly weighted (with Luv occurring less frequently than Love). Clearly, these terms should be treated the same, (e.g. we correctly assume that these terms are semantically identical).

Property 1. *In Twitter analysis, the pervasive use of a few strong sentiment terms, and the light use of analogous terms, must be tolerated by the feature construction and feature weighting approach.*

This property only applies to terms (or acronyms) with the same polarity but different spelling of the same word (or phrase). Terms with the same polarity but different root word (e.g. Terrific vs. excellent) are dealt with using equivalence partitioning concept, introduced in Section 3. We acknowledge that for cases in which a word might represent multiple meanings, such as "bad" sometimes meaning "good" in a sentence, this property cannot and is not applied.

Property 2: Syntactical Context Must Determine Feature Boundary

Stemming is a frequently applied step in traditional text processing. However, stemming presents yet another challenge for Twitter sentiment classification. For example, Ghiassi et al. (2013) show that a stemmed query against the corpus for the term "hate" (e.g. "hate%") results in a record set somewhat evenly divided between authors that "hate" the subject (Justin Bieber) and authors that want to defend Justin from the "haters." If a feature is allowed to include the full set of Tweets found with the stemmed query ("hate%") that feature would not be able to clearly describe sentiment in one direction or another (due to the inclusion of both "hate" and "hater"). Clearly, the stemming process in TSA needs to take this concern into its feature selection process.

Property 2. *The ability to determine precise feature boundaries must be included in the feature construction approach.*

Application of this property is a heuristic process that benefits from domain specificity.

Property 3: The "Please Follow Me" Phenomenon – Unavoidably Broad Classes

Twitter sentiment categorization often creates message categories that are unavoidably large. For example, in the Twitter network, a common activity is to expand one's set of followers by asking a high-profile individual to "follow me." When using term frequency analysis for feature selection, the "follow me" (and its variation) can account for more than 35% of all tweets (Ghiassi et al., 2013).

For some Twitter handles, we find the "Please Follow Me" phenomenon to be prevalent while for some others not as much. We believe this to be a result of the choice of subjects and their user bases; for example, for @JustinBieber, an entertainer, the follow request is very prominent whereas for @Starbucks it is not. However, for all subjects, the follow requests contain some variant of "please" about 30+% of the time (@JustinBieber, 45%, @Starbucks, 30.55%, etc.). Therefore, the sentiment term "please" may not effectively recognize sentiment outside of "follow" requests because follow requests may account for a large percentage of incoming tweets for a particular Twitter account.

Property 3. *The feature weighting approach must be able to weigh the performance of features that are associated with wide sentiment classes. Furthermore, to increase the precision of the wide sentiment classes, the affinity that unigram terms show for one another must be considered at the time of feature engineering.*

Property 4: Real World Bias toward Strongly Positive/Negative Sentiment

Researchers report that the Twitter sentiment classes are not balanced and that there is an overall slant toward positive sentiments (Ghiassi et al., 2013; Lexalytics, 2016; Taboada et al., 2011). This research, however, shows that for some subjects, the slant can also be toward the negative. Any Twitter sentiment analysis model, therefore, should be able to account for this imbalance in both the feature engineering and the final machine learning processes.

Property 4. *The unbalanced classes found in the Twitter corpus require the feature engineering approach to normalize for the larger*

positive/negative sentiment class when measuring term frequency. Unbalanced classes must also be considered when determining the affinity a term has for a sentiment class (a key element of feature boundary engineering).

We next introduce one additional property for Twitter sentiment analysis.

Property 5: Usage of Twitter Handle Rather Than the Word Itself

Sometimes authors create ambiguities with regards to the sentiment target by using "@target" to represent the word "target" or the phrase "at target." For example: "Just saw woman @Starbucks pour sugar into her triple shot for 30 s (I counted). Must be a writer. #AmWriting #AmImpressed" or "Everyone who sits in this particular chair in @Starbucks takes their shoes off immediately. I don't understand." In these examples, the authors are describing an event which takes place at Starbucks and express sentiment about that event. That sentiment does not necessarily represent the author's feelings towards the Starbucks brand.

Property 5. *The usage of multiple user handles within a tweet or the usage of a user handle as the word itself or derivative obscures the intended target of any expressed sentiment*

2.5.2. Special considerations: follow, RT, emoticons, emojis, and hashtags

As discussed earlier, in Twitter, "follow requests" may be prevalent, and the term "follow" must receive special consideration. Similarly, we notice that at least 19% of all messages that mention @subject are retweets of an original message. The impact of a subject's own sentiment in retweet messages must be handled appropriately as well. Identifying these in the model is quite simple. To identify retweets, the model simply has to look for "RT," the acronym that appears before a retweeted sentence. To identify follow requests, the model simply must search for occurrences of "follow" and the misspelled variants (e.g. "folllow").

Another common characteristic of Twitter messages is the use of emoticons and emoji character sets. Although some researchers have dismissed these characters as noisy and not containing adequate information, our analysis demonstrates that sentiment models can benefit from the information held in these characters. We find the information contained in emoticons and emojis to be helpful and the lexicon developed in this research accounts for their existence. We use the emoticon list provided by Ghiassi et al. (2013) and augment it with emoji found during our investigation. Each emoticon and emoji has been selected during scoring by our evaluators to ensure relevance. The list of emoji found is provided in the Tables A.1 and A.2 in the Appendix.

As part of preprocessing, emoticons and emojis are converted to placeholder values so that they can be represented by a single, distinct feature in the feature set. In the case of emoji, we convert each symbol found to its code point or surrogate pair values. These values are cross referenced with our list of positive and negative emoji, provided in Table A.1 and A.2 in the Appendix. Emoticons and emoji are represented in the model by the features HAPPY_EMOTICON and SAD_EMOTICON.

Hashtags are intended to be used as topical markers for tweets. Twitter defines a hashtag as the following: "The # symbol, called a hashtag, is used to mark keywords or topics in a Tweet. They were created organically by Twitter users as a way to categorize messages." (https://support.twitter.com/articles/49309-using-hashtags-on-twitter). However, in practice, users also commonly use hashtags in a way more analogous to emoticons. For example, the following tweet uses hashtags as both a categorical marker and emoticon:

"Dang it my @Starbucks doesn't have #PSL yet but they did tell me Tuesday! #cantwait #Starbucks #pumpkinspice, #PSL." "#Starbucks" and "#pumpkinspice" can be seen as categorical markers. However, "#cantwait" is more of an expression of the user's excitement rather than a topical marker. Another example is the tweet:

"The slowest @Starbucks are located in Santa Clara, CA! #ridiculous #ineedtogettowork #getthesebaristasmoretraining." The user here has strung together the words of a phrase they wish to express as a hashtag. Many n-grams related to these hashtags already exist in our feature set. However, in order to capture the sentiment expressed in these hashtags, we need to add the associated unigram form (e.g. "I need" into "ineed"). The precision of unigram features must also be reassessed so that they are not mistakenly captured from the use of hash tagged phrases.

2.5.3. Machine learning algorithms for TSA

In Ghiassi et al. (2013, 2016), the authors compare the performance of DAN2 with an earlier version of the TSLS (187 features) to demonstrate this approach. In Ghiassi et al. (2013), they use Lexalytics (https://www.lexalytics.com), and in Ghiassi et al. (2016), they use OpinionFinder (Wilson et al., 2005) as sentiment analyzers to demonstrate the effectiveness of TSLS as a new feature engineering method.

In this research, we use two diverse domains to train our two classifiers (DAN2 & SVM) and use three additional datasets as evaluation datasets. We agree with Taboada et al. (2011) that "To a certain degree, acceptable performance across a variety of datasets, and, in particular, improved performance …, provides evidence for the validity of [the method]."

This research contributes to TSA literature by (a) using the TSLS with DAN2 and SVM to produce excellent results, and (b) by utilizing datasets from diverse domains, we show the domain transferability of the TSLS using a supervised machine learning approach. We also use TSLS and other lexicon sets for a comparative analysis using SVM. Finally, we present DAN2 as an alternative machine learning tool with the ability to effectively address the presence of imbalanced datasets. A brief overview of DAN2 is presented in the Appendix.

3. Twitter specific feature engineering

The feature engineering methodology developed in this research follows a multi-phase, hierarchical approach. We seek to identify such a minimal feature set while satisfying three important constraints: (1) achieving a high degree of accuracy as measured by standard metrics, (2) meeting a coverage metric criterion that ensures presence of one or more significant feature in every tweet, and (3) satisfying the "domain transferability" or generality of the feature set across diverse domains.

To achieve these objectives, we present a process that partitions the selection of features into a reusable feature set (*Twitter Generic Feature Set, TGFS*) that can be utilized across domains. TSA for a specific domain then needs only to add some domain specific features (*Twitter Domain Specific Features, TDSF*) to build a lexicon ready for sentiment analysis. We present TSA results from several different domains to show the effectiveness of this approach using DAN2 and SVM.

The Twitter specific feature engineering process introduced in this research applies the traditional computational linguistic measures and hierarchically reduces the feature set by introducing *"feature grouping"* and "*meta features"* to represent a set of n-grams. The process architecture is depicted in Fig. A.2 (In the Appendix) and the process steps are presented in Fig. 1.

The steps described in this process begin with data collection, data cleansing, and tweet labeling (Fig. 1, steps 1–4). These are standard activities and are further described in Section 4. In this section, we describe the remaining steps (Fig. 1, steps 5–17) that are needed to develop the TSLS. We continue the work of

Steps	Level of Automation	Tools
Step 1: Collect (Minimum 40000) tweets for @subject.	Automated	Collection tool from Ghiassi et al. (2013) written in C#
Step 2: Divide collected tweets into candidate sets based on the last digit of their tweet ID and select the candidate set with a distribution most similar to that of the total set.	Semi-automated[1]	MS SQL Server, Excel
Step 3: Data Cleansing – Remove all retweets & spam.	Semi-automated[2]	MS SQL Server
Step 4: Label tweets and select n-grams for analysis.	Manual	
Step 5: Generate a list of unique terms by decomposing n-grams collected by evaluators into unigrams.	Automated	Java
Step 6: Use term frequency to build a unigram set which satisfies a given coverage metric (α*). Standard stop words, pronouns and quantity determiners are excluded.	Automated	Java
Step 7: Review excluded terms for synonyms of the filtered set and add them where applicable.	Manual	
Step 8: Apply term boundary rules.	Manual	
Step 9: Add synonyms & Antonyms to the feature list.	Using dictionaries	
Step 10: Add unigram negation terms	Using dictionaries	
Step 11: Reintroduce n-grams to the feature set through affinity analysis.	Semi-automated[3]	Java
Step 12: Perform equivalence partitioning (stage 1).	Manual	
Step 13: Add polarity tags to the feature set.	Semi-automated[4]	WEKA
Step 14: Perform equivalence partitioning (stage 2).	Manual	
Step 15: Add valence shifters to the feature set (intensifiers, diminishers, negations and sarcasm markers).	Semi-automated[5]	Java, MS SQL Server
Step 16: Set weightings for polarity tags.	Manual	
Step 17: Categorize features by aspect.	Manual	

[1] Candidate set generation is automated. However, the selection process is manually performed.
[2] Markers for retweets and spam are manually extracted.
[3] Affinity scores are generated in an automated fashion after n-grams have been manually grouped.
[4] IG/GR values are generated through WEKA. Assessment of these values is performed manually, but can be done in a semi-automated way.
[5] After intensifier and diminisher frequencies are generated, sample tweets are queried for and manually investigated to find n-gram variants.

Fig. 1. The feature engineering process.

Ghiassi et al. (2013), who used the Justin Bieber dataset, and introduce new datasets. We expand our dataset domain by adding the Starbucks dataset. The striking diversity observed between these two domains allows us to construct a TSLS with a high chance of domain transferability. In the follow up sections, we will examine this premise by introducing three more diverse domains, and we apply the resulting TSLS to them to assess the domain transferability.

We introduce a hierarchical feature engineering method for twitter feature selection and reduction. At this stage, a primary objective of our feature engineering approach is the reduction in the size of the feature set. We, therefore, offer a feature engineering approach that (a) reduces the feature set, (b) creates a denser feature/tweet input matrix, and (c) ensures that no significant feature is excluded from the final feature set. Our approach adheres to the feature engineering guidelines recommended by Mitchell (2005), who suggests that to maximize the predictive ability of a reduced set of features, one should consider the following six attributes: information loss, bias, noise, collision of negative and positive semantic vectors, differences in scale, and overfitting.

3.1. Twitter specific lexicon set (TSLS)

In this research, we present modifications to the feature engineering process used in Ghiassi et al. (2013) to address the domain transfer problem. Mansour et al. (2013), address domain transferability using the "all-in-one" approach. They train a classifier using data from multiple message domains simultaneously. The classifier is then tested against each message domain individually. The results show that this "all-in-one" classifier performs comparably well, if not better, when compared to more sophisticated domain adaptation approaches, such as those previously listed. We take a similar approach to domain adaptation and view the Twittersphere as a single, large dataset. We use frequency analysis, entropy measures, valence shifters, polarity scoring and aspect folding and apply it to four new datasets to address domain transfer-

ability and improve sentiment classification performance. Detailed description of these steps follows next.

3.1.1. Supervised feature engineering – frequency analysis

The application of feature engineering steps is most effective once the collected dataset has been cleansed and labeled (Fig. 1, steps 1–4). The feature selection process can then begin with calculating frequency measures (step 5). The data gathering step collects n-grams of interest, which are decomposed into unigram form. Unigrams with term frequency greater than α (α is user defined) are selected for feature engineering (standard stop words, pronouns and quantity determiners are excluded). The value of α is set to ensure that selected features cover a minimum number of tweets (95%) contained in the associated dataset (step 6). The goal of term frequency evaluation is to arrive at a reduced lexicon that can be generically applied to twitter messages by future investigators. The term frequency parameter is experimentally defined to ensure that the resulting reduced feature set has general applicability

The data gathering step also results in generation of n-grams of interest by our evaluators which are decomposed into unigram forms. Unigram terms with term frequency larger than α (α is user defined) are selected for feature engineering (standard stop words, pronouns and quantity determiners are excluded). The value of α is set in order to ensure that selected features cover a satisfactory number of tweets (95%) contained in the associated dataset (step 6). The goal of term frequency evaluation is to arrive at a reduced lexicon that can be generically applied to twitter messages by future investigators. The term frequency parameter is experimentally defined to ensure that the resulting reduced feature set has general applicability. In our previous analysis, we started with the Justin Bieber dataset, which produced 755 terms with $\alpha = 0.033$. For the next dataset, the Starbucks dataset, we start with the feature set from Justin Bieber dataset, and introduce additional unigrams and n-grams for $n = 2$–5 until our coverage metric of 95% is met ($\alpha = 0.01$). Approximately 1300 terms are selected for the Starbucks dataset. Our additional dataset analyses follow the same procedure and build on the existing lexicon.

We note that for the Starbucks dataset a larger number of terms are needed, compared to the analysis of the Justin Bieber dataset, in order to meet our coverage metric. This suggests a much higher variance in the language used in tweets sent to @Starbucks compared to @JustinBieber. The selection of Starbucks is intentional to create dispersity among the domains being examined. We note that user demographics and the opinions expressed between these two domains differ significantly. During the time-period of the Justin Bieber experiment (2011), he was 17 years old, and his Twitter audience was a much younger age group compared to the Starbucks dataset. Assessing the activity in both feeds, Justin Bieber received over 10 million tweets in a one month period compared to approximately 440,000 over a 3-month period for Starbucks. Furthermore, the language found in tweets during the Justin Bieber experiment was less sophisticated and the vocabulary used was less varied than what is found in the Starbucks corpus. We, therefore, expect a large feature variation between the two corpuses and our results validate this premise. Our other datasets are similarly selected.

During frequency analysis, we find that many unigram terms are synonymous with each other (e.g. “awful”, “horrible”, “terrible”, “dreadful”, “atrocious”, and “horrendous”). If all terms in a set of synonyms do not meet the term frequency requirement individually, but the sum of their frequencies does, meaningful terms can be lost. Therefore, in step 7 we implement this rule and we sum the frequencies of synonymous terms before applying the term frequency requirement.

The next step (step 8) addresses feature boundary definitions. Similar to the approach used in Ghiassi et al. (2013), we develop a feature boundary for each term. To determine the appropriate boundary, the affinity of each term to each sentiment class is examined. Terms that do not clearly indicate sentiment are either set aside for later n-gram engineering or dropped altogether. For the terms that remain, synonyms and antonyms are found and added to the feature definition (step 9). The feature boundary work results in features that are both highly explanatory and combinable with other features of similar sentiment affinity.

We also examine contradictory terms. Ghiassi et al. (2013) use the example of “hate” and “haters” to illustrate the necessity of accounting for such features when applying stemming. To address this concern, we applied stemming after determining feature boundaries. At this junction, we are left with approximately 1000 features.

Step 10 forgoes the inclusion of frequently occurring bi-gram negations employed in Ghiassi et al. (2013), but keeps the unigram variants (e.g. can’t, won’t, don’t, isn’t, not, etc.) as ancillary terms to help deal with negation. The bi-gram negations are covered through the implementation of valence shifters; negations are now collected in n-gram form (e.g. “not”, “not very”, “not really”, etc.). A full discussion of negation and valance shifters is presented in the following sections.

3.1.2. Supervised feature engineering – affinity analysis

The next step in our feature engineering approach is the inclusion and addition of n-grams to the unigram set (step 11). We use the affinity method for n-gram selection. The *Affinity* of a word phrase P is defined as (Kajanan, Shafeeq Bin Mohd Shariff, Datta, Dutta, & Paul, 2011)

$$Affinity(P) = f(P)/min_{\forall w_i \in P}(f(w_i))$$

where $f(P)$ is the frequency of phrase P; min $(f(w_i))$ is the minimum frequency across the words in phase P. The application of affinity method allows us to select a set of n-grams that have “higher collocation frequencies relative to individual occurrence frequencies of the constituent unigrams.” This process expands the feature list. To manage and control this expansion, we remove the constituent word unigrams from the feature list. For example, from the Starbucks tweets corpus, affinity analysis identifies higher order word n-grams like “Pumpkin Spice Latte”, “Christmas Blend”, “Passion Tea”, “horrible experience”, and “customer service.” To further manage bias in the feature set, n-grams are only added to features that are still lacking in precision after the initial refinement. For example, the stemmed “hope%” search term is not as precise as the following set of n-grams {“%hopeful%”, “%I hope u%”, “%I hope you%”, “%I hope he%”, “% hoping%”, “%do not lose hope%”, “%don’t lose hope%”} and is replaced with the n-gram set. Meaningful n-grams are added to the feature set in order to once again regain the 95% coverage rate metric. This results in a total feature set of 1078 features (unigrams and n-grams) for the Starbucks dataset.

When applying the affinity method to our n-grams, we encounter a similar phenomenon as when dealing with the unigrams. As previously mentioned, due to the variance in the way the same root n-gram can be expressed, alternative n-grams may also exist for a root n-gram which receive a low affinity score. An example of this can be seen in the tri-gram “don’t lose hope.” An alternative n-gram form is “don’t ever lose hope”. Because the added term (“ever”) is commonly used, both n-grams evaluate to the same minimum frequency across the words contained in each phrase. However, “don’t ever lose hope” occurs much less frequently, so it receives a lower affinity score. In order to resolve this issue, we sum the frequency of all synonymous n-grams and define the min-

imum frequency of words contained in a phrase as the minimum frequency word contained in all n-grams in that group.

3.1.3. Supervised feature engineering – equivalence partitioning stage 1

Following n-gram analysis, we employ equivalence partitioning, which groups features expressing similar concepts, as in Ghiassi et al. (2013). However, we perform this step in two stages. In this first stage, step 12, we merge new features found in the Starbucks dataset with the existing lexicon (from Justin Bieber), and we partition our features based on a root word or phrase. By "root word" we mean using a root word proxy with the same sentiment meaning (e.g. *horrible, terrible, awful,* and *atrocious*). This step significantly reduces the feature size. We then combine like features to arrive at even smaller number of *"feature groups."* The last step is formed by following the principle of equivalence partitioning. The grouping now contains unigrams and decomposed n-grams ($n = 2$–5) that express equivalent sentiment. For example, the POS_LOVE feature group is composed of the n-grams "love", "lovin" and "luv". The THANK feature group is composed of the n-grams "thank", "thx" and "thks." The HI feature group includes 9 variants of "hi." In a limited set of cases the grouping includes unigrams that have some minor degree of difference in terms of meaning, but were found to be used interchangeably on the Twitter service. For example, the AMAZIN feature group is composed of "amazin", "amaze" and "brilliant." This step allows us to substitute individual "features" with their corresponding "feature group" thus reducing feature set size while still enabling to characterize 95% of the messages in the greater Twitter corpus (e.g. one or more of the "feature groups" appeared in 95% of the tweets – maintaining our coverage goal). This indicates that no explanatory power is lost in the reduction from individual features to the feature group. This step produces a significantly smaller and denser input matrix for our analysis. In this research, we have extended n-gram inclusion for $n = 2$, 3, 4, or 5. The increase in maximum n-gram size is implemented to account for the variation in the presentation of n-grams which are equivalent. For example, the *"don't lose hope", "don't ever lose hope",* and *"don't you ever lose hope"* are considered as equivalents. Similarly, *"makes my day", "made my day", "made my entire day",* and *"made my whole day"* are considered to be variations of the same concept.

This process reduces the number of feature groups to 542 which spans both the Justin Bieber and Starbucks datasets. Once we applied this new feature set to the other three new domains, a 90–95% coverage metric held for all datasets. This observation supports that the 542 features identified from the Justin Bieber and Starbucks datasets are comprehensive enough to be applied to other domains; the domain transferability property is now mostly achieved.

3.1.4. Supervised feature engineering – polarity designation

Next, we examine polarity designations of our features. Following this step (step 13), we will perform another iteration of equivalence partitioning. Individual features may express varying intensities of the concept that they are grouped by. As such, this step is performed here to evaluate the base or root representations of our features before any further categorization is performed.

In Ghiassi et al. (2013), the authors tag the features with "POS" (positive), "NEG" (negative) or "MIXED" to indicate general polarity of each feature.

We use information gain together with gain ratio, as a recommended approach (Sharma & Dey, 2012), where necessary to reassess and update polarity of our features. We use the WEKA data mining application (http://www.cs.waikato.ac.nz/ml/weka/) to evaluate both measures across each of our sentiment classes. High information gain and gain ratio for a given class indicate that a feature should be assigned a polarity associated with that class. Features previously tagged as "MIXED" have low information gain and require further investigation. Any other features with unexpectedly low information gain are similarly investigated. In this research, the features are alternatively tagged with xp (extremely positive), vp (very positive), sp (somewhat positive) sn (somewhat negative), vn (very negative) or xn (extremely negative) to indicate general polarity. The "xp" and "xn" tags are added for those terms which dominate sentiment regardless of the presence of any other features. Features with no information gain towards any class are considered domain specific (e.g. pos_neversaynever) and left untagged. We note that during the manual evaluations, there are no neutral features since evaluators select what they considered the most meaningful terms/phrases.

3.1.5. Supervised feature engineering – equivalence partitioning stage 2

We next perform a second round of equivalence partitioning (step 14) on the feature groups resulting from the previous step. Since at this point the feature groups are partitioned by their root word, there are many instances where feature groups are similar enough in both meaning and usage, such that they can be represented by a single equivalence class. For example, we create an equivalence class named HORRIBLE which contains *horrible, terrible, awful,* and *atrocious.* Similarly, the feature group named NEGLECT contains *notice me, neglected, ignored,* and *abandoned.* This process is performed manually, and features are grouped per the semantics of their usage within our datasets. A more complex example is the feature group ADDICTED. This feature group contains n-gram phrases which represent addiction as well as n-grams formed around the term "addict" and its synonyms. Grouped features may also represent varying intensities of the concept they describe. The feature group DECEIVED contains the feature groups "credibility", "deceptive", "misguided", "liar" and "unbelievable". The first three features are of a mild intensity, while "liar" and "unbelievable" represent a severe degree of deception.

Table 1 below gives a breakdown of the structure of the examples given above. The name of a group is chosen such that it clearly represents the meaning of the features contained within it. However, it is arbitrary as to which term or phrase is specifically chosen should there be multiple candidates. This second pass of equivalence partitioning results in a final count of 211 feature groups for the Starbucks dataset (204 generic, 7 Starbucks-specific). We find that the generic set of feature groups is small, transferable and can be used to describe our other datasets as well.

3.1.6. Supervised feature engineering – negation and valence shifter analysis

Valence shifters are terms that can change the semantic orientation of another term (Kennedy & Inkpen, 2006; Polanyi & Zaenen, 2006). Negations, such as "not", are an example of a valence shifter. The next stage (step 15) in the feature engineering is negation and valence shifter analysis, which focuses on adding negated, intensified, and diminished forms of the word gram features identified in prior stages to the tweet feature representations.

Choi and Cardie (2008) offer a "compositional view" approach to valence shifters by calculating term polarity and then applying "inference rules" to compute a combined polarity score. Taboada et al. (2011) offer a simple approach to negation by just reversing the polarity of the lexicon item (switch negation) (e.g. *good (+3)* and *not good (−3)).* They note that one problem with this designation is that the negation factor might be at a distance from the term itself which will require a backward search. Kennedy and Inkpen (2006) examine the effect of two additional types of valence shifters: intensifiers and diminishers.

Table 1
Equivalence class structure.

Equivalence class name	HORRIBLE	NEGLECT	ADDICTED	DECEIVED
Feature groups contained	horrible, terrible, awful, atrocious	notice me, neglected, ignored, abandoned	addict, can't stop, hooked, i have a problem, not ashamed	credibility, deceptive, misguided, liar, unbelievable

Table 2
Valence shifters (intensifiers & diminishers).

Intensifiers	Diminishers
Absolute, badly, biggest, epic, specially, eternally, exceptionally, extremely, freakin, fuckin, hella, huge, incredibly, major, massive, mighty, most, deatly, ever, really, ridiculous, significant, So, such, super, truly, ultimate, undoubtedly, very	Except, not, but, kind of, pretty, somewhat

We select our valence shifters using the dictionary of intensifiers and diminishers presented in the General Inquirer (http://www.wjh.harvard.edu/~inquirer/) as overstatements and understatements respectively. The General Inquirer contains 576 overstatements and 278 understatements. Each word is examined for its frequency in conjunction with features from our lexicon. However, we keep only those words which are frequently found together with our features resulting in selection of 29 intensifiers and 6 diminishers. Table 2 lists the 29 & 6 valence shifters used in this research.

We take an n-gram approach to valence shifters because it is more accurate than a proximity based approach. We consider negations by employing and keeping frequently occurring bi-gram forms along with the negation itself in unigram form (as discussed earlier). However, we encounter problems in our datasets due to subtle variation in the way users express the same phrase. For example, "not very happy", "not really happy" and "not that happy" all express the same sentiment. We address this issue by including the n-gram form of the negation (e.g. "not very") as a supplementary feature.

We notice that Twitter's users at times repeat characters or words within a message to signify emphasis. We did not find this repetition to add explanatory power (e.g. one happy face emoticon was grammatically identical to many happy face emoticons). We, therefore, ignored repetition and used only one occurrence of the feature in our analysis.

3.1.7. Supervised feature engineering –sarcasm

The presence of sarcasm in any textual document exasperates the difficult task of sentiment evaluation. Creating a system that can automatically detect sarcasm has been a challenge for researchers (Jiang, Yu, Zhou, Liu, & Zhao, 2011; Kim & Hovy, 2004). For Twitter sentiment analysis specifically, the literature discussing sarcasm suggests using "keywords" or "hash tagged words" to analyze the presence of sarcasm with some degree of success. In step 15, we follow this approach and select a number of keywords (such as "thanks", "#smh", #not") that are followed by negative (or positive) sentiment markers, and assigned a slightly higher (or lower) weight to the negative (or positive) feature to counterweigh the presence of the sarcastic qualifying terms. We augment this automated approach with additional manual examination of the tweets to manage sarcasm detection effectively.

3.1.8. Supervised feature engineering – feature sentiment scoring

The next stage in the feature engineering process is feature sentiment scoring, which assigns weights to our polarity tags (step 16). Ghiassi et al. (2013) define a simple staggered scale to define steps of sentiment polarity; (1, 2, 4, 8, 10) for positive sentiments and (−1, −2, −4, −8, −10) for negative sentiments. They use these values to quickly calculate starting point values (for machine learning tools) by representing sentiment as a linear function of these weights mapped to polarity tags or specific features. We use this set as a starting point and adapt it to our set of polarity tags for quick sentiment estimation as well. Our features are weighted with {16, 8, 4, 0, −4.1, −10.1, and −20.1} intensities. These weights map to {xn, vn, sn, neutral, sp, vp, xp}, respectively. The additional intensity weight and offset assigned to terms in negative sentiment groups is designed to ensure that negative tweets requiring the intervention of decision makers are not mistakenly classified. Negative scores are staggered such that two mildly negative scores do not result in a very negative classification, and in the case of an extremely negative feature (−20.1), the presence of combinations of positive features (e.g. sarcasm case), do not result in a somewhat negative classification (e.g. "Starbucks hot chocolate is awful. You're welcome. @Starbucks.")

3.1.9. Supervised feature engineering – aspect categories

As part of our analysis, we discover that the average number of features present for any given tweet is very low (2.74). This results in a sparse input matrix and also suggests that many features overlap. Gamon (2004) asserts that reduction of vector size can lead to improvements in sentiment classification if features are not noisy or redundant. Thus, we offer one additional step to our feature engineering process (step 17).

The final step in our feature engineering process is aspect categorization. Sentiment analysis literature defines subjects as "entities" with attributes or "aspects". In this context, sentiment can be applied to an entity or to attributes or aspects of the entity (Liu, 2015). Taboada et al. (2011) define an "aspect or an opinion target as the opinion expressed in the given document." They offer "aspect extraction" as the challenge of "identifying the aspect in a given opinionated text." Poria et al. (2016) further categorize aspects as either "explicitly" or "implicitly" expressed. In a related work, the authors in Kontopoulos et al. (2013) propose an ontology-based approach to sentiment classification. They argue that text-based sentiment classifiers are often inefficient because tweets typically do not consist of representative and syntactically consistent words, and so posts should receive a sentiment grade for each distinct notion expressed rather than an overall sentiment score. They use the example of a Smartphone for which sentiment can be measured across attributes such as brand name, operating system, and hardware components.

Similar to Kontopoulos et al. (2013), we recognize the limited application of an overall sentiment score. We aim to provide a means for more practical application of our lexicon set and interpretation of our results. Our use of the "aspect" concept differs from its traditional usage. Rather than applying this concept to an entity or a document, we apply it to the whole Twitter corpus. Our intent is to further reduce sparsity in our model and provide users with further insight into the underlying cause of the sentiment expressed in any given tweet. We posit that this is possible because tweets are very short. Therefore, the vast majority of tweets express a single notion and can be categorized based on the language used. We confirm this through the low number of features present per tweet (2.74) and discussion with our evaluators.

Table 3
Aspect categories & descriptions.

Aspect category	Description
Desire	Tweets often express desire for a product or service by simply stating their desire for an item (e.g. "I want Starbucks now!") or by expressing excitement for something related to the brand such as a promotional event or anticipation of something like a seasonal item (e.g. pumpkin spice latte from Starbucks).
Interjections	These features are like emojis or emoticons but are very short text phrases or utterances, e.g. "smh", "shaking my head", "mmm", "yum", "wtf", "f u."
Quality	This could technically be included in review, but we decided to pull it out since it can be applied to a review or transaction and sometimes describes something other than just a level of good or bad, e.g., "good", "bad", "great", "illogical", "unethical."
Review	Features from tweets expressing a review of some experience with a brand, e.g., "make my day", "consider this", "neglected", "hurt", "appalled."
Transactional	Similar to features in the review aspect but these are specific to transactions, e.g., "cheap", "delay", "out of stock", "hassle free", "deliver."
Domain Specific	Features found to be domain specific, e.g., "psl", "tobeapartner", "gunsense", "momsdemand", "gold level."
Ancillary	Supplementary features such as isHttp, base forms of negations, e.g., (can't, don't, won't, etc.) and emoticons and emojis.

We then work with our evaluators to identify general categories of notions expressed in our datasets. The result is two main categories in which a user either expressed a desire for something or provided some suggestion or criticism (review) related to an experience with the represented company or figure. The review category is very broad and we investigate the possibility of further decomposition. We find a distinction between tweets falling into the review category based on the type of account. For example, @Starbucks, @SouthwestAir and @VerizonWireless users commonly engage in transactions with these brands and report on their experience. @JustinBieber and @GovChristie represent public figures and users do not primarily engage in transactions with them. However, there are exceptions. Justin Bieber is an entertainer, and he produces music which he sells and performs at concerts. Users engage in transactions with him regarding these topics and provide feedback. We choose to split the "review" aspect into a general "review" aspect and one for transactions (the "transactional" aspect). Although not all twitter feeds may leverage the "transactional" aspect well, we are interested in brand management and expect most datasets collected to engage in transactions with their user base. Therefore, we classify the "transactional" aspect as generic. After further discussion with our evaluators, the "quality" aspect is abstracted from our review related aspects because it can be generically applied and because quality cannot be fully qualified by terms indicating some level of good or bad.

Sometimes tweets contain, or are made up of, short text phrases or utterances (interjections), where we can identify the position of the user, but cannot identify the underlying cause. Similarly, sometimes tweets contain only supplementary features from which evaluators can only identify the overall sentiment of the tweet but again do not have any information in respect to the underlying cause. We opt to bin the related features into the "interjections" and "ancillary" aspects. Lastly, we designate a "domain specific" aspect for modularization.

We present a set of seven aspects that describe the various ways users may express sentiment regarding a @subject. These aspect categories are labeled as: desire, interjections, quality, review, transactional, domain-specific, and ancillary. These terms were selected based on patterns noticed by evaluators and are described below (Table 3):

In our implementation of the aspect concepts, we map each of the lexicon features to an aspect category. This transformation results in the collapsing of the final representations of our feature set to only seven dimensions. Most applications of aspect concepts simply assign a sentiment score to each aspect. We, however, differ in our sentiment scoring process. Rather than simply assigning a sentiment score to each aspect, we compute an aggregate score reflecting the sum of the sentiment scores of all features belonging to that aspect. This sum reflects the collective sentiment of features belonging to this aspect. In our definition, an aspect can be thought of as a *"meta feature"* representing a class of features referring to a notion. The premise is that a feature's effect on a tweet's sentiment is accounted for through its corresponding aspect, and the semantic relationship that aspect holds with the sentiment class of the tweet. Therefore, the information content of an individual feature and its sentiment is carried through the *"meta feature,"* as represented by the seven aspects. The major benefits of this approach are dimensionality reduction (reducing many features into seven aspect), significant reduction of sparsity in the feature matrix, and reduction of noise introduced to the machine learning model. For example, from the Starbucks tweets, the word grams "very poor" and "hard" are mapped to the quality aspect category, "customer service" is mapped to the transactional aspect category, and "Pumpkin Spice Latte", "Christmas Blend", and "Passion Tea" are mapped to the domain-specific aspect category. Six aspects were defined generically so that they can be applied for TSA across all subjects (Twitter handles) or brands, regardless of the domain of analysis: desire, review, transactional, and quality aspects. The other two aspects were created to handle supplementary items such as negations (the ancillary aspect), or to capture specific emotions (the interjection aspect). Finally, the last aspect presents domain specificity (the domain specific aspect).

Another benefit of this approach is that the model contains metadata which can be leveraged for tasks such as measuring sentiment across an aspect or identifying tweets related to a particular aspect. For example, we can now easily retrieve all tweets related to demand or desire for a brand by searching for tweets containing the related features. This tweet set can be further evaluated to assess overall demand for products and services and what specifically is being demanded. Similarly, the review aspect can be leveraged to better understand customer feedback. If location data is also available from a tweet, it is possible to pinpoint the source of positive or negative feedback and react accordingly.

These aspects, collectively, are used throughout for modeling our various domains. They consistently produce accurate TSA values, which allows us to introduce two feature sets: the *"Twitter Generic Feature Set,"* which are domain agnostic, and the *"Twitter Domain Specific Features,"* which needs to be created for each specific domain. Application of this strategy across multiple diverse domains, has resulted in the generation of a Twitter Generic Feature Set that accounts for roughly 97% of features used in all models, with the domain specific features constituting only the remaining 3% of the feature set. Clearly, the reuse of the Twitter Generic Feature Set can positively contribute to the application of TSA across many domains. This research finding, while not completely resolving the domain transfer problem, does advance the solution to a high degree by allowing researchers to reuse the Twitter Generic Feature Set and identify a minimal set of domain specific features for creating their TSA model.

Table 4
Feature elements of the aspect categories.

Aspect	Features contained
Desire (16 features)	addicted, dislike, don't_want, like, need, never_again, intend_to_visit, really_want, want, hope_for, pray, preferred, regularly, times_a_day, waiting (in anticipation), wish
Quality (34 features)	bad, bad_taste, best, cool, damn, dangerous, delicious, enjoy, fake, good, great, gross, horrible, illogical, inefficient, looks_good, luck, nice, outdated, overrated, professional, refreshing, rude, safe, smart, smooth, special, sweet, talented, unacceptable, unethical, unpleasant, unsightly, worst
Review (98 features)	abuse, angry, apology, appalled, applaud, appreciate, ashamed, asshole, award, baby, belief, bitch, boring, cheer, chillin, completes_me, concern, congrats, consider, courage, coward, crime, death, deceived, digging, disappointing, disapprove, discrimination, dream, embarrass, evil, excessive, excited, favorite, feel, feel_sick, fix_this, friend, frustrated, funny, furious, glad, happy, hate, haters, heart, helpful, hero, highlight, hits_the_spot, hurt, idiot, impressed, inaccurate, inspire, irony, jealous, killing_me, kill_for,, kiss, loser, love, loyal, makes_my_day, marry_me, memories, miss, my_fix, neglected, not_worth_it, party, peace, perks, pleased, proud, redeemed, relax, respect, rumor, sad, saves_me, screwed_over, selfish, sex, smile, success, support, thanks, trend, truth, is_the_truth unfair, unreliable, vacation, violate, waiting (in delay), winner, worth_it
Transaction (24 features)	broke, cheap, delayed, deliver, earned, employee_issues, fuck_up, hassle_free, help, hook_up, is_back, out_of_stock, overcharged, overpriced, oversold, problem, refund, refuse, scam, screwed_up, slow, treat, unusable, wrong
Interjection (15 features)	bravo, enough_said, fml, fuck_you, hell_no, hi, kudos, little_things, lol, please, seriously, woohoo, wtf, yuck, yum
Ancillary (17 features)	but, cant, didn't, doesnt, dont, follow, hasnt, isHttp, isnt, not, please, unfollow, wasnt, wont, wouldnt, happy_emoticon, sad_emoticon
Domain Specific	(@starucks): got_my_starbucks, gunsense, momsdemand, gold_level, psl, tobeapartner, white_girl (7 features) (@govchristie): crime, fat, pig, political, stts, jersey_strong, vote_against, vote_for (8 features) (@verizonwireless): can_you_hear_me_now, cancel_contract, data, human, nsa, switching_to (6 features) (@southwestair): cattle, lost_bag, stranded, avgeek, swa (5 features)

Table 5
Data collection properties per dataset.

Domain	No. of tweets	Date collected
Starbucks	442,443	8/1/2013 to 11/4/2013
Gov. Christie	201,821	8/1/2013 to 11/4/2013
SW Airline	46,888	8/1/2013 to 11/4/2013
Verizon	52,741	8/1/2013 to 11/4/2013
Justin Bieber[a]	10,345,184	5/6/2011 to 6/8/2011

[a] From Ghiassi et al. (2013).

We assign each of our features to an aspect category, the same way that we associate terms and n-grams using equivalence partitioning. Feature matrix space is reduced to seven columns, with entries containing the sum of the polarity values of all present features for a given aspect. This produces a much smaller and denser input matrix for our experiments. The assignment of features to aspects is presented in Table 4.

4. Data collection and preparation

The first step in TSA is data collection (Fig. 1, steps 1 and 2). We use five datasets to (a) develop the TSLS, and (b) to demonstrate and evaluate transferability. These datasets are selected to represent diversity across domains and collectively cover a wide range of applications. We use the Twitter API for our data collection. Table 5 presents the five datasets, the number of collected tweets, and the time of collection.

In this research, we collected four new datasets: Starbucks (@Starbucks), Governor Christie (@GovChristie), Southwest Airlines (@SouthwestAir), and Verizon (@VerizonWireless). For each dataset, we collect a minimum of 40,000 tweets received over a 3 month period. Each account is selected to represent a new message domain. These datasets along with the @JustinBieber dataset (Ghiassi et al., 2013) represent message domains for a consumer product, a political figure, an entertainer, and two consumer services.

For each dataset, several groups of varying sizes are selected as candidate groups for our analysis. For @JustinBieber, tweets are partitioned based on the last two digits of the tweet ID. Our new datasets are smaller in comparison, and we use only the last digit of the tweet ID as our partitioning criterion (Fig. 1, step 2). The resulting tweet distribution is then evaluated across all groups (based on time of day and day of month). The goal is to find a candidate group that most closely resembles the distribution of tweets in the greater dataset. We next perform data cleansing by removing retweets and spam for each dataset (Fig. 1, step 3). The cleansed datasets are then partitioned into the training and testing datasets, for the training and testing of the sentiment engine in a later stage. Finally, these tweets are manually scored and classified (Fig. 1, step 4). Three Information Systems graduate student evaluators independently score tweets for sentiment on a five step positive to negative scale. The scale steps are "Strongly Positive", "Mildly positive", "Neutral", "Mildly Negative", and "Strongly Negative." Only sentiment evaluations that are unanimous are kept. Evaluators also collect the most meaningful n-grams in each tweet and remove obvious spam tweets. We continue to use the tools developed in Ghiassi et al. (2013) to facilitate this step. The tool for tweet evaluation is extended to capture n-grams up to size $n = 5$ and to display emoticons and emojis to the evaluators along with the text of the tweets.

To fully evaluate the performance of the machine learning models for each class, a large number of tweets for each category is required. However, for some domains, the number of tweets in the "Strongly Positive or Negative" buckets are found to be so thoroughly dominant that additional refinement to the final dataset is required to account for the imbalance between classes. This skewness of the distribution of tweet's sentiment is also observed by other researchers and was perceived to be only "slanted" toward positive. Lexalytics (2016) state that "We find that overall document content is slanted to be slightly positive by default." Ghiassi et al. (2013) show that the positive sentiment in their study to reach a high of 82.7% in total (strongly positive sentiment at 56% and mildly positive tweets at 26.7%). Similarly, Boucher and Osgood (1996) report that "On the average there are almost twice as many positive as negative words in our text." In this study, while confirming the positive slant for some cases, we find evidence of a negative slant for others. For instance, the Verizon dataset shows

the sentiment of the tweet corpus to be highly skewed toward the negative (71.22%) with mildly negative sentiment class at 44.12% and the strongly negative ones at 27.10%.

Many studies, especially for three-class sentiment analysis, choose balanced datasets for their analysis. We, however, address the existence of the imbalance, and experimentally determine a sufficient number of representative instances for proper evaluation. We find that requiring each category to have at minimum 500 instances in the final datasets is sufficient to preserve the overall distribution of the data (positive or negative) and provide sufficient representation for our models to train on. To reach this threshold, a supplementary set of negative/positive tweets are included. To acquire these additional instances, we extract another random set of tweets from the previously collected data, which is not already included in the dataset, to be labeled by our team of evaluators, and follow the same scoring criteria as outlined earlier. Therefore, for each dataset, the final, refined model is composed of a set of tweets with sentiment across all five categories plus any additional tweets needed to meet our minimum representation metric. These groups are used as the basis for our analysis.

4.1. Managing imbalanced datasets

To ensure an even and representative distribution between training and testing datasets, the tweets are ordered by sentiment (from strongly positive to strongly negative) and then proportionally divided between training and testing datasets (e.g. 1st row to train, 2nd row to test, etc.). Assessing the distribution of sentiment classes for the datasets, we notice that for some datasets (@JustinBieber, @Starbucks and @GovChristie), the number of strongly positive tweets is very dominant and the number of negative tweets is relatively small; sentiment distribution is strongly slanted towards positive. We observe the reverse for @VerizonWireless and @SouthwestAir; that is, the negative tweets are the dominant class and the positive tweets are fewer in numbers. These observations are understandable since the tweets about a popular entertainer tend to run positively, whereas tweets to a service provider (e.g. Verizon) tend to be customer service related and therefore more likely to be negative in nature. Building models for each class requires acknowledgement and treatment of imbalance. Literature addressing this topic offer several solutions, including modified sampling methods (Piri, Delen, & Liu, 2017). One solution for addressing imbalance datasets is “over sampling.” We adopt this method and address the imbalanced datasets by compensating training and testing datasets for the models to reflect this disparity. We note two points in addressing the imbalanced datasets: (1) in training the less populated classes, we need not include the entire available out-of-class data. A sufficient representation of out-of-class data is enough to allow their recognition by our machine learning tools, and (2) we need to collect enough labeled data for the less dominant classes to allow for our machine learning tools to be successfully trained and tested. We comply with both of these points throughout our analysis. These characteristics are one reason for having different sized datasets for the five models for some of our domains.

Specifically, we address the imbalance in datasets by defining an approach that begins with requiring a sufficient number of “in-class” instances. This allows for a proper representation of “in-class” tweets, the number of “out-of-class” instances should not totally dominate the “in-class” tweets in a particular sentiment class. We, therefore, use the class distribution as a guideline for managing the number of “out-of-class” tweets in the dataset. For the unbalanced models, the training and testing datasets are scaled such that the number of in-class instances represents approximately no less than 30% of the instances in each set respectively. To avoid any bias, out-of-class records are scaled to maintain a distribution similar to the total dataset; out-of-class records are randomly added until the desired distribution is achieved. This is done to ensure sufficient representation of in-class instances so that the model can be satisfactorily trained and tested on a distribution of tweets which is similar to that identified in our candidate dataset. The sizes of the collected datasets are large enough that this selection still leaves us with sufficient data for both training and testing; the consistency of our results is a testimonial to validity of this approach.

4.2. Domain and dataset characterization and sentiment analysis results

We next present dataset properties for all database domains. The first database, Justin Bieber, was collected and introduced in our earlier work (Ghiassi et al., 2013). Here we begin by characterizing the Starbucks dataset and use lessons learned for characterizing the other domains. Similar to Aue and Gamon (2005), we use two datasets to generate and validate the TSLS and to assess its effectiveness. Although, Aue and Gamon (2005) used one additional dataset for evaluation we offer three new datasets from diverse domains to evaluate the transferability of the TSLS. We start by using the first two datasets to identify the TSLS and perform sentiment analysis using DAN2 and SVM. Once the TSLS is identified, we apply it to the three new datasets to evaluate its effectiveness with the two machine learning tools.

4.3. Data cleansing, preprocessing & input matrix creation

For all databases in this research, we start with dataset cleansing (retweets and spam removal) and then begin to perform data preprocessing (Fig. 1, step 3). Since the collected tweets often include “spam” and “retweets”, the raw tweets need to first be cleansed. Retweets are automatically identified through keyword search of “RT” and its variants. To search for spams, we divide the tweets into groups based on the last digit of tweet ID. This is done in order to select random groups of tweets that span the entire investigation period and to limit the influence of “spam” tweets. A “spammer” typically sends the same tweet as many as 100 times in a row. Thus, an algorithm that only takes one row out of every 100 sequential tweets will reduce the impact of a “spammer” (to that of a normal follower). Therefore, selection of tweets, based on tweet ID, will reduce the presence of spam to a minimum. We take further measures for spam removal by identifying common phrases, such as “enter to win,” and remove any tweets containing these phrases. Complete spam cleansing is beyond the scope of this research. However, our experience with our selection approach followed by additional manual spam removal results in reasonably cleansed datasets.

For the data preprocessing phase, we follow some techniques suggested by Agarwal et al. (2011), and transform complex values into placeholder variants so that they can be parsed using simplified queries. We perform the following:

- URLs are replaced with [http].
- Single exclamations are replaced with [s-exclm].
- Multiple exclamations are replaced with [m-exclm].
- Quoted retweets are replaced by [q-rt].
- Text emoticon characters are also replaced by [p-emoticon] and [n-emoticon].
- Unicode characters (emoji) are first converted to their code point or surrogate pair values. These values are parsed against our collected emoji list and then replaced by either [p-emoticon] or [n-emoticon] respectively.

After preprocessing, tweets are parsed for the presence of features. As part of the feature engineering process, our features are

given a tag, described earlier, which maps to a polarity score. Similar to Ghiassi et al. (2013), we use these values to produce our starting point values for our experiments. When a feature is identified, its polarity score is added to the sum for its corresponding aspect. Thus, each entry in the input matrix is a vector representation which contains a sum of the polarity scores for each of the seven aspects for a given tweet. Finally, a label column is added for each entry to show a tweet's classification into one of the four sentiment classes. This results in an input matrix spanning seven columns and the corresponding label.

4.4. Additional modeling considerations for Twitter sentiment analysis: sentiment scale

We continue to use the sentiment scale developed by Ghiassi et al. (2013). The primary objective of developing this sentiment scale was to create a scale that will enable brand marketers to identify extreme statements regarding his or her product as well as mildly expressed sentiments. It is expected that the extremely positive messages would be promoted by the brand, and the extremely negative messages would be referred to customer service to resolve. Creation of the mildly expressed sentiment, while difficult to assess, are of paramount value to brand management. It is the tweets in these two classes that provide actionable opportunities for product managers to influence customer opinions and to refine product features to better match customers' needs.

In our scale, we represent sentiment with a set of labels. We chose to use labels that traverse a linear scale (negative to positive) as opposed to labels that categorizes tweets into multiple moods. We find that a "positive to negative" scale is less hampered by interpretation bias. This scale uses five steps {2, 1, 0, −1, −2} to categorize sentiment. The definition of each step is as follows:

- 2 = author clearly loves the brand (Strongly Positive).
- 1 = author likes the brand (Mildly Positive).
- 0 = unclear how author feels about the brand (Neutral).
- -1 = author dislikes the brand (Mildly Negative).
- -2 = author clearly hates the brand (Strongly Negative).

The simplicity of this scale is of great benefit during the manual scoring process. Examples of tweets belonging to each sentiment class are given in the Table A.3 in the Appendix.

5. Experiments and results

Two machine learning tools, DAN2 and SVM, are used for Twitter sentiment analysis. The input to these tools is a collection of vector representations of tweets from the training dataset along with their manually determined class label. DAN2 and SVM use the "one vs. all" approach for classification. This approach necessitates the development of four separate models; one for each sentiment class aside from neutral. Each model is trained with its corresponding training dataset and is evaluated using its corresponding testing dataset. We have selected two machine learning tools to ensure that application of the TSLS is not biased or "tuned" toward a specific tool. SVM is selected because it is the most widely available and used tool. The SVM models are prepared using Thorsten Joachims' SVM Light application (Joachims, 1999). Experiments are performed using linear, polynomial, and PUK kernels, and the best results are reported for all models. DAN2 is selected since it has been shown to be very accurate, even for imbalanced datasets. In this research, we do not intend to advocate, the efficacy of either of these machine learning engines.

We present results for four new datasets to show (a) the effectiveness of TSLS, and (b) the transferability of TSLS, and (c) using the five class categorization approach.

Table 6
TGFS and TDSF values.

Dataset	TGFS no. (%)	TDSF no. (%)
Justin Bieber	181 (97%)	6 (3%)
Starbucks	204 (97%)	7 (3%)
Gov. Christie	204 (96%)	8 (4%)
Southwest Airlines	204 (98%)	5 (2%)
Verizon Wireless	204 (97%)	6 (3%)

5.1. Starbucks dataset

5.1.1. Starbucks– training & testing data preparation

For the Starbucks dataset, the 442,443 collected tweets are divided into candidate sets based on the last digit of the tweet ID. Retweets are removed and spam is identified and removed as well. After this step, 254,196 tweets remain. Ten candidate groups are selected with sizes ranging from 9365 to 41,508. Tweet distribution is then evaluated across the groups and a group of 9367 tweets with IDs ending in 5 was selected (a random selection).

A team of three judges is used for scoring and they provide 4905 labeled tweets for analysis. This results in 2861 strongly positive tweets, 560 mildly positive tweets, 798 mildly negative tweets, and 686 strongly negative tweets. These tweets are randomly distributed into training (∼80%) and testing (∼20%) datasets (Table A.4, Appendix).

We use a "one vs. all" approach for classification and build four models representing each step of our sentiment scale, except neutral. The order of experiments is as follows: strongly positive, strongly negative, mildly positive, and mildly negative. Experiments are ordered by estimated difficulty. As stated earlier, when developing multiple models for TSA, we need to address the challenge of dealing with imbalanced datasets. The difference in total number of tweets for each model reflects the process described earlier to address existence of the unbalanced distribution of the sentiment classes. The processing order of the models, therefore, is selected to allow us to address this challenge.

Applying the feature engineering process to the Starbucks dataset results in identification of 204 TGFS (97%) and 7 TDSF (3%) for a total of 211 features. Table 6 presents TGFS and TDSF values for all five datasets used in this research. Table 6 shows that application of the TSLS to the five diverse datasets has resulted in 97% feature reusability (domain transferability) and only between 5 to 8 domain specific features (3%). This finding should encourage researchers to use the TGFS and only need to identify a hand full of TDSF for their Twitter sentiment analysis models.

5.1.2. Starbucks – sentiment analysis results

The input to DAN2 and SVM is the vector representation of the tweets from the feature matrix, their manually classified sentiment value, and their starting point value (for DAN2). Ghiassi et al. (2013) used a linear equation to compute a polarity score for each tweet as the starting point values. In this research, since the introduction of polarity sums to our sentiment classification, we revise the starting point equation described in Ghiassi et al. (2013) to be based on the sum of the aspect scores for a given tweet. This value is mapped to a sentiment class. Our results for using the TSLS with DAN2 and SVM are presented in Tables 7 and 8, respectively.

As previously mentioned, model distribution is scaled so that in-class instances represent at least 30% of the total instances and out-of-class instances maintain a distribution similar to the greater dataset distribution.

For the testing results, in terms of recall, DAN2's performance is very good (recall values around 85%) except for the mildly positive case. SVM, on the other hand, produces good results for the

Table 7
DAN2 results for @Starbucks.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	89.96	89.76	89.96	86.76	85.96	87.59
Mildly positive	78.76	83.07	75.06	71.84	77.89	66.67
Mildly negative	78.07	81.11	75.24	82.97	79.29	87.01
Strongly negative	85.91	92.28	80.36	88.37	91.94	85.07

Table 8
SVM results for @Starbucks.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	90.00	93.80	91.70	84.40	83.80	85.00
Mildly positive	83.20	84.40	82.00	69.70	77.80	63.10
Mildly negative	85.20	84.40	86.00	75.50	77.00	74.00
Strongly negative	88.20	96.40	81.20	69.10	93.00	55.00

Table 9
DAN2 results for @GovChristie.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	91.17	84.95	92.22	89.17	82.14	89.84
Mildly positive	80.51	76.55	60.00	84.89	81.58	68.89
Mildly negative	83.94	84.50	83.73	82.04	81.32	85.06
Strongly negative	88.65	83.94	82.88	86.21	78.38	78.02

Table 10
SVM results for @GovChristie.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	91.90	90.90	93.00	76.40	84.80	69.50
Mildly positive	78.80	88.00	71.40	64.10	75.80	55.60
Mildly negative	87.70	86.30	89.20	78.30	82.30	74.70
Strongly negative	91.30	94.50	88.30	78.50	75.00	82.40

strongly positive case (85%), poor results for the strongly negative (55%), and average results for the other two classes. In terms of precision, DAN2 and SVM both perform well.

When comparing DAN2 and SVM using the more balanced F_1 measure, defined as $F_1 = 2^*P^*R/(P+R)$, where P and R are the precision and recall values, respectively (Manning et al., 2008), the same relative performance is observed. DAN2's values range from 72% to 88% while SVM values range is from 69% to 84%.

Finally, we note that the balanced values of training and testing results are an indication of a lack of over-fitting of all models. We use such balanced values to ensure that none of our models suffer over-fitting.

5.2. Gov. Christie dataset

5.2.1. Gov. Christie – dataset specific properties

The @GovChristie dataset represents a new public figure domain for this research. We find that although both Justin Bieber and Gov. Christie are public figures, for @GovChristie, tweet content is noticeably different between these datasets. For the Governor, tweet content is primarily political in nature, and we posit that tweeter demographics differ significantly from those messaging @JustinBieber. As such, the way in which users interact with this account is different, thus representing a distinct domain.

Using this domain allows us to begin to address and to demonstrate domain transferability of the TSLS. We start with reusing the 204 features from the Twitter Generic Feature set (TGFS) constructed from the previous two domains (Justin Bieber and Starbucks). We then identify and use the Twitter Domain Specific Features (TDSF) for Governor Christie (8 features) and combine the two sets to create 212 features for this dataset. In this section, we demonstrate that using this set for TSA for this domain also results in similarly excellent accuracy values; thus, validating the domain transferability of the TSLS for this dataset.

At this point, a significant number of tweets have been parsed over the lifetime of the Twitter Specific Lexicon Set. The Twitter Generic Feature Set (204 features) resulting from the analysis on the @Starbucks dataset covers 92.37% of the tweets in the @GovChristie dataset. This signifies that we have collected most generic n-grams and their variants. We then identify 8 features that are specific to this domain (TDSF) and bin new variations of existing features captured by evaluators into their appropriate equivalence classes. Thus, the total number of feature used is 212 (204 TGFS + 8 TDSF) which allowed us to reach our coverage goal of 95%.

5.2.2. Gov. Christie – training & testing data preparation

For the @GovChristie dataset, the 201,821 collected tweets (Table 5) are divided into candidate sets based on the last digit of the tweet ID. Retweets and spam are removed and 110,285 tweets remain. Ten candidate groups are selected with sizes ranging from 6660 to 9912. Tweet distribution is then evaluated across the groups and a group of 4302 tweets with IDs ending in 4 was selected. We aim to score at least 2000 tweets, and the subset of candidate tweets that were manually scored results in a set of 2222 tweets. Because our working dataset is smaller for @GovChristie, we relax our previous requirements and only require that each class hold at least 200 instances and that each feature occurs at least 3 times. We are left with 1756 tweets for building our models. This results in 642 strongly positive tweets, 235 mildly positive tweets, 386 mildly negative tweets, and 493 strongly negative tweets (Table A.5, Appendix). We then follow the same process as the Starbucks dataset for testing. We build a model for each sentiment class except neutral. We again require that in-class instances represent at least 30% of total data in each model. The distributions of tweets in our models are shown in Table A.5 (Appendix).

5.2.3. Gov. Christie – sentiment analysis results

The results for the DAN2 and SVM models are presented in the Tables 9 and 10, respectively.

The testing results of this dataset show that in terms of recall, DAN2's performance is very good (between 78% and 90%) except for the mildly positive case (recall value of 69%). SVM produces its best results for the strongly negative case (82%). For both tools, the weakest results come from the mildly positive case, which suggests that there is large overlap between strongly positive and mildly positive instances.

We note that the characteristics of political tweets cause difficulty in evaluation using our sentiment scale. In general, political tweets express the author's agreement or disagreement on some issue. This does not typically translate well into "like" and "love" on our sentiment scale. Our evaluators decided to evaluate these tweets by estimating the likelihood that the author would or would not vote for Governor Christie in an election. If the tweet suggested that the sentiment expressed from a tweet was focused on a political issue rather than the Governor himself, it was scored as mildly positive or negative, as the author's support is tied to the Governor's actions and views on said issue. The Governor also receives non-political tweets, and the mixture of these tweet types further complicates evaluation.

In terms of precision, both tools produce a similar range of values. Comparing F_1, DAN2 outperforms SVM in all cases.

5.3. Southwest Airlines dataset

5.3.1. Southwest Airlines – dataset specific properties

The @SouthwestAir dataset represents a business service in the transportation domain. In this dataset, we now encounter tweets which contain words from our lexicon but do not contain sentiment towards @SouthwestAir and are not filterable. These tweets are informational in nature but contain sentiment words. A common offender is the word "delay." Some examples are included below:

@SouthwestAir If I miss my connecting flight due to a delay on my 1st flight. I wont be penalized right?

My @SouthwestAir flight home from Seattle is delayed. Do I try my luck with an earlier flight?

Drats. @SouthwestAir has a flight from Tulsa -> STL on Nov. 18 at 5:15pm. I get out of class at 4:50. I want to see soccer at Busch.

We find that the Southwest Airlines and Verizon Wireless (presented next) datasets offer a different type of domain, in which the overall distribution of tweet sentiments tends to be more negative than positive. We acknowledge this property and adjust formation of our training and testing datasets accordingly. The majority of the tweets to @SouthwestAir are complaints. These tweets tend to fall in the mildly negative class which dominates this set. This may be indicative of the way in which users choose to interact with some business service brands on Twitter.

5.3.2. Southwest Airlines – training & testing set preparation

For the @SouthwestAir dataset, the 46,888 collected tweets are divided into candidate sets based on the last digit of the tweet ID. After removing retweets and spam, 39,066 tweets remain. Ten candidate groups are selected with sizes ranging from 6026 to 8859. Tweet distribution is then evaluated across the groups and a group of 5599 tweets with IDs ending in 2 was selected (a random selection). Evaluators classify each instance and collect n-grams to be used later in feature engineering, which results in 1759 scored tweets.

We use the same data requirements as @GovChristie for this dataset; we require that each class hold at least 200 instances and that each feature occurs at least 3 times. However, due to the much smaller working dataset, we allow the number of strongly negative tweets to be 184. We are left with 1589 tweets for building our models. This results in 493 strongly positive tweets, 342 mildly positive tweets, 24 no sentiment tweets, 546 mildly negative tweets, and 184 strongly negative tweets (Table A.6, Appendix).

We find that for this dataset, a model using all available tweets is suitable for all experiments for all models; the 1589 tweets are divided into 1269 tweets for training and 320 tweets for testing.

We note that the strongly negative class holds only 11.58% of total instances. However, tweets falling into this class tend to use distinct features and are easily identifiable. This is shown by the performance metrics collected from previous datasets; we find that classifier performance is still satisfactory for the Very Negative Model, despite being unbalanced in this case. The distributions of tweets for all models is shown in Table A.6, (Appendix).

We use the same approach for feature additions as in the @GovChristie dataset. However, after adding synonymous terms and n-grams and domain specific n-grams collected by evaluators, we reach a coverage metric of 91%. Due to the smaller dataset size, we relax our coverage metric slightly from 95% to 91% and proceed with testing. The TSLS for the Southwest dataset is comprised of 209 features (the same 204 generic features and 5 domain specific features). We note that although we encounter a pronounced difference in sentiment distribution, our lexicon set maintains very high coverage and that this is a strong evidence of transferability of the TSLS.

Table 11
DAN2 results for @SouthwestAir.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	84.17	81.47	87.06	86.27	83.81	88.89
Mildly positive	68.92	63.95	74.73	69.74	63.86	76.81
Mildly negative	71.32	77.09	83.73	78.79	83.87	74.29
Strongly negative	81.27	83.94	78.77	77.33	78.38	76.32

Table 12
SVM results for @SouthwestAir.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	86.30	85.50	87.10	75.60	84.00	68.70
Mildly positive	71.00	65.40	77.70	66.20	62.00	71.00
Mildly negative	71.70	84.90	62.00	63.90	86.90	50.50
Strongly negative	83.80	88.50	79.50	51.90	87.50	36.80

5.3.3. Southwest Airlines – sentiment analysis results

For this dataset, a single model was adequate for all experiments. This model includes all available tweets and did not require scaling. Our results are presented in the Tables 11 and 12, respectively. Overall TSA performance values continue to be excellent, indicating that lowering the coverage metric from 95% to 91% for this dataset, did not have significant impact on the model performance.

Recall values for DAN2 average to 79.08%. SVM performs significantly worse with an average recall value of 56.75%. The contingency tables for the mildly negative and strongly negative models show that SVM misclassifies a significant number of in-class instances. The precision values shown do not reflect this fact because in-classes instances represent the minority of total instances in a model; the precision values given for SVM are inflated due to better performance in classifying out-of-class instances. This may suggest that "mildly negative" and "strongly negative" instances in this dataset are characterized by the presence of multiple features rather than a dominating single feature, which we commonly saw previously.

Since DAN2 continues to perform well with the TSLS, we attribute low performance of SVM to its functionality with the smaller dataset rather than the choice of feature set.

5.4. Verizon Wireless dataset

5.4.1. Verizon Wireless – dataset specific properties

The @VerizonWireless dataset represents a business service in the telecommunications domain. Tweet distribution in this dataset is negatively skewed, much more so than that found in the @SouthwestAir dataset. Once again, the Mildly Negative class dominates the dataset; it represents nearly half of the dataset (44.12%, 923 tweets). The Strongly Negative class is the second most dominant, representing 27.10% of total tweets (567 tweets). Positive tweets represent only 27.91% of tweets, and we again find tweets which contain no sentiment but contain features from our lexicon set that represent the remaining tweets (0.87%) in the dataset. Some examples are below:

Shall I make the switch? #verizon #att #confused (at @VerizonWireless) http://t.co/Sqq0JoFV0W

Table 13
DAN2 results for @VerizonWireless.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	89.78	90.06	89.95	93.98	92.86	95.12
Mildly positive	86.76	89.93	81.97	89.96	83.12	84.21
Mildly negative	87.08	87.86	86.31	86.19	88.14	84.32
Strongly negative	88.30	88.50	88.11	94.22	94.64	93.81

Table 14
SVM results for @VerizonWireless.

	Training (%)			Testing (%)		
Model	F1	Precision	Recall	F1	Precision	Recall
Strongly positive	92.70	96.00	89.50	78.90	93.30	68.30
Mildly positive	87.80	90.90	84.90	83.40	84.00	82.90
Mildly negative	89.00	90.00	87.90	82.00	88.70	76.20
Strongly negative	92.60	93.30	91.90	91.40	93.50	89.40

@VerizonWireless will this one support simultaneous voice and data unlike the 5?
@VerizonWireless would I be able to get on the edge program even if it's not time for my upgrade

These tweets tend to be informational in nature and contain generic terms which take a domain specific meaning in the context of tech devices. The above examples use the terms "support" and "upgrade" for instance. "Support" in this context means having the ability to do something rather than giving or receiving assistance. Similarly, the term "upgrade" refers to the next generation of a device and not necessarily an improvement in quality.

We note that similar to the @SouthwestAir dataset, the majority of tweets received are again complaints. From these two datasets, we observe that users use Twitter as a platform for customer engagement. This may represent a business opportunity for service providers that maintain a Twitter account.

5.4.2. Verizon Wireless – training & testing set preparation

For the @VerizonWireless dataset, the 52,741 collected tweets (Table 5) are divided into candidate sets based on the last digit of the tweet ID (a random choice). Retweets and spam are removed, and 40,235 tweets remain. Ten candidate groups are selected with sizes ranging from 9540 to 14,778. Tweet distribution is then evaluated across the groups and a group of 9597 tweets with IDs ending in 0 was randomly selected. A subset of 2500 candidate tweets are manually scored for this analysis.

We then check the tweets for sufficient representation of features and of instances by class. We use the same data requirements as @GovChristie for this dataset; we require that each class hold at least 200 instances and that each feature occurs at least 3 times. We are left with 2092 tweets for building our models. This results in 203 strongly positive tweets, 381 mildly positive tweets, 18 no sentiment tweets, 923 mildly negative tweets, and 567 strongly negative tweets (Table A.7, Appendix).

We use the same approach for feature additions as in the @GovChristie and @SouthwestAir datasets. After adding synonymous terms and n-grams and new domain specific terms collected by evaluators, we reach our coverage metric of 95%. The TSLS for the Verizon Wireless dataset is comprised of 210 features (the same 204 generic features and 6 domain specific features).

5.4.3. Verizon Wireless – sentiment analysis results

Following the same process as before the results for the DAN2 and SVM models are presented in the Tables 13 and 14, respectively.

DAN2 exhibits excellent recall values for this dataset with testing recall values ranging from 84% to 95%. Examining SVM recall values, overall, we see better performance than most cases for previous datasets. SVM performs reasonably well for the mildly positive and strongly negative cases, with testing recall values of 83% and 89%. However, the strongly positive and mildly negative recall values are low 68% and 76% respectively. Unlike other datasets, we see SVM produce lower than expected recall values for the strongly positive class (testing recall value of 68%). This may suggest that tweets falling into this class are less distinct for this dataset. We note that the strongly positive class is also the least represented class.

Looking at precision values, both tools produce strong results. We note that SVM reports the best precision values for the strongly positive and strongly negative cases. This supports our observation that SVM performs well only when data points are distinct. Finally, both tools produce strong F_1 values except SVM for the very positive case.

5.5. Evidence of domain transferability

Reviewing results for all experiments, we have applied the TSLS to five datasets. These datasets represent distinctly different subject domains and are also characterized by tweet distributions representing varying degrees of both positive and negative skew. A simple averaging of the results for all datasets, using the F_1 value (from testing), yields 84% accuracy for DAN2 and 75% accuracy for SVM. Both values are better than what is generally reported in the literature, especially for five class sentiment models.

The domains of Starbucks and Justin Bieber were found to be diverse, but we postulated that the union of their Twitter Generic Feature Sets (TGFS) may offer a domain transferable feature set. To assess this assumption, we applied the TGFS to three new datasets: Governor Christie, Southwest Airlines, and Verizon. For all three new datasets, the TGFS satisfied the 95% coverage metric in two cases and reached 91% in the remaining dataset. To perform TSA on the three datasets, we needed only to identify the TDSF for each dataset and capture any new variations of existing features.

The results show this approach to be highly effective as measured by various accuracy metrics. Given the domain variety represented by these datasets, we have shown the effectiveness of the TSLS and have evaluated and validated its domain transferability across these five datasets. Strong, consistent results from both SVM and DAN2, with DAN2 performing consistently better, suggest their strength for use in TSA.

5.6. Twitter specific lexicon set vs. Other feature sets

Dictionary based lexicon sets offer transferability across domains but they often lack sufficient accuracy. The TSLS offers both; transferability while still producing excellent accuracy values. We have selected the MPQA Subjectivity Lexicon and SentiWordNet Ensuli & Sebastiani (2006) lexicon sets as two alternatives for this analysis for comparison against the TSLS. These lexicon sets are shown to be very effective in two studies (Musto, Semeraro, & Polignano, 2014; Taboada et al., 2011). The MPQA Subjectivity Lexicon contains 6886 terms and is part of the OpinionFinder System. SentiWordNet is a collection of 147,306 terms which have been organized into groups of synonyms and assigned positive and negative weightings. To offer an unbiased comparison, we select SVM as the machine learning tool for these experiments. The choice of SVM is based on its wide availability and its extensive application in existing literature.

The effectiveness of an earlier version of the TSLS vs. a traditional feature set was studied in Ghiassi et al. (2013). The analysis was based on the TSLS defined for the Justin Bieber dataset

Table 15a
Lexicon comparison.

		@Starbucks						@GovChristie					
		Training			Testing			Training			Testing		
Lexicon	Model	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
MPQA	Str. pos.	86.4	80.0	93.9	78.1	70.2	88.1	83.7	92.7	76.3	60.4	70.1	53.1
	Mildly pos.	70.8	89.0	58.8	42.9	57.6	34.2	84.5	94.0	76.8	42.5	51.5	36.2
	Mildly neg.	76.5	89.3	66.9	49.8	58.3	43.5	92.4	91.9	92.8	63.3	61.3	65.5
	Str. neg.	81.7	92.8	72.9	58.5	70.5	50.0	90.8	88.8	92.9	62.6	58.7	67.0
SWN	Str. pos.	97.9	96.7	99.2	80.0	77.0	83.3	99.5	99.8	99.2	64.2	61.4	67.2
	Mildly pos.	99.1	99.3	98.8	42.0	41.6	42.3	100	100	100	31.6	31.3	31.9
	Mildly neg.	98.9	100	97.9	54.7	57.0	52.6	99.5	99.7	99.4	69.0	70.2	67.8
	Str. neg.	98.8	99.8	97.8	69.7	69.9	69.4	99.6	99.5	99.7	70.8	72.4	69.2
TSLS	Str. pos.	90.0	93.8	91.7	84.4	83.8	85.0	91.9	90.9	93.0	76.4	84.8	69.5
	Mildly pos.	83.2	84.4	82.0	69.7	77.8	63.1	78.8	88.0	71.4	64.1	75.8	55.6
	Mildly neg.	85.2	84.4	86.0	75.5	77.0	74.0	87.7	86.3	89.2	78.3	82.3	74.7
	Str. neg.	88.2	96.4	81.2	69.1	93.0	55.0	91.3	94.5	88.3	78.5	75.0	82.4

Table 15b
Lexicon comparison.

		@SouthwestAir						@VerizonWireless					
		Training			Testing			Training			Testing		
Lexicon	Model	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
MPQA	Str. pos.	73.9	92.0	61.7	43.7	63.5	33.3	75.5	97.1	61.7	55.9	70.4	46.3
	Mildly pos.	48.1	94.6	32.2	6.5	13.0	4.3	56.6	91.2	41.0	29.5	39.1	23.7
	Mildly neg.	72.4	90.7	60.2	38.4	52.4	30.3	73.8	85.7	64.8	45.9	52.8	40.5
	Str. neg.	71.3	97.6	56.2	24.6	36.8	18.4	83.9	93.1	50.2	44.6	69.8	32.7
SWN	Str. pos.	99.5	100	99.0	54.7	53.9	55.6	99.4	100	98.8	52.1	59.4	46.3
	Mildly pos.	99.1	100	98.2	34.0	33.3	34.8	99.0	99.3	98.7	61.3	57.5	65.8
	Mildly neg.	99.5	99.8	99.3	60.8	58.5	63.3	97.7	96.5	98.9	60.5	62.8	58.4
	Str. neg.	100	100	100	38.0	36.6	39.5	97.8	97.4	98.2	58.1	57.9	58.4
TSLS	Str. pos.	86.3	85.5	87.1	75.6	84.0	68.7	92.7	96.0	89.5	78.9	93.3	68.3
	Mildly pos.	71.0	65.4	77.7	66.2	62.0	71.0	87.8	90.9	84.9	83.4	84.0	82.9
	Mildly neg.	71.7	84.9	62.0	63.9	86.9	50.5	89.9	90.0	87.9	82.0	88.7	76.2
	Str. neg.	83.8	88.5	79.5	51.9	87.5	36.8	92.6	93.3	91.9	91.4	93.5	89.4

which included 187 features. To test the effectiveness of the TSLS, the authors select a commercially available TSA system using a traditional feature set based on bag-of-words, OpinionFinder (Wilson et al., 2005), for comparison,

The Justin Bieber and OpinionFinder lexicon sets were used as input for SVM. The authors in Ghiassi et al. (2013) show that the sentiment analysis based on the TSLS feature set produces better values for accuracy metrics.

In this research, we use the MPQA and SentiWordNet lexicon sets to evaluate sentiment across four models; one model for each sentiment class except neutral. The results are presented in Tables 15a and 15b. We compare these results with the same values for the TSLS for each of the four datasets. Tables 15a and 15b present the results using the three metrics: precision, recall, and F_1. We present our comparison using the F_1 metric, as it is more balanced, from the testing set results.

For the Starbucks dataset, the F_1 values for MPQA range from 43% to78%. The corresponding values for SentiWordNet range from 42% to 80%. The TSLS values range from 69% to 84% and the values are consistent throughout the four classes. Both MPQA and SentiWordNet perform very poorly for the mildly positive and negative classes. Similar behavior can be observed for the Governor Christie dataset. The F_1 values for MPQA range from 42% to 63% and for SentiWordNet from 32% to 71%. The TSLS values for this dataset range from 64% to 78% and are consistently better.

The performance of both MPQA and SentiWordNet becomes even less favorable for the next two datasets. For the Southwest Airlines dataset, the F_1 values for MPQA are very poor, ranging from 6% to 43%. The F_1 values for SentiWordNet are slightly better, ranging from 34% to 61%. The TSLS values range from 52% to 76% and are more consistent. The Verizon dataset results are better with the F_1 values for MPQA ranging from 29%% to 56%, and for SentiWordNet from 52% to 61%. The TSLS values are significantly better, ranging from 79% to 91%.

The overall performance of SVM using TSLS vs. MPQA and SentiWordNet lexicons shows that TSA using TSLS can yield more accurate results.

6. Conclusion

This research makes several contributions to Twitter sentiment analysis. We introduce an innovative approach to supervised feature reduction using n-grams and statistical analysis to develop a Twitter specific lexicon set for sentiment analysis, which results in a more accurate estimation of tweet sentiments.

We agree with authors in Oliveira et al. (2014) that, in general, the “domain independent lexicon” approach to sentiment analysis has not been the most accurate “approach,” even though studies have historically avoided the laborious task of document labeling. On the other hand, domain specific approaches fail to generalize beyond their chosen target. However, tweets, characterized by their short length, offer an opportunity to employ a transferable lexicon set across all domains with very good accuracy as presented in this research. The process introduced in this research uses a hierarchical approach that begins with gathering of large corpuses that are manually labeled to provide for large training and testing datasets. Next, our feature engineering approach results in a Twitter specific lexicon set, which is composed of a Twitter generic feature set and

small number of Twitter domain specific features; the TGFS constitutes 97% of the entire TSLS, and the TDSF only accounts for the remaining 3% of the TSLS. More importantly, the TGFS is shown to be domain transferable, allowing its reuse across diverse Twitter domains. The identification of the TGFS, allows researchers to adopt this feature set and to only identify a few domain specific features in order to perform Twitter sentiment analysis. Additionally, applying the feature engineering steps introduced in this research, results in a smaller Twitter lexicon feature set which reduces problem complexity (model size reduction), maintains a high degree of coverage over the Twitter corpus, reduces input matrix sparsity, and yields improved sentiment classification accuracy.

In identifying the TSLS and using a five point scale for classification, we determine that the neutral category accounts for no more than 10% of all tweets. This finding is in contrast to earlier research that used a three-class scale and classified more than 80% of all tweets collected as neutral (Go et al., 2009). Additionally, where some previous investigations have limited or removed emoticons from consideration, this research finds emoticons and emojis to have high explanatory power. And finally, we find the "retweet" class to be a distinct class, separate from direct sentiment and much more easily characterized in terms of sentiment. However, inclusion of retweets did not offer additional value and retweets are removed from our datasets.

To validate the effectiveness of the TSLS, and the domain transferability of its associated TGFS, we selected two machine learning tools, DAN2 and SVM, to model and perform TSA for all the five datasets. The results of our experiments validate that domain transferability has been achieved for our datasets and the accurate results of the TSA models show its effectiveness with both machine learning tools.

Our approach to sentiment analysis increases sensitivity, accounting for tweets with mild sentiment (positive and negative), and results in more accurate identification of the neutral category. The five point scale offered in this research enables decision makers to identify the mildly expressed sentiments (positive or negative) for actionable recommendations. When implementing this process on a larger scale, we show that even when datasets are imbalanced, our approach, especially when coupled with DAN2, produces very good results.

The use of a highly explanatory n-gram feature set in this research offers additional tools for brands to recognize emerging issues with their brand identity. This information, when used to influence brand decisions (brand offerings, frequency of brand messaging, timing of messaging, type of brand messaging, brand reactions to external factors), will allow brand managers to make better use of the Twitter service and to best influence public perception.

Extending this approach to documents larger than tweets may offer challenges requiring further research. However, existing research has only shown modest success in reaching improved accuracy when applied to "similar" applications. Specialized domains such as stocks, specific medical applications, or similar fields may offer the best chance for success.

We acknowledge that the spam removal solution offered in this research is neither comprehensive nor fully automated. Complete spam removal, especially in the presence of "for hire" spammers, is a challenging research topic that needs to be addressed separately.

Finally, we present results from several diverse domains as evidence to show the effectiveness of a domain transferable feature set with excellent TSA values. Although we compare TSLS against two leading lexicon sets, an indisputable and complete validation of these conclusions against all available feature sets will require a side-by-side analysis with the corresponding statistical evaluations for all five datasets. Additionally, using the TSLS with other sentiment analyzers, will allow evaluation of its effectiveness decoupled from DAN2 and SVM. Further research is needed to reach such complete and comprehensive conclusions. However, our results suggest that the effectiveness and reusability of the feature set can offer researchers an excellent alternative for Twitter sentiment analysis compared to the current state-of-the-art approaches.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2018.04.006.

References

Abbasi, A., Chen, H., & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS), 26*(3), #12.

Abbasi, A., Hassan, A., & Dhar, M. (2014). Benchmarking twitter sentiment analysis tools. In *Proceedings of the 9th international conference on language resources and evaluation* (pp. 823–829).

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30–38).

Andreevskaia, A., & Bergler, S. (2008). When specialists and generalists work together: Domain dependence in sentiment tagging. In *Proceedings of 46th annual meeting of the association for computational linguistics* (pp. 290–298).

Arnold, A., Nallapati, R., & Cohen, W. W. (2007). A comparative study of methods for transductive transfer learning. In *Proceedings of the 7th IEEE international conference on data mining workshops* (pp. 77–82).

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing*.

Barbosa, L., & Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd international conference on computational linguistics (COLING'10)* (pp. 36–44).

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the annual meetings of the ACL* (pp. 440–447).

Boucher, J. D., & Osgood, C. E. (1996). The Pollyanna hypothesis. *Journal of Verbal Learning and Verbal Behavior, 8*, 1–8.

Choi, Y., & Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 793–801).

Chung, J. E., & Mustafaraj, E. (2011). Can collective sentiment expressed on twitter predict political elections? In *Proceedings of the 25th international AAAI conference on artificial intelligence* (pp. 1770–1771).

Daume, H., III, & Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research, 26*, 101–126.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519–528).

Davidov, D., Tsur, O., & Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 241–249).

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231–240).

Dredze, M., Blitzer, J., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 440–447).

Dredze, M., & Crammer, K. (2008). Online methods for multi-domain learning and adaptation. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 689–697).

Ensuli, A., & Sebastiani, F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th conference on language resources and evaluation* (pp. 417–422).

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM, 56*(4), 82–89.

Gamon, M. (2004). Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the twentieth international conference on computational linguistics* (p. 841).

Ghiassi, M., Olschimke, M., Moon, B., & Arnaudo, P. (2012). Automated text classification using a dynamic artificial neural network model. *Expert Systems with Applications, 39*(12), 10967–10976.

Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications, 40*(16), 6266–6282.

Ghiassi, M., Zimbra, D., & Lee, S. (2016). Targeted twitter sentiment analysis for brands using supervised feature engineering and the dynamic architecture for artificial neural networks. *Journal of Management Information Systems, 33*(4), 1034–1058.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning* (pp. 513–520).

Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision* (pp. 1–6). Stanford Digital Library Technologies Project. Technical report.

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology, 60*(11), 2169–2188.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th annual meeting of the association for computational linguistics* (pp. 151–160).

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. Burges, & A. J. Smola (Eds.), *Advances in kernel methods* (pp. 169–184). Cambridge, MA: MIT Press.

Kajanan, S., Shafeeq Bin Mohd Shariff, A., Datta, A., Dutta, K., & Paul, D. (2011). Twitter post filter for mobile applications. In *Proceedings of the 21st workshop on information technology and systems* (pp. 1–6).

Kennedy, A., & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence, 22*(2), 110–125.

Kim, S. M., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on computational linguistics* (p. 1367).

Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications, 40*(10), 4065–4074.

Lexalytics. Frequently asked questions (FAQ) (2016). http://dev.lexalytics.com/wiki/pmwiki.php?n=Main.FAQ Accessed: 2014.08.15.

Lexalytics (2017). https://www.lexalytics.com/ Accessed: 2017.02.10.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. New York, NY: Cambridge University Press.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10Ks. *The Journal of Finance, 66*(1), 35–65.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Mansour, R., Refaei, N., Gamon, M., Abdul-Hamid, A., & Sami, K. (2013). Revisiting the old kitchen sink: Do we need sentiment domain adaptation? In *Recent advances in natural language processing* (pp. 420–427).

Mitchell, T. (2005). Reducing data dimension. In *Machine learning, 10-701* (pp. 1–40). Carnegie Mellon University. http://www.cs.cmu.edu/~guestrin/Class/10701-S05/slides/dimensionality.pdf.

Moreno-Ortiz, A., & Hernández, C. P. (2013). Lexicon-based sentiment analysis of twitter messages in Spanish. *Procesamiento del Lenguaje Natural, 50*, 93–100.

Musto, C., Semeraro, G., & Polignano, M. (2014). A comparison of lexicon-based approaches for sentiment analysis of microblog posts. In *Proceedings of the 8th international workshop on information filtering and retrieval. CEUR workshop proceedings* (pp. 59–68). *1314*.

Oliveira, N., Cortez, P., & Areal, N. (2014). Automatic creation of stock market lexicons for sentiment analysis using StockTwits data. In *Proceeding of the 18th international database engineering & applications symposium* (pp. 115–123).

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the 7th conference on language resources and evaluation* (pp. 1320–1326).

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering, 22*(10), 1345–1359.

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting of the association for computational linguistics* (p. 271).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the conference on empirical methods in natural language *processing* (pp. 79–86).

Piri, S., Delen, D., & Liu, T. (2017). A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. *Decision Support Systems*. https://doi.org/10.1016/j.dss.2017.11.006.

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In J. G. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing attitude and affect in text: Theory and applications* (pp. 1–10). Dordrecht: Springer.

Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge Based Systems, 108*, 42–49.

Repustate (2015). https://www.repustate.com/ Accessed: 2017.02.10.

Rogers, S. (2014). Insights into the #WorldCup conversation on Twitter. https://blog.twitter.com/2014/insights-into-the-worldcup-conversation-on-twitter Accessed: 2014.08.15.

Saif, H., He, Y., & Alani, H. (2012a). Alleviating data sparsity for twitter sentiment analysis. In *Proceedings of the 21st ACM international World Wide Web conference* (pp. 2–9).

Saif, H., He, Y., & Alani, H. (2012b). Semantic sentiment analysis of twitter. In *Proceedings of the 11th international semantic web conference* (pp. 508–524).

Salvetti, F., Reichenbach, & Lewis, S. (2006). Opinion polarity identification of movie reviews. In J. G. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing attitude and affect in text: Theory and applications* (pp. 303–316). Dordrecht: Springer.

Samdani, R., & Yih, W. T. (2011). Domain adaptation with ensemble of feature groups. In *Proceedings of the 22nd international joint conference on artificial intelligence* (p. 1458).

Sentiment140 (2013). http://www.sentiment140.com/ Accessed: 2017.02.10.

Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM research in applied computation symposium* (pp. 1–7).

Silva, N. F. F. D., Coletta, L. F., & Hruschka, E. R. (2016). A survey and comparative study of tweet sentiment analysis via semi-supervised learning. *ACM Computing Surveys (CSUR), 49*(1), #15.

Taboada, M., Brook, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307.

Tan, S., Wu, G., Tang, H., & Cheng, X. (2007). A novel scheme for domain-transfer problem in the context of sentiment analysis. In *Proceedings of the 16th ACM conference on information and knowledge management* (pp. 979–982).

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance, 62*(3), 1139–1168.

Thakkar, H., & Patel, D. (2013). Approaches for sentiment analysis on Twitter: A state-of-the-art study. In *Proceedings of the international network for social network analysis conference*.

Thrun, S., & Pratt, L. (Eds.). (2012). *Learning to learn*. New York, NY: Springer Science & Business Media.

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 417–424).

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., et al. (2005). OpinionFinder: A system for subjectivity analysis. In *Proceedings of conference on human language technology and empirical methods in natural language processing* (pp. 34–35).

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011).Combining lexicon-based and learning-based methods for twitter sentiment analysis.www.hpl.hp.com/techreports/2011/HPL-2011-89.html Accessed: 2014.08.15.

Zimbra, D., Ghiassi, M., & Lee, S. (2016). Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. In *Proceedings from the 49th Hawaii international conference on system sciences* (pp. 1930–1938).