# ONLINE MEDIA SENTIMENT: UNDERSTANDING MACHINE LEARNING-BASED CLASSIFIERS

*Research in Progress*

Riekert, Martin, University of Hohenheim, Stuttgart, Germany, martin.riekert@uni-hohenheim.de

Leukel, Joerg, University of Hohenheim, Stuttgart, Germany, joerg.leukel@uni-hohenheim.de

Klein, Achim, University of Hohenheim, Stuttgart, Germany, achim.klein@uni-hohenheim.de

## Abstract

*Online media is an important source for sentiments exposed by individuals on goods, services, organizations, and other objects of interest. While firms can benefit from using these sentiments for decision-making, the classification of sentiments is difficult because of volume, velocity, and variety. Machine learning is an effective technique for sentiment classification, which neither requires formalized knowledge about the domain nor the language used. Although the literature provides a rich body of classification methods, system designers and researchers still face the problem of reasonably selecting designs. In this paper, we seek to contribute to the understanding of machine learning for sentiment classification. We report an experimental study that tests the effects of three design factors, i.e., text representation, feature weighting, and machine learning algorithm, on accuracy. The findings can be useful for empirically informed classifier design.*

*Keywords: Machine Learning, Web Mining, User-generated Content, Sentiment Analysis.*

## 1 Introduction

Online media is an important source for consumer sentiments, which in turn impact consumer behavior (Dhar and Chang, 2009; Krauss et al., 2008). Consumer sentiments can be helpful for improving marketing campaigns (Fan and Gordon, 2014), product quality (Abrahams et al., 2012), and supply chain coordination (Leukel et al. 2011). However, the extraction of consumer sentiment is made difficult by the volume, velocity, and variety of the unstructured text data found. Sentiment analysis provides techniques for retrieving sentiments from these sources (Feldman, 2013). A particularly important subfield of sentiment analysis is *sentiment classification*.

Sentiment classification estimates whether a given document holds a positive or negative sentiment polarity concerning the object of interest (e.g., movie, restaurant, stock). Early classification methods suffered from the knowledge acquisition bottleneck, i.e., the methods required the formalization of knowledge by human experts (Sebastiani, 2002). This bottleneck was overcome by supervised machine learning, particularly with (1) the introduction of support vector machines (SVM) to text classification (Joachims, 1998) and (2) the publication of reference datasets (Pang et al., 2002).

Researchers in information systems (IS) have begun adopting sentiment classification for developing IT artifacts that support managerial decisions (Martens and Provost, 2014; Yang et al., 2010). However, adoption by IS researchers insufficiently reflects the understanding of machine learning-based methods in the sentiment classification literature. For instance, a review of sentiment classification in the finance domain found that most studies still follow the dictionary-based approach (Kearney and Liu, 2014), which typically yields lower accuracies than machine learning approaches (Tang et al.,

2009; Tsytsarau and Palpanas, 2011). System designers face the problem of reasonably devising sentiment classifiers. Often, rather complex classifiers are constructed but not validated against the baseline classifiers (e.g., Klein et al. 2013; Oh and Sheng, 2011). This deficit impairs the accumulation of knowledge on when and why particular methods work (Ren et al., 2013). To make matters worse, the no-free lunch theorem in machine learning makes it impossible that one classifier is a-priori superior than another (Wolpert, 1996).

Against this background, the aim of this paper is to provide guidance to IS researchers for how to increase the accuracy of SVM-based sentiment classification by more rigorously using the archival knowledge provided by the sentiment classification literature. Our research sets out to contribute to the understanding of SVM-based sentiment classification that is adapted to the needs of IS research. Specifically, the research question investigated in this paper is: What is the effect of three key design factors, i.e., text representation, feature weighting, and machine learning algorithm, on the accuracy of sentiment classifiers for online media? The main finding of our experimental study is that using simpler methods can increase accuracy, and thus reduce the effort of method design. System designers can use the tested propositions as an empirical underpinning of their design decisions. This basis can help reducing extensive experimentation with custom classifiers.

The remainder of this paper is organized as follows. We first present our research model and hypotheses. Then, we describe our experiment and the results obtained. We discuss the contributions, implications, and limitations of our study before concluding the paper.

## 2 Research Model

An overview of our research model is provided in Figure 1. We describe the components of our model by drawing on two sources of extant knowledge. First, we define machine learning-based sentiment classification as a process structured into three steps, which correspond to the three factors (independent variables) of our research model. Second, for each factor we discuss findings from prior research.

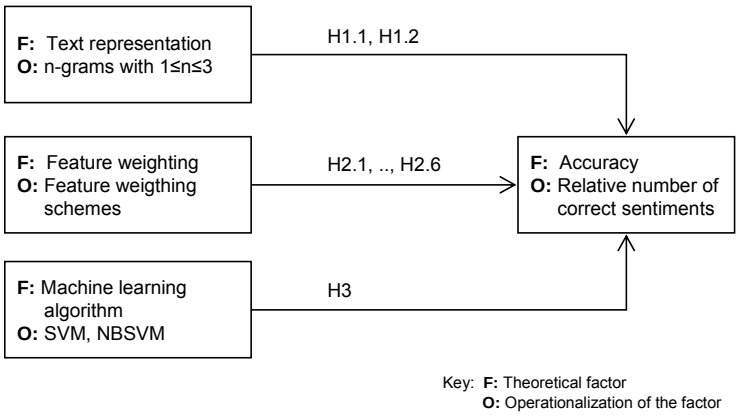


*Figure 1. Research model*

The task of *sentiment classification* is automatically determining the sentiment polarity of a given document that contains text in natural language (Tang et al., 2009). In the broadest meaning, sentiment can be defined as an opinion or belief toward an object of interest such as a particular movie, person, company, or stock (Liu, 2012). Sentiment is measured on an ordinal scale. In most cases, a binary scale is used, i.e., negative (−1) and positive (+1). Thus, the ultimate goal of sentiment classification is to map each document onto a sentiment class, i.e., $d_i \rightarrow y$, with $d_i \in D$ for the document from the set of all documents $D = \{d_1, \ldots, d_n\}$ and $y \in \{-1, +1\}$ for the sentiment (in case of only two sentiment values on an ordinal scale).

The classification process can be structured into three steps:

1. The input document $d_i$ is transformed into an initial vector representation $x_i$ (feature vector), where words or other elements in the document constitute the features (i.e., elements of the vector).
2. The numerical value of each feature in the vector is transformed (feature weighting).
3. The final feature vector $x_i$ is handed over to a machine learning algorithm that applies a classification model on the vector and determines the sentiment $y_i$.

For the final step to succeed, the machine learning algorithm has learned the classification model from an annotated training set. This learning takes place prior to the classification of documents with unknown sentiment. Accuracy is the percentage of correctly classified documents (Manning et al., 2008).

## 2.1 Design factor: Text representation

The first consideration is which terms in a document will be represented as a feature of the unified vector space. The rationale is to include features that carry relevant information for the classification problem and to exclude features that do not help discriminating similar documents of different sentiment (Manning et al., 2008). From a linguistic perspective, the meaning of documents can be represented through complex structures (phrases, sentences, paragraphs) that conform to some grammar (Chomsky, 1965). From a machine learning perspective, the goal is to represent texts on a much lower level of complexity. Still, identifying these features might be contingent to the domain of interest. Prior research suggests three general procedures for text representation (Joachims, 2002):

- *Unigrams:* Each single term will be represented in the vector, irrespective of its ordering and position in the document, e.g., {'the', 'best', 'movie', 'ever', 'made'}.
- *Bigrams:* Each two sequential terms will be represented in the vector, e.g., {'the-best', 'best-movie', 'movie-ever', 'ever-made'}.
- *Trigrams:* Each sequence of three terms will be represented in the vector, e.g., {'the-best-movie', 'best-movie-ever', 'movie-ever-made'}.

These text representations are also referred to as *n*-grams, with *n* denoting the number of terms that will be represented as a feature in the vector. In our research model, n-grams are the levels of the text representation factor. Discrimination analysis suggests that even the occurrence of single terms (unigrams) can very well serve as a discriminator (Manning et al., 2008), specifically for judging the *topic* of a text. While early research produced mixed results when using phrases as features and suggested that accuracy might even decrease and comes at high computational costs, studies into bi- and trigrams for sentiment classification provide evidence for increases of accuracy (Kennedy and Inkpen, 2006; Matsumoto et al., 2005; Wang and Manning, 2012). The argument for our first proposition is that bigrams allow for representing phrases such as adjective-noun, and adverb-verb, which are often used for expressing sentiments. Therefore, we state the hypothesis that adding two-term phrases (i.e., bigrams) into the representation will positively affect classification accuracy:

*H1.1: Using uni- and bigrams leads to higher accuracy than using unigrams.*

The argument for using trigrams is similar by better capturing language patterns that occur in sentiment-laden documents. The empirical findings for the usefulness of trigrams are still ambiguous and might have been confounded by other factors. While some studies detected decreases by adding trigrams (Pak and Paroubek, 2010; Wang and Manning, 2012), the study by Ng et al. reports higher accuracies for uni/bi/trigrams than Wang and Manning (2012) but does not compare results to uni/bigrams (Ng et al., 2006). These divergent findings motivate us to conjecture the effect of uni/bi/trigrams as follows:

*H1.2: Using uni-, bi- and trigrams leads to higher accuracy than using uni- and bigrams.*

## 2.2 Design factor: Feature weighting

Feature weighting determines the numerical values stored in a feature vector. Approaches for feature weighting can be described by three general components as follows: term frequency, inverse document frequency, and length normalization (Salton and Buckley, 1988). These components represent the (mathematical) factors that are multiplied to get the actual weight. Table 1 summarizes the components by giving their formula and a three-letter code (Paltoglou and Thelwall, 2010).

| **Component** | **1. Term frequency** | | **2. Inverse document frequency** | | **3. Length normalization** | |
|---|---|---|---|---|---|---|
| **Code** | *n* | *b* | *n* | *t* | *n* | *c* |
| **Formula** | *tf* | 1, if $tf > 0$<br>0, if $tf = 0$ | 1 | $log \frac{N}{df}$ | 1 | $\frac{1}{\| x_i \|_2}$ |

Key: ***tf:*** term frequency, ***N:*** total number of documents, ***df:*** number of documents that contain the feature.

*Table 1. Components of feature weighting*

Based on these components, we can derive particular *feature weighting schemes*, e.g., $ntn = tf \times log(N/df) \times 1$. In each code, *n* signifies the basic weighting schema. For instance, *nnn* determines the feature weight by the absolute term frequency (tf), whereas *ntc* calculates the inverse document frequency (idf), and then normalizes the vector to length 1. The coding also allows describing the most commonly used schemes such as tf-idf (*ntn*) and term presence (*bnn*, with *b* indicating the binary value). The general pattern of effects is that all codes different from *n* should increase accuracy. Although the effects of each component compared to the baseline schema *nnn* are well understood, controlling for interaction effects from either component is necessary (Leopold and Kindermann, 2002).

Term frequency mapped onto a binary variable (so called term presence) was found superior over absolute frequency (O'Keefe and Koprinska, 2009; Pang et al., 2002). One possible explanation is that word frequency per document has little impact on the sentiment but the occurrence is more important (Pang et al., 2002). We posit the effect of document frequency (in the first component) in two hypotheses. H2.1 considers no length normalization (third component), while H2.2 does – as follows:

*H2.1: Using bnn leads to higher accuracy than using nnn.*

*H2.2: Using bnc leads to higher accuracy than using nnc.*

The rationale for inverse document frequency (idf) stems from *Zipf's Law*, which states that few words occur very often, whereas most words occur very seldom (Zipf, 1949). Common words (e.g., 'a', 'the') do not help in discriminating documents. The idf term weighting will decrease the importance of such common words. Therefore, we concur that idf increases accuracy (Joachims, 1998; Robertson, 2004) relative to the previous design:

*H2.3: Using btn leads to higher accuracy than using bnn.*

*H2.4: Using btc leads to higher accuracy than using bnc.*

The argument for length normalization is due to situations in which documents greatly differ in size. Then, shorter documents will be represented by shorter feature vectors, while longer documents by feature vectors with overall higher values (Salton and Buckley, 1988). Dissimilar vector length potentially reduces classification accuracy because documents with similar content but different length will be represented differently. Therefore, inserting a normalization factor into the weighting formula can increase accuracy (Ng et al., 2006; Pang et al., 2002). We posit this effect as follows:

*H2.5: Using nnc leads to higher accuracy than using nnn.*

*H2.6: Using bnc leads to higher accuracy than using bnn.*

### 2.3 Design factor: Machine learning algorithm

Machine learning algorithm concerns the means for processing vector-based representations to (1) learn a classification model and (2) apply it to input documents with unknown sentiment. The most used algorithm for sentiment classification are *Support Vector Machines* (SVM) (Cortes and Vapnik, 1995). SVM-based classifiers can achieve accuracies of about 90% (Tang et al., 2009), and thus outperform earlier classifiers that use the *Multinomial Naïve Bayes* (MNB) algorithm (McCallum and Nigam, 1998). Therefore, our research model includes SVM but not MNB. A recently proposed classifier is NBSVM, for which higher accuracies than for SVM have been reported (Wang and Manning, 2012). NBSVM combines a Naïve Bayes classifier with a SVM. For this purpose, an interpolation parameter $\beta$ is used, which allows to weight the importance of the SVM compared to the Naïve Bayes classifier (Wang and Manning, 2012). We add this proposition to our research model as follows:

*H3: Using NBSVM leads to higher accuracy than using SVM.*

## 3 Method

We describe the experimental method to test our research model. The experiment had a three-way factorial repeated measures design. Hence, each document was subject to three treatments, i.e., text representation, feature weighting, and machine learning algorithm. The 3x8x2 factorial experiment allowed us to compare results obtained for a total of 48 treatment conditions.

Our dataset provided movie sentiments and contained 50,000 reviews retrieved from the Internet Movie Database (IMDb). The dataset was created by Maas et al. (2011). Our *training set* and *test set* had each 25,000 documents. Each set provided an equal number of positive and negative reviews (rated on the 1-10 scale; positive for values larger than five and negative for values smaller than six). Our dataset was of sufficient size to form distinct sets for training and testing (without overlaps). We were able to perform McNemar's test, which is specific to repeated measures designs with one dichotomous independent variable (here: the two levels compared in each hypothesis) and one dichotomous dependent variable (here: correct vis-à-vis incorrect classification). In each test, we chose between two classifiers for a large sample. For this statistical question, McNemar's test has low probability of type I errors but high power and outperforms the so called 10-fold cross validation with t-tests (Dietterich, 1998). We report *p*-values with the conservative Yates correction.

We used LIBLINEAR, which is a publicly available SVM implementation (Fan et al., 2008). We applied the default configuration (L2-regularized and L2-loss dual-form SVM with linear kernel, penalty $C$=1, and margin of tolerance $\varepsilon$=0.01). Similarly, we used the standard configuration of the NBSVM algorithm (with $\beta$=0.25) (Wang and Manning, 2012).

## 4 Results

Table 2 provides accuracies for the three factors under study. First, screening the data allows to identify the best setup as follows: The highest accuracy was achieved for uni-, bi-, and trigrams as the text representation level and btc as the feature weighting level 91.24% for NBSVM, i.e. {uni/bi/tri, btc, NBSVM}, and 91.06% for {uni/bi/tri, btc, SVM}, closely followed by {uni/bi/tri, ntc, NBSVM}, {uni/bi/tri, bnn, NBSVM} and {uni/bi, btc, NBSVM}. Second, we made pair-wise comparisons of classifiers as required for each hypothesis. For instance, H1.1 was tested by comparing all the results in the columns for uni/bi of Table 2 with all the results in the columns for uni.

| | | *Factor 3: Machine learning algorithm* | | | | | |
|---|---|---|---|---|---|---|---|
| | | **SVM** | | | **NBSVM** | | |
| | | ***Factor 1: Text representation*** | | | ***Factor 1: Text representation*** | | |
| | | **uni** | **uni/bi** | **uni/bi/tri** | **uni** | **uni/bi** | **uni/bi/tri** |
| ***Factor 2: Feature weighting*** | **nnn** | 84.41 | 88.45 | 89.02 | 84.60 | 89.58 | 90.55 |
| | **nnc** | 88.24 | 89.58 | 89.69 | 88.10 | 89.62 | 89.70 |
| | **ntn** | 84.18 | 89.38 | 89.97 | 82.68 | 88.82 | 89.87 |
| | **ntc** | 87.28 | 90.19 | 90.64 | 87.36 | 90.61 | **91.04** |
| | **bnn** | 84.70 | 88.60 | 89.32 | 85.67 | 90.14 | **90.92** |
| | **bnc** | 87.98 | 89.88 | 90.20 | 88.60 | 90.50 | 90.62 |
| | **btn** | 84.26 | 89.76 | 90.37 | 83.64 | 89.14 | 89.92 |
| | **btc** | 87.41 | 90.77 | **91.06** | 87.74 | **90.94** | **91.24** |

*Table 2. Accuracies for all 48 treatment conditions*

**Effect of text representation (H1):** Here we investigate that uni and bigrams (uni/bi) outperform unigrams (H1.1) and uni-, bi-, and trigrams achieve the highest accuracy (H1.2). Perusal of Table 3 indicates strong support for both hypotheses. The effect posited in H1.1 holds for all levels of text representation and machine learning algorithm, while the effect stated in H1.2 was observed for all levels but was nonsignificant in 3 out of 16 tests.

| **Hypothesis** | ***Factor 3: ML algorithm*** | ***Factor 2: Feature weighting*** | | | | | | | | **Summary** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **nnn** | **nnc** | **ntn** | **ntc** | **bnn** | **bnc** | **btn** | **btc** | |
| **H1.1** uni/bi > uni | **SVM** | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | Support |
| | **NBSVM** | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | |
| **H1.2** uni/bi/tri > uni/bi | **SVM** | .000 | .152 | .000 | .000 | .000 | .000 | .000 | .000 | Support but not for nnc/bnc (NBSVM) |
| | **NBSVM** | .000 | .198 | .000 | .000 | .000 | .085 | .000 | .001 | |

*Table 3. Test of hypotheses H1.1 and H1.2 (McNemar's test: significant at p<.05)*

**Effect of feature weighting (H2):** Here we examine how the three components of feature weighting affect accuracy. The first two hypotheses are concerned with using binary term frequency in comparison with using absolute term frequency. Perusal of Table 4 shows strong support for H2.1 and H2.2 in case of NBSVM. Support in case of SVM was less strong, with two tests reporting non-significant effects and one test reporting a non-significant effect in the opposite direction (signified by 'opp').

| **Hypothesis** | ***Factor 3: ML algorithm*** | ***Factor 1: Text representation*** | | | **Summary** |
|---|---|---|---|---|---|
| | | **uni** | **uni/bi** | **uni/bi/tri** | |
| **H2.1** bnn > nnn | **SVM** | .121 | .298 | .022 | Support for NBSVM but weak for SVM |
| | **NBSVM** | .000 | .000 | .000 | |
| **H2.2** bnc > nnc | **SVM** | opp / .095 | .023 | .000 | Support but not for SVM-unigrams |
| | **NBSVM** | .000 | .000 | .000 | |
| **H2.3** btn > bnn | **SVM** | opp / .001 | .000 | .000 | Some support for SVM, no support for NBSVM |
| | **NBSVM** | opp / .000 | opp / .000 | opp / .000 | |
| **H2.4** btc > bnc | **SVM** | opp / .000 | .000 | .000 | Support but not for unigrams |
| | **NBSVM** | opp / .000 | .000 | .000 | |
| **H2.5** nnc > nnn | **SVM** | .000 | .000 | .000 | Support for SVM but weak for NBSVM |
| | **NBSVM** | .000 | .804 | opp / .000 | |
| **H2.6** bnc > bnn | **SVM** | .000 | .000 | .000 | Support |
| | **NBSVM** | .000 | .012 | .000 | |

*Table 4. Test of hypotheses H2.1 through H2.6 (McNemar's test: significant at p<.05)*

In H2.3 and H2.4, we hypothesize that idf causes an increase of accuracy. The supposed effect is contingent to text representation. Specifically, we observed substantially lower accuracies for all tests with unigrams as text representation. Finally, H2.5 and H2.6 describe effects of length normalization

on accuracy. Support in case of SVM was strong across all other factors but weaker for NBSVM (significant in four tests across all other factors; one test was non-significant regarding the hypothesized orientation and one test was statistically significant opposite to the hypothesized orientation).

**Effect of machine learning algorithm (H3):** Here we enquire whether NBSVM is superior to SVM. Perusal of Table 5 reveals significant differences for four feature weighting schemes (strong support in case of bnn and bnc, slightly weaker support in case of nnn and ntc). The tests for the other four schemes yielded mixed results, with some tests signifying a decrease of accuracy.

| Hypothesis | *Factor 1: Text representation* | *Factor 2: Feature weighting* | | | | | | | | Summary |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **nnn** | **nnc** | **ntn** | **ntc** | **bnn** | **bnc** | **btn** | **btc** | |
| **H3** NBSVM > SVM | **uni** | .439 | opp .366 | opp .000 | .697 | .000 | .000 | opp .012 | .080 | Support for 2 of 8 levels of factor 2 |
| | **uni/bi** | .000 | .803 | opp .005 | .005 | .000 | .000 | opp .001 | .261 | Support for 4 of 8 levels of factor 2 |
| | **uni/bi/tri** | .000 | .934 | .600 | .004 | .000 | .003 | opp .000 | .211 | |

*Table 5. Test of hypothesis H3 (McNemar's test: significant at p<.05)*

# 5 Discussion

The results show that two hypotheses received full support (H1.1, 2.6), four hypothesis received slightly weaker support (H1.2, 2.1, 2.2, 2.5), and support of three hypotheses was subject to considerable confounding effects (H2.3, 2.4, 3). While the basic causal relationships stated in our research model were largely observed, we did not hypothesize about confounding effects.

Our study has several implications for sentiment classification tasks with long documents:

- Our study suggests that the more advanced levels of each design factor positively affect accuracy. In summary, we recommend the configurations {uni/bi/tri, btc, SVM}, {uni/bi, btc, SVM}, and {uni, btc, SVM}. In contrast to our findings, Fang et al. (2012) used only unigrams (which is contradictory to H1.1 and H1.2), and Oh and Sheng (2011) described their text representation and feature weighting ambiguously as "bag of words" and used a decision tree-based algorithm.
- Making the right decision for text representation and feature weighting has a bigger impact than for machine learning algorithm. As our results show, the mean difference between the worst and best configurations was 4.40% for SVM and 4.43% for NBSVM (over all the levels of feature weighting). The second largest effect was due to feature weighting: The worst/best mean difference was 2.81% for SVM and 3.19% for NBSVM, respectively (over all the levels of text representation).
- Although NBSVM achieved highest accuracy, our testing of hypothesis H3 showed mixed results. Thus, NBSVM should be used with caution.

However, we note that the no-free lunch theorem in machine learning limits the generalizability of our results (Wolpert, 1996). Three characteristics of the problem are of importance. First, small datasets weaken the applicability of classification models with a large number of features (Hastie et al., 2009). Specifically, bigrams and trigrams vastly increase the number of features, which then hinders high accuracy. Second, the classification task must be considered. For sentiment classification, binary features (e.g., btc) were shown to increase accuracy, while for topic classification the classical tf-idf (ntc) is recommended (Joachims, 1998). Third, the length of the documents also affects performance, i.e., for short snippet tasks MNB achieves higher accuracy than SVM but not higher than NBSVM (Wang and Manning, 2012). In summary, the results of our study do not necessarily generalize to settings of small datasets, other classification tasks than sentiment classification, or documents of smaller size.

Our study also has implications for IS research. Prior research that did not evaluate their classifiers in terms of accuracy could be advanced (e.g., Benthaus et al., 2013; Madlberger and Nakayama, 2013). Much of the IS literature focused on the impact of sentiments on a dependent variable for decision-making without assessing the classification accuracy. The dependent variables include, for instance, abnormal returns on financial markets (Feuerriegel and Neumann, 2013; Nann and Krauss, 2013) and firm equity value (Yu et al., 2013). We believe that in case of a positive effect of sentiments on such decision variables, an increase of classification accuracy will yield further improvement. We suggest future research to study the relationship between classification accuracy and the decision variable.

Finally, the findings must be interpreted in light of its limitations due to the experimental form. Although our experimental setup and the procedures followed were similar to practice described in prior research (e.g., by using the default configurations of the SVM and NBSVM classifier), we did not control for the parameter settings offered by the classifier implementations. More experimentation is required to study the strengths of effects and their interactions. We acknowledge that further determinants of classification accuracy exist such as feature selection because large document sets result in increasing vector spaces (Guyon and Elisseeff, 2003; O'Keefe and Koprinska, 2009).

# 6 Conclusions

Our research contributes to understanding the determinants of sentiment classification accuracy for online media texts. These determinants should be considered when designing sentiment classifiers. Our study suggests concrete propositions to this respect, which shorten the path for practitioners and researchers for designing highly accurate sentiment classifiers for online media domains. Our research model includes three fundamental design factors, i.e., text representation, feature weighting, and machine learning algorithm. We experimentally examined effects on accuracy for 48 treatment conditions on a large dataset of 50,000 documents taken from the movie reviews domain. While our experimental validation provides substantial statistical evidence for the validity of most of our hypotheses, we also observed considerable confounding effects. For reproducibility of or results, the MATLAB code used in our experiment can be retrieved from https://wi2.uni-hohenheim.de/analytics.

# References

Abrahams, A. S., Jiao, J., Wang, G. A. and Fan, W. (2012). "Vehicle defect discovery from social media." *Decision Support Systems* 54 (1), pp. 87–97.

Benthaus, J., Pahlke, I., Beck, R. and Seebach, C. (2013). "Improving sensing and seizing capabilities of a firm by measuring corporate reputation based on social media data." In: *Proceedings of the 21st European Conference on Information Systems (ECIS 2013)*, Utrecht, The Netherlands.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge: MIT Press.

Cortes, C. and Vapnik, V. (1995). "Support-vector networks." *Machine Learning* 20 (3), 273–297.

Dhar, V. and Chang, E. (2009). "Does chatter matter? The impact of user-generated content on music sales." *Journal of Interactive Marketing* 23 (4), 300–307.

Dietterich, T. G. (1998). "Approximate statistical tests for comparing supervised classification learning algorithms." *Neural Computation* 10 (7), 1895–1923.

Fan, R., Chang, K. and Hsieh, C. (2008). "LIBLINEAR: A library for large linear classification." *Journal of Machine Learning Research* 9, 1871–1874.

Fan, W. and Gordon, M. D. (2014). "The power of social media analytics." *Communications of the ACM* 57 (6), pp. 74–81.

Fang, F., Datta, A. and Dutta, K. (2012). "A hybrid method for cross-domain sentiment classification using multiple sources." In: *Proceedings of the 33rd International Conference on Information Systems (ICIS 2012)*, Orlanda, FL, USA.

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

Feuerriegel, S. and Neumann, D. (2013). "News or noise? How news drives commodity prices." In: *Proceedings of the 34th International Conference on Information Systems (ICIS 2013)*, Milano, Italy.

Guyon, I. and Elisseeff, A. (2003). "An introduction to variable and feature selection." *Journal of Machine Learning Research* 3, 1157–1182.

Hastie, T., Tibshirani, R. and Friedman, F. (2009). *The Elements of Statistical Learning.* 2nd Edition. New York: Springer.

Joachims, T. (1998). "Text categorization with support vector machines: Learning with many relevant features." In: *Proceedings of the 10th European Conference on Machine Learning*. Ed. by C. Nédellec and C. Rouveirol. Berlin: Springer, pp. 137–142.

Joachims, T. (2002). *Learning to Classify Text using Support Vector Machines*. Norwell: Kluwer Academic Publishers.

Kearney, C. and Liu, S. (2014). "Textual sentiment in finance: A survey of methods and models." *International Review of Financial Analysis* 33, 171–185.

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

Klein, A., Altuntas, O., Riekert, M. and Dinev, V. (2013). "Combined approach for extracting object-specific investor sentiment from weblogs." In: *Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI 2013)*, Leipzig, Germany, pp. 691–705.

Krauss, J., Nann, S., Simon, D., Fischbach, K. and Gloor, P. (2008). "Predicting movie success and academy awards through sentiment and social network analysis." In: *Proceedings of the 16th European Conference on Information Systems (ECIS 2008)*, Galway, Ireland.

Leopold, E. and Kindermann, J. (2002). "Text categorization with support vector machines. How to represent texts in input space?." *Machine Learning* 56 (1-3), 423–444.

Leukel, J., Karaenke, P., Jacob, A., Kirn, S. and Klein, A. (2011). "Individualization of goods and services – Towards a logistics knowledge infrastructure for agile supply chains." In: *Proceedings of the 2011 AAAI Spring Symposium on AI for Business Agility*. Palo Alto: AAAI, pp. 36–49.

Liu, B. (2012). "Sentiment analysis and opinion mining." *Synthesis Lectures on Human Language Technologies* 5 (1), 1–167.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. and Potts, C. (2011). "Learning word vectors for sentiment analysis." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*. Stroudsburg: ACL, pp. 142–150.

Madlberger, M. and Nakayama, M. (2013). "On top of the world, down in the dumps: Text mining the emotionality of online consumer reviews." In: *Proceedings of the 21st European Conference on Information Systems (ECIS 2013)*, Utrecht, The Netherlands.

Manning, C. D., Raghavan, P. and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Martens, D. and Provost, F. (2014). "Explaining data-driven document classifications." *MIS Quarterly* 38 (1), 73–99.

Matsumoto, S., Takamura, H. and Okumura, M. (2005). "Sentiment classification using word subsequences and dependency sub-trees." In: *Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2005)*, Ed. by T. B. Ho, D. Chueng and H. Liu. LNCS 3518. Berlin: Springer, pp. 301–311.

McCallum, A. and Nigam, K. (1998). "A comparison of event models for Naive Bayes text classification." In: *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, Madison, WI.

Nann, S. and Krauss, J. (2013). "Predictive analytics on public data – The case Of stock markets." In: *Proceedings of the 21st European Conference on Information Systems (ECIS 2013)*, Utrecht, The Netherlands.

Ng, V., Dasgupta, S. and Arifin, N. (2006). "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews." In: *Proceedings of the Conference on Computational Linguistics (Coling) and Association for Computational Linguistics (ACL) Main Conference Poster Sessions*. Stroudsburg: ACL, pp. 611–618.

O'Keefe, T. and Koprinska, I. (2009). "Feature selection and weighting methods in sentiment analysis." In: *Proceedings of the 13th Australasian Document Computing Symposium*, Sydney, Australia.

Oh, C. and Sheng, O. (2011). "Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement." In: *Proceedings of the 32nd International Conference on Information Systems (ICIS 2011)*, Shanghai, PR China.

Pak, A. and Paroubek, P. (2010). "Twitter as a corpus for sentiment analysis and opinion mining." In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malta, pp. 1320–1326.

Paltoglou, G. and Thelwall, M. (2010). "A study of Information Retrieval weighting schemes for sentiment analysis." in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: ACL, pp. 1386–1395.

Pang, B., Lee, L. and Vaithyanathan, S. (2002). "Thumbs up? Sentiment classification using machine learning techniques." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg: ACL , pp. 79–86.

Ren, J., Ge, H., Wu, X., Wang, G., Wang, W. and Liao, S. (2013). "Effective sentiment analysis of corporate financial reports." In: *Proceedings of the 34th International Conference on Information Systems (ICIS 2013)*, Milano, Italy.

Robertson, S. (2004). "Understanding inverse document frequency: On theoretical arguments for IDF." *Journal of Documentation* 60 (5), 503–520.

Salton, G. and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval." *Information Processing & Management* 24 (5), 513–523.

Sebastiani, F. (2002). "Machine learning in automated text categorization." *ACM Computing Surveys* 34 (1), 1–47.

Tang, H., Tan, S. and Cheng, X. (2009). "A survey on sentiment detection of reviews." *Expert Systems with Applications* 36 (7), 10760–10773.

Tsytsarau, M. and Palpanas, T. (2011). "Survey on mining subjective data on the web." *Data Mining and Knowledge Discovery* 24 (3), 478–514.

Wang, S. and Manning, C. D. (2012). "Baselines and bigrams: Simple, good sentiment and topic classification." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers – Volume 2*. Stroudsburg: ACL, pp. 90–94.

Wolpert, D. H. (1996). "The lack of a priori distinction between learning algorithms." *Neural Computation* 8 (7), 1341–1390.

Yang, C. C., Tang, X., Wong, Y. C. and Wei, C.-P. (2010). "Understanding online consumer review opinions with sentiment analysis using machine learning." *Pacific Asia Journal of the Association for Information Systems* 2 (3), 73–89.

Yu, Y., Duan, W. and Cao, Q. (2013). "The impact of social and conventional media on firm equity value: A sentiment analysis approach," *Decision Support Systems* 55 (4), 919–926.

Zipf, G. (1949). *Human Behavior and the Principle of Least Effort*. Mansfield Centre: Martino Publishing.