On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data

Peter C. R. Lane[a], Daoud Clarke[a,b], Paul Hender[b]

[a]*School of Computer Science, University of Hertfordshire, College Lane, Hatfield AL10 9AB, Hertfordshire, UK*
[b]*Metrica, Banner Street, London EC1V 9BJ, UK*

Abstract

Locating documents carrying positive or negative favourability is an important application within media analysis. This article presents some empirical results on the challenges facing a machine-learning approach to this kind of opinion mining. Some of the challenges include: the often considerable imbalance in the distribution of positive and negative samples; changes in the documents over time; and effective training and evaluation procedures for the models. This article presents results on three datasets generated by a media-analysis company, classifying documents in two ways: detecting the presence of favourability, and assessing negative vs. positive favourability. We describe our experiments in developing a machine-learning approach to automate the classification process. We explore the effect of using five different types of features, the robustness of the models when tested on data taken from a later time period, and the effect of balancing the input data by undersampling. We find varying choices for the optimum classifier, feature set and training strategy depending on the task and dataset.

Keywords: Bayesian models, favourability analysis, imbalanced data, machine learning, sentiment analysis, support-vector machines

1. Introduction

Media analysis is a discipline closely related to content analysis [17], with an emphasis on analysing content with respect to: (1) *Favourability*, how favourable an article is with respect to an entity. This will typically be on a five point scale: very negative, negative, neutral, positive or very positive. (2) *Key messages*, topics or areas that a client is interested in. This allows the client to gain feedback on the success of particular public

Published in Journal of Decision Support Systems 2012, doi:10.1016/j.dss.2012.05.028

relations campaigns, for example. Media analysis has traditionally been done manually, however the explosion of content on the world-wide web, in particular social media, has led to the introduction of automatic techniques for performing media analysis, e.g., [20, 32].

In this article (an extended version of [7]), we discuss our recent findings in applying machine learning techniques to favourability analysis. The work is part of a two-year collaboration between Gorkana Group, which includes one of the foremost media analysis companies, Metrica, and the University of Hertfordshire. The goal is to develop ways of automating media analysis, especially for social media. However, the data used here are from traditional media (newspapers and magazines) since at the time of starting the experiment there were more manually analysed data available.

The Gorkana Group consists of three companies servicing the public relations industry:

Gorkana who provide a journalist database

Durrants who provide a media monitoring service

Metrica who provide media analysis and evaluation

The documents we use are supplied by Durrants, the media monitoring company within the Gorkana Group, and consist of text from newspaper and magazine articles in electronic form. Each document is analysed by trained human analysts at Metrica, given scores for favourability, as well as other characteristics which the client has requested. This dataset is used to provide feedback to the clients about how they are portrayed in the media, and is summarised by Metrica for clients' monthly reports.

The main challenges confronting a machine-learning project of this kind are firstly, how to represent the data; secondly, how to train the models; and thirdly, how to evaluate the resultant models giving the best information for their expected use. The representation problem has been widely discussed in the document classification literature; we explore the effectiveness of five different types of features, including different n-grams, entity words and dependency words. The training problem falls into two parts: given the large number of features typical of document classification, how can these be reduced to a more manageable number; and, given the imbalance in the dataset, how can we train the models to respect both classes equally? The evaluation problem also falls into two parts: given the imbalanced datasets, what is a useful measure of effectiveness of the models' performances; and, given that the models will be used on documents from a later time period,

what is the likely impact on performance? We consider the training problems by comparing approaches with balanced and imbalanced training sets, and by using feature selection to reduce the number of features. We tackle the evaluation problems using the geometric mean to measure performance, as this measure reflects the performance of both classes, and by using separate held-out test sets to look at performance from a later time period.

2. Favourability, Sentiment and Opinion Mining

Favourability analysis is very closely related to sentiment analysis, with the following distinction: sentiment analysis generally focuses on a (subjective) sentiment implying an opinion of the author, for example:[1]

(1) Microsoft is the greattteesssst at EVERYTHING

expresses the author's opinion (which others may not share) whereas favourability analysis, whilst taking into account sentiment, also measures favourable *objective* mentions of entities. For example:[2]

(2) Halloween Eve Was The Biggest Instagram Day Ever, Doubling Its Traffic

is an objective statement (no one can doubt that the traffic doubled) that is favourable with respect to the organisation, Instagram.

Similarly, subjectivity analysis focuses on distinguishing documents that are entirely objective from those expressing "opinions, evaluations and speculations" [37]. Our task is subtly different, since we are not so interested in the opinion of the author, but whether the expressed content (be it subjective or objective) is good or bad publicity for Metrica's client.

Also closely related to this task is that of determining implicit sentiment [2] in statements such as "I'm going to a party." Here it is assumed that the speaker is conveying an opinion in addition to the objective expressed meaning, and that this opinion is implicit in the meaning of the sentence: if the speaker likes parties then a positive sentiment is implied. If the sentence was spoken, we might imagine the sentiment conveyed in the tonality of expression. This task seems much closer to ours, since sentiment is now attached to statements that seem objective on the surface. Nevertheless we maintain that a distinction should be made, as the objective is

[1] Actually, this is an ironic comment on a blog post at TechCrunch.
[2] A headline from TechCrunch

different: we attempt to determine the favourability of the document for the *client* as opposed to discerning the opinion of the *speaker*.

The distinction may be important when considering the available information and opportunities for creating a machine-learning model. In sentiment or subjectivity analysis, the words themselves may convey the target sentiment, which may be picked up by statistical analysis. In favourability analysis, the words may need more interpretation and background knowledge, which will not be available to a pure learning-based system. Although many authors treat favourability and subjectivity (or sentiment) analysis together, we believe it is useful to distinguish the two tasks, if only to assist in evaluating comparative performance of learning systems applied to different kinds of documents. Having said this, the task of determining favourability clearly subsumes that of sentiment (subjectivity) analysis, since positive or negative sentiment generally implies a corresponding favourability. We therefore hypothesise that similar techniques will be useful in developing classifiers in both areas, and so provide a brief summary of the more relevant techniques.

The most closely related task to ours is opinion mining, determining sentiment with respect to a particular target. Balahur, Hermida and Montoyo [3] examine this task for newspaper articles. They show that separating out the objective favourability from the expressed sentiment led to an increase in inter-annotator agreement, which they report as 81%, after implementing improvements to the process; this improvement lends further support to our argument above, that it is worthwhile distinguishing favourability from sentiment. Melville, Gryc and Lawrence [22] report on an automated system for opinion mining applied to blogs, which achieves between 64% and 91% accuracy, depending on the domain, while Godbole, Srinivasaiah and Skiena [11] describe a system applied to news and blogs; these systems take advantage of prior lexical knowledge about the "sentiment-polarity" of words when assessing a text.

Pang, Lee and Vaithyanathan [27] introduced machine learning to perform sentiment analysis. They used naïve bayes, support-vector machines (SVMs) and maximum entropy on the movie review domain, and report accuracies between 77% and 83% depending on the feature set, which included unigrams, bigrams, and part-of-speech tagged unigrams. More recent work along these lines is described in [26, 28].

One approach to sentiment analysis is to build up a lexicon of sentiment carrying words. Turney [33] describes a way to automatically build such a lexicon based on co-occurrences of words with other words whose sentiment is known. This idea was extended by Gamon et al. [10] who also considered

the lack of co-occurrence as useful information.

Koppel and Schler [16] show that it is important to make use of neutral sentiment documents when considering the task of positive versus negative sentiment. They also show that some non-standard combinations of these tasks (such as negative against neutral) can provide useful information. Koppel and Schler suggest using stacked decisions to optimise the separation of documents into different classes. We separate the tasks of detection and type in our experiments, although we do not combine the results from the two tasks.

3. Feature Selection and Evaluation

Document classification, of which favourability analysis is an example, is an ideal domain for machine learning, because the raw data, the text, are easily manipulated, and often large amounts of text can be obtained, making the problems amenable to statistical analysis.

A classification model is essentially a mapping, from a document described as a set of feature values to a class label. In most cases, this class label is a simple yes-no choice, such as whether the document is favourable or not. In the experimental section of this article we describe results from applying a range of different classification algorithms to our datasets.

In general, two issues that affect machine-learning approaches are the selection of features, and the presence of imbalanced data. Their impact will vary depending on the precise learning algorithm used, and how training is conducted, as we discuss in our experiments.

3.1. Features

Useful features for constructing classification models from text documents include sets of unigrams, bigrams or trigrams, dependency relationships or selected words: we describe these features in the next section in relation to our datasets. From a machine-learning perspective, it is useful for the features to include only relevant information [5], and also to be independent of each other. This feature-selection problem has been tackled by several authors in different ways, and various studies of the effect of different feature-selection techniques have been performed, e.g., [9, 13, 24, 30]. In our experiments, we evaluate a technique to reduce the number of features using feature selection.

Alternative approaches to identify the sentiment of text attempt to go beyond the simple labelling of the presence of a word. Some authors have

described experiments augmenting the above feature sets with additional information. For example, WordNet has been used to add information about words found within text [4, 25], leading to improved classification performance in a sentiment analysis task [25]. Also, Li and Wu [20] analyse a document for sentiment carrying words, use K-means clustering to identify hotspots, and then employ machine-learning techniques to identify documents which fall within hotspots.

3.2. Imbalanced Data

Our datasets, as is usual in many real-world applications, present varying degrees of imbalance between the two classes, ranging from 70:30 to 94:6. Various approaches to handling imbalanced data, and experiments on the effect this has on the performance of learning algorithms, have been explored in the literature. Techniques for handling imbalanced data can be separated into two areas: during *training*, to ensure the model is capable of working with both classes, and in *evaluation*, to ensure a model with the best performance is selected for use on novel data.

First considering *evaluation*, the standard measure of accuracy (proportion of correctly classified examples) is inappropriate if 90% of the documents are within one class. A simple ZeroR classifier (selecting the majority class) will score highly, but it will never get any examples of the minority class correct. A good overview of the dangers of relying on accuracy is provided by Gray et al. [12], who argue that measures such as *precision* must be provided to give any meaningful evaluation of performance.

One proposal for managing imbalanced data is to modify the output threshold of those algorithms where it is appropriate to do so [29, 31]. For example, we might train a naïve Bayes classifier on a dataset, and adjust its output threshold to reflect the class distribution in the data. However, in our target application, classifying live data in a media-analysis company, we cannot assume anything about new data, including the class distribution, so adjusting the output threshold is not feasible in live use, although it is useful in evaluating a model against a known test set.

Frequently in the literature, the imbalanced data problem is discussed in the context of a single class. One class (usually the minority class) is of particular interest, and so evaluation measures are typically decided with respect to that class, for example, the F-measure, or taking an average of precision and recall. Alternative measures, such as those based on the ROC curve [36], are also frequently employed, especially to look at the impact of varying sample distribution [14]. In our application, it is important that both classes have good accuracy, so we use the geometric mean [18], which

combines the separate accuracy measures on the two classes as $\sqrt{a_1 \times a_2}$, where a_i denotes the proportion of instances from class i that were judged correctly. This has the property that it strongly penalises poor performance in any one class: if either a_1 or a_2 is zero then the geometric mean will be zero. This characteristic is important for our purposes, since it is "easy" to get high accuracy on the majority class, the measure will favour classifiers that perform well on the minority class without significant loss of accuracy in the majority class. In addition, the geometric mean does not give preference to any one class, unlike, for example, the F-measure.

The second area is to consider the *training* process. An imbalanced training set can lead to *bias* in the construction of a machine-learning model (as discussed, for example, by Mitchell [23]). Such effects are well-known in the literature. The explanation is that machine-learning algorithms generally attempt to construct the simplest hypothesis to fit the training data, following a principle such as Occam's razor or minimal description length. For example, a decision tree algorithm attempts to find the smallest tree fitting the training criteria. With an imbalanced dataset, the simplest hypothesis may tend to ignore examples from the minority class.

A popular range of approaches for handling imbalanced data in training use some kind of *sampling*: under or over sampling the examples from one class to create a balanced dataset based on the original training data. In our experiments we use undersampling (where a random sample is taken from the majority class to balance the size of the minority class); this technique has the disadvantage of discarding training data. In contrast, approaches such as SMOTE [6] or SMOTE with Different Costs (SDC) [1] are techniques for artificially creating new instances of the minority class, to balance the number in the majority class.

The 'best' approach is dependent on the precise dataset. Hulse et al. [34] present a series of experiments on imbalanced data, considering different choices of sampling technique and use of performance metrics. They found that undersampling produced "very good performance" overall, and this was particularly so under performance measures such as geometric mean. In one study [1], support-vector machines were tested on a range of imbalanced datasets, comparing undersampling, SMOTE and the author's own algorithm (SDC). As shown in their results, undersampling produced on average the second-best results (behind the author's own) for the support-vector machine on 10 UCI datasets. Sun et al. [31] applied ten different strategies to handling imbalanced data in a textual domain, and came to the conclusion that SVMs perform well on moderate amounts of imbalanced data without special treatment. Overall, these results support the idea that

Dataset	Mixed	V. Neg.	Negative	Neutral	Positive	V. Pos.
A	472	86	138	1610	1506	1664
C	7	0	5	2824	852	50
S	522	94	344	9580	2057	937

Table 1: Number of documents in each class for the datasets A, C and S.

random under-sampling of the majority class is a promising strategy, especially when there are adequate training data, but for optimal performance in any particular application some evaluative experiments on a range of techniques should be carried out.

4. Experiments in Favourability Analysis

In this section, we report on some experiments in favourability analysis. We conduct two sets of experiments, looking for the presence of favourability, and also if the favourability is positive or negative. Experiments are performed using three source datasets, a range of classifiers, five feature sets, and different choices for training strategy.

4.1. Description of Data

The source documents have been tagged by analysts for favourability and unfavourability, both of which are given a non-negative score that is indicative both of the number of favourable/unfavourable mentions of the organisation and of the degree of favourability/unfavourability. Neutral documents are assigned a score of zero for both favourability and unfavourability. We assign each document a class based on its favourability f and unfavourability u scores. Documents are categorised as follows:

$f > 0$ and $u > 0$: **mixed**
$f = 0$ and $u > 1$: **very negative**
$f = 0$ and $u = 1$: **negative**
$f = 0$ and $u = 0$: **neutral**
$f = 1$ and $u = 0$: **positive**
$f > 1$ and $u = 0$: **very positive**

Table 1 shows the number of documents in each category for three datasets A, C and S, which are anonymised to protect Metrica's clients' privacy. A and S are datasets for high-tech companies, whereas C is for a charity. This is reflected in the low occurence of negative favourability with dataset C. Datasets A and C contain only articles that are relevant to the client,

Dataset	Neutral	Non-neutral
A	1610	3866
C	2824	914
S	9580	3954

Table 2: Class distributions for pseudo-subjectivity task

Dataset	Positive	Negative
A	3170	224
C	902	5
S	2994	438

Table 3: Class distributions for pseudo-sentiment task

whereas S contains articles for the client's competitors. We only make use of favourability judgments with respect to the client, however, so those that are irrelevant to the client we simply treat as neutral. This explains the overwhelming bias towards neutral sentiment in dataset S.

In our experiments, we consider only those documents which have been manually analysed and for which the raw text is available. Duplicates were removed from the dataset. Duplicate detection was performed using a modified version of Ferret [19] which compares occurrences of character trigrams between documents. We considered two documents to be duplicates if they had a similarity score higher than 0.75.

We performed experiments for two tasks:
Pseudo-subjectivity — detecting the presence or absence of favourability. This is a two-class problem with **neutral** documents in one class, and all other documents in the other.
Pseudo-sentiment — distinguishing between documents with generally positive and negative favourability. In our experiments, we treat this as a two class problem, with **negative** and **very negative** documents in one class and **positive** and **very positive** documents in the other (ignoring mixed sentiment).

4.2. Method

We follow a similar approach to [27]: we generate features from the documents, and train a classifier using the manually analysed data.

We sorted the documents by time, and then selected the earliest two thirds as a training set, and kept the remainder as a held out test set. This allows us to get an idea of how the system will perform when it is in use,

Type	Relation	Term
governor	det	the
governor	rcmod	sued
governor	nn	leader
dependent	poss	conference
dependent	nsubj	bullish
dependent	dep	beat

Table 4: Example dependency relations extracted from the data. "Type" indicates whether the term referring to the organisation is the governor or the dependent in the expression.

since the system will necessarily be trained on documents from an earlier time period. We performed cross validation on the randomised training set, giving us an upper bound on the performance of the system, and we also measured the accuracy of every system on the held out dataset. We hypothesised that new topics would be discussed in the later time frame, and thus the accuracy would be lower, since the system would not be trained on data for these topics.

We also experimented with balancing the input data to the classifiers; each system was run twice, once with all the input data, and once with data which had been undersampled so that the number of documents in each class was the same. And we experimented with feature selection: reducing the number of features used to describe the dataset.

4.2.1. Features for documents

We used five types of features:

Unigrams, bigrams and trigrams: produced using the WEKA tokenizer [38] with the standard settings.[3]

EntityWords: unigrams of words occurring within a sentence containing a mention of the organisation in question. Mentions of the organisation were detected using manually constructed regular expressions, based on datasets for organisations collected elsewhere in the company. Sentence boundary detection was performed using an OpenNLP[4] tool.

Dependencies: we extract dependencies using the Stanford dependency parser [8]. For the purpose of this experiment, we only considered dependencies directly connecting the term relating to the organisation. Table 4 gives

[3]We used the StringToWordVectorClass constructed with an argument of 5,000.

[4]`http://opennlp.sourceforge.net`

example dependencies extracted from the data. For example, the phrase "...prompted [organisation name] to be bullish..." led to the extraction of the term *bullish*, where the organisation name is the subject of the verb and the organisation name is a dependent of the verb *bullish*. For each dependency, all this information is combined into a single feature.

4.3. Classification Algorithms

We used the following classifiers in our experiments: naïve Bayes, Support Vector Machines (SVMs), k-nearest neighbours with $k = 1$ and $k = 5$, radial basis function (RBF) networks, Bayesian networks, decision trees (J48) and a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (JRip). We also included two baseline classifiers, ZeroR, which simply chooses the most frequent class in the training set, and Random, which chooses classes at random based on their frequencies in the training set.

These are taken from the WEKA toolkit [38], with the exception of SVMs, for which we used the LibSVM implementation, naïve Bayes (since the Weka implementation does not appear to treat the value occurring with a feature as a frequency) and Random, both of which we implemented ourselves. We used WEKA's default settings for classifiers where appropriate.

4.3.1. Parameter search for SVMs

We used a radial-basis kernel for our SVM algorithm which requires two parameters to be optimised experimentally. This was done for each fold of cross validation. Each fold was further divided, and three-fold cross validation was performed for each parameter combination. We varied the gamma parameter exponentially between 10^{-5} and 10^{5} in multiples of 100, and varied cost between 1 and 15 in increments of 2. We used the geometric mean of the accuracies on the two classes to choose the best combination of parameters; using the geometric mean enables us to train and evaluate the SVM from either balanced or imbalanced datasets.

4.3.2. Feature Selection

Because of the long training time of many of the classifiers, we also looked at whether reducing the dimensionality of the data before training by performing feature selection would enhance or hinder performance. The feature selection was done by ranking the features using the Chi-squared measure and taking the top 250 with the most correlation with the class. The exception to this was k-nearest neighbours, for which we used random projections with 250 dimensions. For the RBF network we tried both feature

Dataset	Features	Best Classf.	Feat. Sel.	Balance	Cross val. acc.	Held out acc.
S		*Random*			*0.465 ± 0.008*	*0.461 ± 0.007*
S	Ent. Words	SVM	X		**0.912** ± 0.002	**0.952** ± 0.001
S	Unigrams	JRip	X	X	0.907 ± 0.002	**0.952** ± 0.002
S	Bigrams	SVM	X	X	0.875 ± 0.007	0.885 ± 0.004
S	Trigrams	Naïve Bayes			0.791 ± 0.003	0.759 ± 0.003
S	Dep.	RBFNet		X	0.853 ± 0.005	0.766 ± 0.054
C		*Random*			*0.417 ± 0.017*	*0.419 ± 0.027*
C	Ent. Words	Naïve Bayes	X		0.704 ± 0.011	0.640 ± 0.018
C	Unigrams	Naïve Bayes	X		0.735 ± 0.007	0.659 ± 0.032
C	Bigrams	Naïve Bayes			**0.756** ± 0.012	0.640 ± 0.014
C	Trigrams	Naïve Bayes			**0.757** ± 0.004	**0.679** ± 0.017
A		*Random*			*0.453 ± 0.004*	*0.453 ± 0.017*
A	Ent. Words	BayesNet	X		0.691 ± 0.008	**0.625** ± 0.019
A	Unigrams	SVM	X	X	**0.696** ± 0.005	0.619 ± 0.010
A	Bigrams	SVM	X	X	0.680 ± 0.012	0.609 ± 0.026
A	Trigrams	Naïve Bayes		X	0.610 ± 0.011	0.536 ± 0.019

Table 5: Results for the pseudo-subjectivity task, distinguishing documents neutral with respect to favourability from those which are not neutral. The accuracy was computed as the geometric mean of accuracy. The best-performing classifier on cross-validation is shown for each feature set, along with the Random classifier as a baseline. An indication is given of whether the best-performing system used feature selection and/or balancing on the input data.

selection and random projections, and naïve Bayes was run both with and without feature selection.

4.4. Results

Tables 5 and 6 show the best classifier on the cross-validation evaluation for each dataset and feature set for the pseudo-subjectivity and pseudo-sentiment tasks respectively, together with the Random classifier baseline. The accuracies shown were computed using the geometric mean of the accuracy on the two classes. This was computed for each cross-validation fold; the value shown is the (arithmetic) mean of the accuracies on the five folds, together with an estimate of the error in this mean. The values for the held out data were computed in the same way, dividing the data into five, allowing us to estimate the error in the accuracy.

5. Discussion

The results clearly show variability in the best classifier and results across datasets and training technique. We now discuss these results, looking sep-

Dataset	Features	Best Classifier	Balance	Cross val. acc.	Held out acc.
S		*Random*		*0.332 ± 0.023*	*0.365 ± 0.03*
S	EntityWords	Naïve Bayes	X	0.738 ± 0.008	0.552 ± 0.033
S	Unigrams	Naïve Bayes	X	0.718 ± 0.017	0.650 ± 0.024
S	Bigrams	Naïve Bayes	X	0.748 ± 0.013	0.682 ± 0.023
S	Trigrams	Naïve Bayes	X	**0.766** ± 0.014	**0.716** ± 0.038
S	Dependencies	Naïve Bayes		0.566 ± 0.014	0.523 ± 0.060
A		*Random*		*0.253 ± 0.026*	*0.111 ± 0.072*
A	EntityWords	Naïve Bayes	X	0.737 ± 0.016	0.656 ± 0.067
A	Unigrams	Naïve Bayes	X	0.769 ± 0.008	**0.756** ± 0.031
A	Bigrams	Naïve Bayes		0.755 ± 0.009	0.618 ± 0.157
A	Trigrams	Naïve Bayes		**0.800** ± 0.02	**0.739** ± 0.088

Table 6: Results for the pseudo-sentiment task, distinguishing positive and negative favourability. See the preceding table for details. None of the best performing systems used feature selection on this task. Dataset C is not shown, as there were not enough negative documents in the test set to compute the accuracies.

arately at the overall accuracy, feature choice and selection, and techniques for training from imbalanced data.

5.1. Overall accuracy

The most notable difference between the two tasks, pseudo-subjectivity and pseudo-sentiment, is that the best classifier for the pseudo-sentiment task was naïve Bayes in every case, whereas the best classifier varies with dataset and feature set for the pseudo-subjectivity task. This is presumably because the independence assumption on which the naïve Bayes classifier is based holds very well for the pseudo-sentiment task, at least with our datasets.

The level of accuracy we report for the pseudo-sentiment task is lower than that typically reported for sentiment analysis, e.g., [27], but in line with that from other results, such as [22]. This could be because favourability is harder to determine than sentiment. For example it may require world knowledge in addition to linguistic knowledge, in order to determine whether the reporting of a particular event is good news for a company, even if reported objectively.

Accuracy on the held out dataset is up to 10% lower than the cross-validation accuracy on the pseudo-subjectivity task, and up to 6% lower on the pseudo-sentiment task. This is probably due to a change in topics over time. This degradation in performance could be reduced by techniques such as those used to improve cross-domain sentiment analysis [21, 35].

		Imbalanced			Balanced		
Features	Classifier	Neut.	Non.	Cross val. acc.	Neut.	Non.	Cross val. acc.
EntityWords	BayesNet	0.968	0.850	0.907 ± 0.003	1	0	0 ± 0
EntityWords	J48	0.794	0.907	0.849 ± 0.005	0.908	0.882	0.895 ± 0.002
EntityWords	JRip	0.917	0.883	0.900 ± 0.004	0.966	0.852	0.907 ± 0.003
EntityWords	SVM	0.962	0.864	0.912 ± 0.003	0.959	0.864	0.911 ± 0.002
EntityWords	Naïve Bayes	0.969	0.850	0.908 ± 0.003	1	0	0 ± 0
EntityWords	RBFNet	0.856	0.894	0.875 ± 0.006	0.832	0.895	0.863 ± 0.011
Unigrams	BayesNet	0.933	0.745	0.834 ± 0.004	0.978	0.338	0.575 ± 0.008
Unigrams	J48	0.802	0.906	0.852 ± 0.002	0.933	0.867	0.899 ± 0.004
Unigrams	JRip	0.941	0.859	0.900 ± 0.005	0.967	0.851	0.907 ± 0.002
Unigrams	SVM	0.959	0.857	0.907 ± 0.002	0.954	0.859	0.905 ± 0.002
Unigrams	Naïve Bayes	0.774	0.789	0.781 ± 0.006	0.910	0.581	0.727 ± 0.008
Unigrams	RBFNet	0.402	0.946	0.616 ± 0.017	0.413	0.943	0.622 ± 0.019
Bigrams	BayesNet	0.899	0.821	0.859 ± 0.007	0.957	0.517	0.703 ± 0.012
Bigrams	J48	0.715	0.921	0.812 ± 0.006	0.844	0.845	0.844 ± 0.004
Bigrams	JRip	0.746	0.912	0.825 ± 0.005	0.801	0.828	0.813 ± 0.007
Bigrams	SVM	0.747	0.933	0.835 ± 0.006	0.849	0.901	0.875 ± 0.007
Bigrams	Naïve Bayes	0.883	0.716	0.795 ± 0.004	0.947	0.569	0.734 ± 0.005
Bigrams	RBFNet	0.614	0.941	0.760 ± 0.008	0.609	0.939	0.757 ± 0.006
Trigrams	BayesNet	0.620	0.883	0.739 ± 0.009	0.975	0.118	0.289 ± 0.086
Trigrams	J48	0.356	0.964	0.586 ± 0.012	0.441	0.942	0.644 ± 0.008
Trigrams	JRip	0.422	0.963	0.637 ± 0.003	0.388	0.963	0.605 ± 0.042
Trigrams	SVM	0.575	0.921	0.728 ± 0.008	0.604	0.909	0.740 ± 0.009
Trigrams	Naïve Bayes	0.810	0.758	0.784 ± 0.003	0.922	0.593	0.739 ± 0.005
Trigrams	RBFNet	0.459	0.949	0.659 ± 0.010	0.478	0.934	0.667 ± 0.013
Dependencies	BayesNet	0.678	0.931	0.794 ± 0.006	1	0	0 ± 0
Dependencies	J48	0.377	0.963	0.602 ± 0.012	0.437	0.947	0.643 ± 0.005
Dependencies	JRip	0.452	0.953	0.656 ± 0.008	0.638	0.929	0.769 ± 0.009
Dependencies	SVM	0.645	0.943	0.780 ± 0.006	0.683	0.928	0.796 ± 0.006
Dependencies	Naïve Bayes	0.764	0.914	0.835 ± 0.005	0.990	0.006	0.074 ± 0.007
Dependencies	RBFNet	0.508	0.961	0.698 ± 0.008	0.529	0.959	0.711 ± 0.012

Table 7: Balanced versus imbalanced cross validation accuracies (geometric mean) for dataset S, pseudo-subjectivity task, together with the accuracies on the individual classes, neutral and non-neutral. (Models trained without feature selection.)

		Imbalanced			Balanced		
Features	Classifier	Neut.	Non.	Cross val. acc.	Neut.	Non.	Cross val. acc.
EntityWords	BayesNet	0.987	0.044	0.205 ± 0.02	0.987	0.045	0.208 ± 0.019
EntityWords	J48	0.784	0.498	0.623 ± 0.011	0.596	0.712	0.651 ± 0.008
EntityWords	JRip	0.89	0.349	0.554 ± 0.023	0.643	0.694	0.668 ± 0.008
EntityWords	SVM	0.872	0.394	0.587 ± 0.006	0.575	0.812	0.683 ± 0.007
EntityWords	Naïve Bayes	0.972	0.111	0.326 ± 0.021	0.944	0.192	0.426 ± 0.015
EntityWords	RBFNet	0.71	0.591	0.596 ± 0.087	0.574	0.787	0.671 ± 0.01
Unigrams	BayesNet	0.835	0.378	0.56 ± 0.015	0.802	0.424	0.581 ± 0.013
Unigrams	J48	0.816	0.438	0.597 ± 0.013	0.67	0.629	0.649 ± 0.005
Unigrams	JRip	0.902	0.294	0.511 ± 0.024	0.653	0.658	0.654 ± 0.003
Unigrams	SVM	0.837	0.464	0.622 ± 0.011	0.694	0.698	0.696 ± 0.005
Unigrams	Naïve Bayes	0.896	0.318	0.531 ± 0.018	0.736	0.582	0.652 ± 0.012
Unigrams	RBFNet	0.829	0.37	0.552 ± 0.016	0.851	0.369	0.557 ± 0.021
Bigrams	BayesNet	0.89	0.321	0.534 ± 0.008	0.849	0.392	0.577 ± 0.009
Bigrams	J48	0.847	0.324	0.523 ± 0.013	0.593	0.716	0.652 ± 0.012
Bigrams	JRip	0.942	0.141	0.353 ± 0.038	0.636	0.674	0.654 ± 0.01
Bigrams	SVM	0.852	0.36	0.553 ± 0.006	0.58	0.8	0.68 ± 0.012
Bigrams	Naïve Bayes	0.959	0.203	0.439 ± 0.017	0.86	0.433	0.605 ± 0.024
Bigrams	RBFNet	0.908	0.28	0.501 ± 0.019	0.804	0.428	0.56 ± 0.02
Trigrams	BayesNet	0.919	0.216	0.443 ± 0.019	0.903	0.24	0.464 ± 0.019
Trigrams	J48	0.963	0.102	0.306 ± 0.03	0.376	0.864	0.57 ± 0.011
Trigrams	JRip	0.999	0.006	0.036 ± 0.036	0.366	0.858	0.561 ± 0.006
Trigrams	SVM	0.935	0.173	0.401 ± 0.018	0.407	0.851	0.588 ± 0.009
Trigrams	Naïve Bayes	0.938	0.249	0.481 ± 0.013	0.84	0.446	0.61 ± 0.011
Trigrams	RBFNet	0.951	0.144	0.368 ± 0.014	0.948	0.17	0.398 ± 0.02

Table 8: Balanced versus imbalanced cross validation accuracies (geometric mean) for dataset A, pseudo-subjectivity task, together with the accuracies on the individual classes, neutral and non-neutral. (Models trained without feature selection.)

5.2. Features

Trigrams proved the most effective feature type in 3 out of the 5 different experiments, with unigrams and entity words proving the best in 1 case each. However, in many cases, there is not a significant difference between the results for different datasets.

Although we only computed dependencies for one dataset, S, we found that they did not provide significant benefit on their own. This may be due to the sparseness of the data, since we only extracted dependencies with respect to the organisation in question. Dependencies may be useful when combined with other features, such as unigrams.

Feature selection was not always effective in improving classification, even with the high-dimensionality of the data. In the pseudo-sentiment task, none of the best classifiers used feature selection. In the pseudo-subjectivity task, 8 out of 13 results showed a benefit in using feature selection. This is probably because, as shown by [15], there are few irrelevant features in text. This issue deserves further exploration, not least because reducing the number of features can considerably speed-up the training process. Even if few features are totally irrelevant, there will likely be a subset of greater importance. Some authors [9, 39] argue that selecting the right features may be more important for classification in situations with imbalanced data, and this is an important area for future work.

5.3. Imbalance

Finally, we look at our results considering the imbalanced data problem. Within some of the algorithms, balance is actively taken account during the training process: e.g., naïve Bayes has a weighting on its class output to compensate for different frequencies, and the SVM training process uses geometric mean for computing performance, which encourages a good performance on imbalanced data. In addition, we have presented results on the difference between training with balanced and imbalanced datasets. Better results are obtained in 6 out of the 13 results for the pseudo-subjectivity task (Table 5), and in 6 out of 9 results for the pseudo-sentiment task (Table 6), suggesting that balancing the training data is a useful technique in most cases.

However, a surprising result is found in Table 7, which shows pseudo-subjectivity results for dataset S with and without balanced input data, and the best results for each type of classifier. This dataset has an approximately 70:30 imbalance in the class distribution. Interestingly, balancing the data shows mixed results for this dataset. In particular, the accuracy of

the Bayesian network, and sometimes the naïve Bayes classifier, are severely reduced when the training data are balanced. We found similar behaviour with dataset C (with a 75:25 imbalance), however, as shown in Table 8, we found the converse on dataset A (with a 30:70 imbalance): every classifier performed better with balanced data. Further, Table 6 shows that balancing data has proven effective for the naïve Bayes classifiers in the pseudo-sentiment task, where the imbalance is more severe (94:6 for A, and 88:12 for S).

Given these results, we suggest that balancing the training datasets using random undersampling of the majority class is usually an effective strategy, although sometimes the benefits are small if account of balancing is also part of the parameter-selection process for the learning algorithm. These results broadly support those obtained across a range of different datasets [34], but are in contrast to those obtained in other document-classification studies [31]. It is therefore advisable, when applying these techniques, that preliminary experiments are performed to identify the most effective technique on the specific dataset used.

6. Conclusion and Further Work

We have empirically analysed a range of machine-learning techniques for developing favourability classifiers in a commercial context. These techniques include different classification algorithms, use of feature selection to reduce the feature sets, and treatment of the imbalanced data problem. Also, we used five different types of features to create the datasets from the raw text. We have found a wide variation, from less than 0.7 to over 0.9 geometric mean of accuracy, depending on the particular set of data analysed. We have shown how balancing the class distribution in training data can be beneficial in improving performance, but some algorithms (in particular naïve Bayes) can be adversely affected. In future work we will apply these techniques to larger volumes of social media, and further explore the questions of balancing datasets, other features and feature selection, as well as the opportunities and implications of embedding these algorithms within the workflow of the company.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support-vector machines to imbalanced datasets. In J-F. Boulicaut, F. Esposito, F. Gi-

annotti, and D. Pedreschi, editors, *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 39–50, 2004.

[2] A. Balahur, J.M. Hermida, and A. Montoyo. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In A. Balahur, E. Boldrini, A. Montoyo, and P. Martinez-Barco, editors, *Proceedings of the Second Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 53–60, Portland, Oregon, 2011. Association for Computational Linguistics.

[3] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. Van Der Goot, M. Halkia, B. Pouliquen, and J. Belyaeva. Sentiment analysis in the news. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 2216–2220, 2010.

[4] J.P. Bao, C.M. Lyon, and P.C.R. Lane. A text annotation method based on semantic sequence. In *Proceedings of the Seventh International Workshop on Computational Semantics*, 2007.

[5] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97:245–271, 1997.

[6] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6:1–6, 2004.

[7] D. Clarke, P.C.R. Lane, and P. Hender. Developing robust models for favourability analysis. In A. Balahur, E. Boldrini, A. Montoyo, and P. Martinez-Barco, editors, *Proceedings of the Second Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 44–52. Association for Computational Linguistics, Portland, Oregon, 2011.

[8] M-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parsers from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, 2006.

[9] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.

[10] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. *Advances in Intelligent Data Analysis VI*, pages 121–132, 2005.

[11] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[12] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson. Further thoughts on precision. In *Proceedings of Evaluation and Assessment in Software Engineering*. 2011.

[13] P.D. Green, P.C.R. Lane, A.W. Rainer, and S. Scholz. Selecting measures in origin analysis. In M. Bramer, M. Pehidis, and A. Hopgood, editors, *Research and Development in Intelligent Systems XXVII: Proceedings of the Thirtieth SGAI International Conference on Artificial Intelligence*, pages 379–92. Springer-Verlag, 2011.

[14] H. He and E.A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, 2009.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer Berlin / Heidelberg, 1998.

[16] M. Koppel and J. Schler. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22:100–109, 2006.

[17] K. Krippendorff. *Content analysis: An introduction to its methodology*. Sage Publications, Inc, 2004.

[18] M. Kubat, R.C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30:195–215, 1998.

[19] P.C.R. Lane, C.M. Lyon, and J.A. Malcolm. Demonstration of the Ferret plagiarism detector. In *Proceedings of the Second International Plagiarism Conference*, 2006.

[20] N. Li and D. D. Wu. Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems*, 48:354–368, 2010.

[21] T. Li, V. Sindhwani, C. Ding, and Y. Zhang. Knowledge transformation for cross-domain sentiment classification. In *Proceedings of the Thirty-Second International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 716–717. ACM, 2009.

[22] P. Melville, W. Gryc, and R. D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the Fifteenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 1275–1284, New York, NY, USA, 2009.

[23] T. Mitchell. *Machine Learning*. New York: McGraw-Hill, 1997.

[24] D. Mladenić. Feature subset selection in text-learning. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 95–100. Springer Berlin / Heidelberg, 1998.

[25] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, volume 4, pages 412–418, 2004.

[26] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2008.

[27] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[28] R. Prabowo and M. Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3:143–157, 2009.

[29] F. Provost. Machine learning from imbalanced data sets 101. In *AAAI Workshop on Learning from Imbalanced Data Sets*. AAAI Press, 2000.

[30] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.

[31] A. Sun, E-P. Ling, and Y. Lui. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, 48:191–201, 2010.

[32] G. Tatzl and C. Waldhauser. Aggregating opinions: Explorations into graphs and media content analysis. In *TextGraphs-5 Workshop, ACL 2010*, page 93, 2010.

[33] P.D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[34] J. van Hulse, T.M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In Z. Ghahramani, editor, *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 935–942, 2007.

[35] X. Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1, pages 235–243. Association for Computational Linguistics, 2009.

[36] C.G. Weng and J. Poon. A new evaluation measure for imbalanced datasets. In *Proceedings of the Seventh Australasian Data Mining Conference*, 2008.

[37] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin. Learning subjective language. *Computational linguistics*, 30:277–308, 2004.

[38] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[39] Z. Zheng, X. Wu, and R. Srihari. Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter: Special issue on learning from imbalanced datasets*, 6:80–89, 2004.