Enabling Reproducible Sentiment Analysis: A Hybrid Domain-Portable Framework for Sentiment Classification

Matthias Eickhoff(✉)

Georg-August University, Göttingen, Germany
matthias.eickhoff@wiwi.uni-goettingen.de

Abstract. In this paper a hybrid framework for Sentiment Analysis is presented. In the first part, dictionary based and machine learning based Sentiment Classification are introduced and the two approaches are contrasted. In the second part of the paper, the HSentiR framework, which combines the two approaches, is introduced. Consequently, the framework is evaluated regarding scoring accuracy and practical concerns.

Keywords: Sentiment analysis · Reproducible research

1 Introduction and Research Problem

Content and sentiment analysis as fields of study have intrigued researchers for a long time. As early as the nineteenth century, the quality of newspaper-articles was studied on a statistical basis [1]. However, due to the exponential increase in readily available digital texts that also has resulted from the rise of social media, sentiment analysis has become one of the most active data mining topics. Popular techniques include the use of (1) machine learning approaches, such as of Support Vector Machines (SVM) and naïve Bayes classification, or (2) scoring by comparing the words in a text with a dictionary of sentiment words of known polarity [2]. These two approaches to sentiment analysis have specific advantages and limitations. While dictionary based scoring methods offer a higher level of domain portability than machine learning based ones, their ability to detect sentiment in a document remains limited to the used dictionary and, for optimal scoring accuracy, a domain-specific dictionary is desirable nonetheless [3]. On the other hand, machine learning based classifiers are typically not domain-portable at all because they are based on different statistical measures of similarity and consequently perform much worse, if the documents at hand are not comparable to initial training data [4]. Due to these drawbacks of the individual approaches, researchers have combined them in hybrid models, which strive to combine advantages of both methods. It has been shown that the combination of two or more methods can improve scoring accuracy [5]. However, these models remain widely inaccessible to the scientific community at large. Thus, neither the validation of existing research, nor the application of existing implementations of these models, can be

B. Donnellan et al. (Eds.): DESRIST 2015, LNCS 9073, pp. 215–229, 2015.
DOI: 10.1007/978-3-319-18714-3_14

done easily. However, reproducible research has been identified as being key to build trust in the validity of empirical research [6], especially when it is computationally-assisted [7]. In order to be able to reproduce the results of sentiment analysis, both the data used for the study and the computational method used to calculate the results may need to be made available, at least to the reviewers of the paper, ideally to the general public. While the availability of data is a research project-specific problem and often hindered by licensing and privacy concerns, the methods used to perform the analysis should be made available whenever possible. The goal of this research is to combine the advantages of hybrid-classification methods with enabling reproducible research in sentiment analysis tasks. Due to this goal, the presented approach is implemented in R, is domain portable and improves scoring-accuracy over dictionary-based scoring alone. The proposed framework for hybrid domain-portable sentiment analysis is modular and can be easily reproduced or modified using the publicly available source code and R-script files.[1] The remainder of the paper is organized as follows: The first section gives a brief overview of sentiment analysis methods and their different prerequisites, as well as some of their strengths and limitations. The second part presents the developed hybrid sentiment analysis framework HSentiR as a framework to combine different approaches leveraging their individual strengths. The third part presents an empirical evaluation of the framework, using the popular movie-review polarity corpus by Pang and Lee [8]. Results show that the dictionary-based stage of the process performs comparably to other implementations, provided a domain-appropriate dictionary is used. The machine learning stage of the process improves the scoring accuracy. Whether this is a result of the machine-learning algorithm used in the example (k-NN) or a domain-specific result is an interesting question for further research. Apart from the slight increase in scoring accuracy, the machine learning stage offers the advantage of faster scoring of new documents and independence of sentiment-dictionaries.

2 Theoretical Background

Sentiment analysis, as a subcategory of opinion mining [2], describes the field of study that tries to summarize the emotional, or opinionated, contents of texts in a manner that allows for a quick grasp of these properties for arbitrary amounts of text. Practical sentiment analysis applications range from improving the quality of restaurant reviews [9], over stock market prediction [10, 11], to the classification of movie reviews [12], political analysis [13] or the measurement of consumer confidence [14]. As noted, the field has a long history. This hardly surprises, as the opinion of the masses have always been of interest to scientific scrutiny. However, in the past such studies had to concentrate on topics like newspaper articles [1], because the personal opinions of the individual were not available to researchers. Social Media has changed this data landscape fundamentally. Today, for every major public event, thousands if not millions of tweets, blog or forum posts are available online and often

[1] These files are available on: http://www.uni-goettingen.de/en/482273.html.

can be accessed through an Application Programming Interface (API) in real time to those willing to pay for the privilege. Due to this exponential increase in available data, sentiment analysis—yesterday's scientific curiosity—has become a necessity for businesses and the politically ambitious alike. The task of automated sentiment extraction from texts is not a trivial one, even if digital texts are as freely available as they are today. This is due to the fact, that unlike human readers, automated classification systems are not able to detect the subtleties of human communication by default. In example, while the sentence "I love chocolate" will easily lend itself to analysis, another example such as "Don't I just love politicians, they are all so honest!" obviously poses a number of challenges. In fact, the second example contains three major challenges to sentiment analysis; sarcasm (irony), negation and the use of sentiment words to express the opposite of their expected sentiment. A fourth such challenge is identified by Liu, who notes that superficially objective sentences, such as "My car's motor stopped working a week after I bought it", carry a sentiment that, while being obvious to a human reader, will be virtually undetectable through a pattern based analysis [2]. Another especially difficult problem is the use of the rrealis [15], e.g. "Had Rome not fallen, we might all be called Julius". While the use of a single grammatical phenomenon, such as the irrealis, might not seem problematic since the usage of the construct is relatively rare, these challenges to sentiment analysis have to be considered as a whole, as errors due to them will accumulate and skew the results of the analysis. While there are a number of publications on each of those specific problems, here the focus will remain on sentiment analysis in general. Still, the addition of mechanisms that deal with these problems would constitute worthwhile extensions to the processes described later on. Meanwhile, a possible way to mitigate such problems would be to analyze texts that are assumed to contain sentiment but employ factual language, such as governmental press releases.

2.1 Two Approaches to Sentiment Analysis

There are two popular approaches to sentiment analysis. One is to treat the analysis as a classification problem and use supervised or unsupervised learning methods to cluster texts, sentences or individual words into categories (e.g. positive and negative), while the other is to use sentiment lexica containing the semantic orientation of a given set of words [16]. Both approaches have a number of advantages and disadvantages and have been the subject of a variety of research, both for dictionary-based methods [17, 18] and machine learning based approaches [19, 20]. One major difference between the methods is portability, i.e. the ability to use a method established in one domain on text in another. While it is almost pointless to try to use the result of supervised learning outside of the domain used to train the classification algorithm [16], it is intuitive that the words contained in sentiment lexica will carry most of their semantic orientation across domains (and perhaps more importantly the orientation will seldom change to its opposite). Still, a dictionary intended for cross-domain use cannot be expected to perform as well as a domain-specific one because the choice of words that express a particular sentiment differs greatly from domain to domain. In example, a positive movie review might use words like "entertaining" or "stunning", while a positive analyst report regarding a company’s financial performance might contain words like “continuity” or “increase”. Thus, specialized dictionaries are, of course, ideal and are a core-requirement whenever a high scoring accuracy is desired. Another disadvantage of supervised learning approaches is their reliance on frequentist properties of the data. Frequentist properties denote properties related to the field of frequentist statistics, which focuses on relative frequencies [21]. For an introduction to such properties see Held or Mayo and Cox [22, 23]. In example, if a given corpus of reviews contains only one case of each positive and negative attribute expressed by the reviewers, there will be no statistical pattern to discern and the machine learning (ML) approach will fail. A similar argument can be made for Bayesian approaches. In either case, the models rely on a statistically discernable difference between the groups or categories of texts. A dictionary might still contain a large part of these attributes [16]. On the other hand, machine learning based classification can assign texts to categories, which are not “identical” to the training data used to create the classifier, while dictionary based scoring requires an absolute matching of terms. In a sense ML based methods capture the latent sentiment of words via their relation to one another, while dictionary based approaches rely on explicit mappings to categories. It is also important to remember that these two approaches are by no means mutually exclusive. Hybrid approaches have been successfully employed to combine the portability of dictionary-based approaches with some automation [16]. Examples include Read and Carroll, as well as Li et al. [24, 25]. Indeed, such a combined method is the basis for the HSentiR framework introduced here. Liu differentiates between three types of sentiment dictionaries [2]. At first, sentiment dictionaries were created manually, which of course takes time and cannot be done on a project specific basis [2]. The second approach is to create sentiment dictionaries from normal dictionaries and is called "The Dictionary based Approach" by Liu [2]. This approach will be used in the work at hand. When talking about it, it is important to remember that according to Liu the name refers to the (general) dictionaries used to create the sentiment dictionary, and not the newly created lists of sentiment words that will be called sentiment dictionaries. Since all dictionaries used in the paper will fit this description, here "dictionary-based" will simply refer to sentiment analysis using a dictionary instead of a machine learning approach. A third approach described by Liu is to create the sentiment dictionary-based on a specific corpus. This approach, while being attractive due to the perspective of creating a sentiment dictionary that is corpus specific, requires large corpora to function properly [2]. Any dictionary-driven content analysis approach is "[...] done on a hit-or-miss basis" [26]. Either the sentiment vocabulary used in the studied corpus is contained in the dictionary or it is not. Thus, a dictionary suited to the domain of interest could not be more critical to a study’s success. Consequently, many different dictionaries have been developed since the early days of computer-aided content analysis. As there now are a great number of different dictionaries available from various publications, the following section will be restrained to giving an overview of possible ways to create dictionaries and listing some of the most commonly used in secondary research, thus being by no means comprehensive. The basic assumption made when using such dictionaries is that the words contained therein have a prior polarity [27], e.g. the word "good", when considered without context, will be perceived as positive by most people. This prior polarity is used to assign words to a sentiment category. Of course, a word's prior polarity will not always coincide with its contextual polarity, e.g. "fast" might be contained as a positive word in a dictionary for the automobile domain and a text might contain the phrase "it broke fast". Where such violations of the assumption occur, they introduce a bias to the analysis. Muhammad et al. propose a distant-supervision approach to generate domain specific dictionaries that mitigates this bias [3]. Such a method might be a prudent addition to dictionary bootstrapping techniques. Stone et al. pioneered the dictionary-driven approach and the work they started has been continually improved ever since [28]. The dictionary-based content analysis tool they created is commonly referred to as The General Inquirer (GI) and performs a variety of tasks from corpus pre-processing to result summarization [28]. The original goals of this tool are still relevant today. It was created to provide a possibility to operationalize theory, enable researchers to use a comparable and reproducible procedure and reduce the manual work required for content analysis [28]. For the purposes of this text, GI is considered mainly for its dictionary. For a more detailed description including the corpus pre-processing techniques and tagging procedures used by GI in its original version see [28]. Today, GIs dictionary has been extended to encompass a total of 175 categories and includes both The Harvard IV-4 categories (IV-4) and The Lasswell dictionary (Lasswell). The first two categories contained in the dictionary are positive and negative words with 1915 and 2291 entries respectively. For both types of sentiment classification, texts are commonly aggregated in a data-structure referred to as a corpus [29]. In addition to the texts themselves, this corpus can also contain metadata, such as authors, geo-locations or the time a certain text was created.

2.2 Requirements Facing a Sentiment-Classification Framework

As noted in the last section, both dictionary-based and machine learning based sentiment classification methods have a number of disadvantages. Therefore, combined methods are desirable because they can mitigate these disadvantages. What are the key issues researchers face when working with sentiment classification systems? When working with different solutions, data-formats can make it difficult to use the output of one tool as the basis for further analysis. In addition, most methods described in the literature are simply not available for research-use. When tools are available, they are generally not intended for hybrid use. Based upon these practical concerns, what requirements should a good framework for sentiment classification meet? First, it should be integrated into the research-workflow. This requirement can help to reduce the need for data-transformation and re-entry, thus reducing the likelihood of errors during these tasks. Due to the diversity of available statistics packages covering these and many other fields of interest, R offers a wide user base already familiar with a powerful statistical toolset and programming language. Furthermore, these existing packages can be used to perform the entire content analysis process, from data import to the statistical examination of the results, within one application framework. In addition, the framework should be modular, in order to allow researchers to use project-specific methods. This is a key requirement for research-purposes because only by allowing for the adoption of novel-methods, new knowledge can be incorporated in future research. In addition, ideally, the framework should be open and reproducible because "black-box" methods are undesirable in practical research. Most of all, the framework should be easy to use. Ideally, researchers would always use the most accurate classification solution, which reflects the latest advances in Sentiment Analysis. However, due to factors like time constraints and familiarity with certain software implementations, the best solution will not prevail if it is hard to use. Thus, ease of use and proper documentation are key features of content analysis software. Finally, the framework should provide the needed facilities needed in order to create reproducible research. While graphical user interfaces may be more intuitive for beginners, researchers have a need for script-based input formats because such scripts can easily be shared with reviewers and the public.

3 HSentiR: A Hybrid Sentiment Analysis Framework

Here, a two-staged hybrid framework for sentiment analysis using R is presented. Figure 1 shows a simplified illustration of the process.

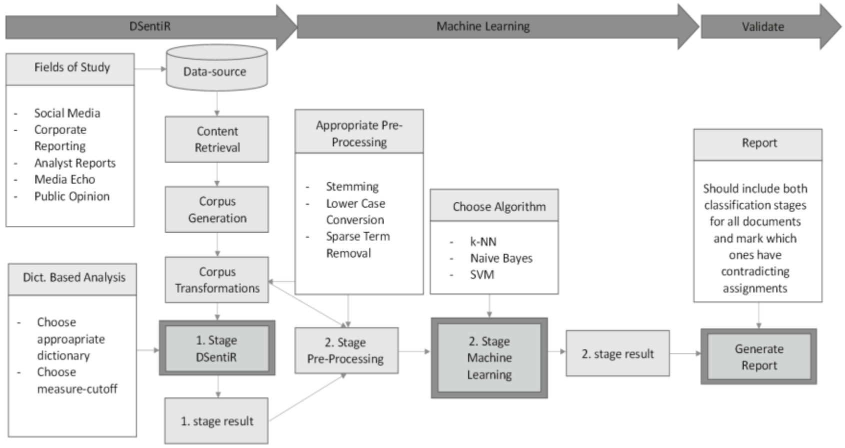

Fig. 1. The HSentiR Framework (Hybrid Sentiment Analysis in R)

The analysis process begins by importing text-data from a data source into R. Due to the large number of available R packages for such tasks, many APIs (e.g. Twitter) can be directly accessed for this purpose. Of course, data-import from a variety of file formats is also possible "out of the box". In order to be able to work with large amounts of text, a standardized storage format needs to be chosen. Such collections of text are commonly referred to as Corpora [29]. Here, due to the mutually exclusive needs of the two classification methods used throughout HSentiR, two storage formats are used. The dictionary-based classification uses a list-structure, which allows for corpus-wide transformations and cleanup tasks while retaining the input data in its original form. The machine learning stage utilizes the corpus class of the tm package available on the Comprehensive R Archive Network (CRAN). Note that the tm package also includes a variety of other text mining related tools, as well as pre-processing capabilities, such as stemming and the creation of term-document-matrices (TDM), which are a convenient basis for machine learning based analysis in R. This twofold storage structure enables custom pre-processing for the two stages of the analysis. Indeed, this possibility is needed because the pre-processing needs of the two stages are mutually exclusive. Pre-processing for a dictionary-based analysis should aim to increase the matching probability between the dictionary and the corpus, while machine learning based classification benefits from pre-processing tasks like sparse term removal, which would decrease matching probability with the dictionary. In the subsequent subsections, the two stages of the HSentiR framework will be described in more detail, before putting the framework to the test using movie-review data [8].

Step1: DSentiR – Dictionary Based Scoring

As shown in Figure 1, the dictionary based scoring phase of the process generates the training data for stage 2. Figure 2 gives a more detailed overview of the dictionary based classification process this stage of the HSentiR process.

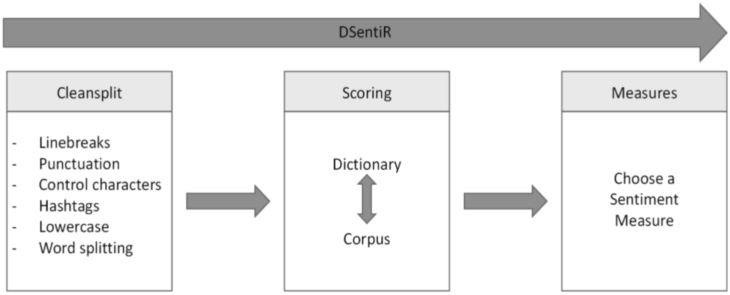

Fig. 2. The dictionary based stage of HSentiR: DSentiR

As the figure illustrates, the cleansplit function provided by the DSentiR package covers a number of common pre-processing tasks, such as removing characters from that data that might hinder the matching of words with the sentiment dictionary and splitting texts into individual words. This function is easily expandable using custom Regular-Expression (Regex) patterns and is therefore easily adapted to domain-specific pre-processing needs. Processing a text with this function results in a vector consisting of individual words. Afterwards, the scoring function is used to match this word-vector with the sentiment dictionary. The function returns the match-count between the dictionary and the supplied text. Optionally, the function can also return the matched words themselves if a "sanity check" is desired. Finally, the match counts are handed over to a function containing the chosen sentiment measures, which determine the classification of each text based on the scores. As is, the *sentiment.measures* function outputs the proportion of positive matches in relation to the total match-count as the default measure:

$$score = \frac{pos-neg}{pos+neg} \tag{1}$$

It is assumed that if this percentage exceeds 50 a text is of positive sentiment. Indeed, this intuitive cut-off value is very near to the empirical optimum determined in the validation section of this paper. Apart from this pos.-percentage measure the function is also able to return polarity (centered around 0). Furthermore, the function is easily adaptable to other measures should those be desired instead of the already available implementations. Finally, each document is assigned a sentiment category (e.g. positive or negative) based on the chosen sentiment measure. If a corpus is expected to contain neutral documents, adding a "dead zone" to the measure might be desirable. Consequently, the output of the first stage of HSentiR consists of category assignments for all documents, which contained at least one word present in the sentiment dictionary. Of course, the percentage of documents assigned a score using this method is a function of document length. However, the movie review corpus used to assess the method shows that for medium length documents, all texts were assigned a score. This can not be expected to be the case for shorter documents. In example, in a corpus of 7,000 tweets with hashtag "#google", 51% were assigned a score. Factoring in both the limited length of tweets and the fact that not all tweets in such a random sample are expected to carry sentiment, this still is considered a solid basis for further classification.

Step2: Machine Learning Based Scoring

Figure 3 gives an overview of the machine learning stage of the process. Typically, the input required to train a machine learning classifier consists of three components. The data is split into training and test sets, additionally the true classifications of the training data is supplied to the algorithm. In the case of HSentiR, an estimation of this true classification is supplied by the DSentiR stage. As noted, there are a number of different algorithms, such as naïve Bayes, k-NN or SVMs, which are known to perform well in text classification tasks. All of these (and more) are already available as R packages and can be utilized with the training data resulting from DSentiR. Choosing a suitable classifier for a given domain is not a trivial task and involves trial and error, i.e. trying a number of different algorithms on a given corpus. Thus, this flexibility is a prerequisite for domain-portability.

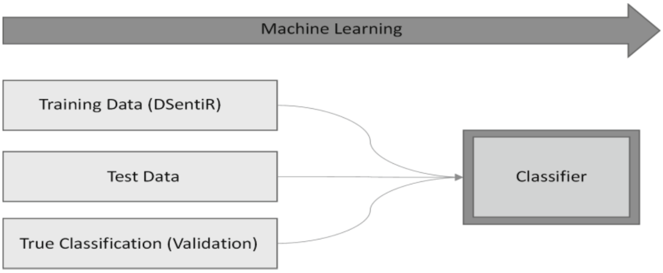

Fig. 3. The machine learning stage of HSentiR

4 Framework Evaluation

In this section, the HSentiR framework will be applied to a corpus of 2,000 movie reviews provided by the well-established movie review corpus 2.0, which has been the basis for over a hundred analyses to this date [8]. This corpus has been the subject of this many studies due to the fact that is has been manually pre-categorized, allowing for reliable process assessment and reliable comparison of different methods. The evaluation consists of three different assessments. First, the dictionary based scoring phase alone will be applied to the corpus, in order to provide an overview of the abilities and limits of this basic scoring method. Consequently, a k-NN classifier will be introduced and trained using the true (true label) training data given by the pre-classified movie-review data. This provides a baseline to compete against for another run of k-NN training using the DSentiR result as training data (estimated label), allowing for a comparison of accuracy within the movie-review domain. As previously discussed, using a domain-appropriate sentiment dictionary is key to dictionary based sentiment classification accuracy. Thus, in this section, different dictionaries will be used to score the documents contained in the movie-review corpus. It is expected that scoring-accuracy varies depending on the used dictionary. In particular, three dictionaries are used:

1: The positive and negative word categories from the current version of the General Inquirer (GI) dictionary [28], as available from the GI-Homepage.
2: The AFINN dictionary, created by [30] for use with form 10-K annual reports, which give an overview of a company's financial situation and its business(domain-inappropriate for movie-reviews).
3: The current version of the dictionary introduced by [31].

As the AFINN dictionary was created for the financial domain, it serves as an example of choosing the wrong dictionary. This should lead to a significant loss of scoring accuracy. The following table shows the classification accuracy for all three dictionary, for both positive and negative reviews, as well as the average across both categories.

Table 1. Percentages of correct sentiment classification within Movie-Review Data

	Positive	Negative	Average
Liu	59.5	77.2	68.35
AFINN	28.6	83.3	55.95
GI	70.2	50.5	60.35

Indeed, the AFINN dictionary results in 5-13% loss of average accuracy. Therefore, choosing the right dictionary for the domain is imperative. The GI has the highest success rate for positive reviews but hardly beats a coin flip for negative ones. Finally, the Liu dictionary seems to be most consistent for this dataset with 68.35% average accuracy. As mentioned before, these results are for positivity with 50% as cut-off value. Although cutting at this threshold seems intuitive, the reasoning behind this value is worth a closer look, as it might not be optimal. How does the cut-off value affect correct scoring in both negative and positive reviews? To answer this question, the percentages of correct scoring are calculated in 0.5% steps for cut-offs ranging from 0.05 to 1, which results in a 200-step distribution of results.

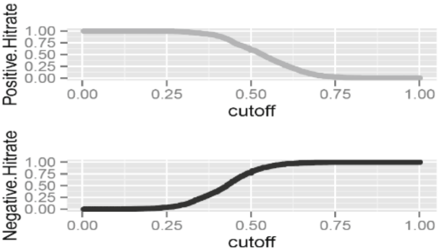

Fig. 4. Distributions of correct scoring for cut-offs 2 (0; 1], 200 steps for positive (upper) and negative (lower) reviews

As expected, all reviews are scored positive for cut-offs near zero and all negative for those near 1. More importantly, both distributions are symmetrical. While the distribution for positive reviews is centered on a value slightly higher than 0.5, the negative case is centered around a value smaller than 0.5. This should balance in the mean of both cases and allow 0.5 to serve as a reasonable cut-off. Figure 5 shows the mean of simultaneous correct scoring for both positive and negative reviews. The maximum of simultaneous correct scoring for both categories is close to 0.5. To be exact, it is found at the cut-off value 50.5% with 70.6% correct scorings, thus providing a 2.25% improvement over the original average.

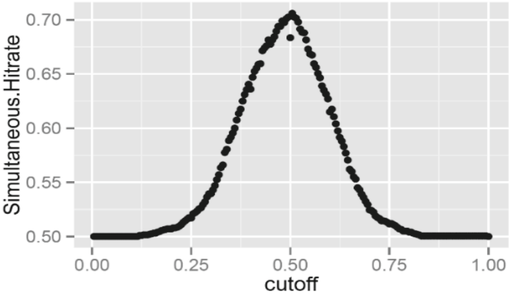

Fig. 5. Distribution of simultaneously correct scores

While this result should not be considered a general truth, it seems that at least in this dataset, positive reviews contain about as many positive matches with the dictionary as negative reviews contain negative matches. Whether this is due to good balancing in the Liu dictionary or a natural property of the dataset would be another interesting question for future research. Of course, such an analysis is only possible with pre-labeled data, which is generally not available when sentiment analysis is desired. However, the example shows that the proposed classification method works and offers up to 70% accuracy, even though the dictionaries were not specifically intended for use in the domain. Next, the accuracy of a k-NN classifier using the predetermined classification of the movie-review dataset is assessed, in order to provide a baseline for the combined scoring approach of HSentiR. The k-NN classifier is only one of the possible classifiers available through various R-packages. Other options include SVMs or advanced methods like string-kernels [29]. A domain-appropriate classifier has to be chosen on a trial and error basis using the data in question. The "class" R-package provides the k-NN implementation used here. Pre-processing in this case includes the removal of all punctuation, white space, lower case conversion, as well as the removal of stop-words. Also, those words in the term-document-matrix, which occur not at least in half the documents are removed (sparseness factor 50%). A random sample of 70% of the data is used to determine the training data for the algorithm. The remaining 30% are used for validation. The table 2 shows the confusion matrix resulting from the process.

Table 2. Confusion Matrix for k-NN classification (true labels)

	Actual	
Prediction	Negative	Positive
Negative	288	69
Positive	21	222

Based on this confusion matrix, the overall accuracy of the k-NN classifier, trained with the actual categories of the data (true label), can be calculated as 85%. Although this scoring accuracy could certainly be improved by using alternate algorithms or fine-tuning of the input-parameters, it is sufficient to act as a benchmark for the hybrid approach, using the same parameterization. In addition, it is important to remember that this real label information would usually not be available in practical research, which is why hybrid approaches, such as the one introduced by the HSentiR framework, are needed in praxis. Using the estimated sentiment-classification, resulting from the DSentiR stage of the process, the same process as before is repeated, yielding the results portrayed in table 2.

Table 3. Confusion Matrix for k-NN (estimated labels)

	Actual	
Prediction	Negative	Positive
Negative	249	123
Positive	66	162

As the new confusion matrix illustrates, the accuracy drops to 68.5% using the estimated label information instead of the true label. Note that, due to the random sampling of the training and prediction cases, the accuracy varies between runs of the model. However, it is reasonably stable on a level comparable to the accuracy of the DSentiR stage. This raises the question of the benefit of the machine learning stage of the process. As described earlier, the machine learning stage allows to classify documents not containing words included in the sentiment dictionary. The answer to this question also has to be answered on a project specific basis, depending on factors, such as the number of documents and their individual length. When very large corpora are analyzed, using the DSentiR stage on a subsample to create training data is computationally preferable, while small corpora can be analyzed entirely using dictionary based scoring. Compared to the k-NN model trained with the real label information, 16.5% accuracy were lost by the label-estimation. Of course, if this is a reasonable price to pay for not having to manually create the training data, is project specific. While it is feasible to create training data manually in datasets like the movie-review example (n=2,000), larger corpora require an automated approach to the problem, like DSentiR. In addition, the k-NN method allows for the classification of documents that do not contain words included in the sentiment dictionary but are otherwise similar to those which do. Due to its modular nature, the HSentiR framework can be applied to both cases. In addition, the increase in scoring accuracy has to be determined on a corpus-specific level and different machine-learning algorithms, such as naïve Bayes or SVMs may increase accuracy even more. Since the goal of this research is to establish a framework for such optimizations, this will not be investigated here because there is no general answer to the question of the most suitable classifier.

5 Conclusions and Outlook

The goal of this research was to create an open, hybrid and domain-portable approach for sentiment classification that meets the requirements of domain-portability and public accessibility, while limiting the level of complexity in order to enable a large amount of users to make use of the process. The evaluation of the two stages of HSentiR shows that the dictionary based stage (DSentiR) performs well if an appropriate sentiment-dictionary is used. This confirms that dictionary based sentiment classification is only as good as the dictionary used to score the texts. The movie-review example shows that a k-NN classifier, when trained with the true classifications of the data, achieves ~85% accuracy in this domain. When using the estimated classifications from DSentiR, the accuracy drops to ~68%. Of course, this level of accuracy leaves room for further process refinements. These can be achieved in three key areas. First, the scoring accuracy of the DSentiR stage should be optimized. There are three possible ways to expand upon the proposed techniques. First, the existing functionality could be made more performant, thus enabling usage on larger data quantities. There are several possible ways to achieve this goal. First, the code could be revised with the goal of vectorization. However, most of the functions used here already comply with this paradigm of R performance. Another way to improve performance could be making use of the existing interfaces between R and other programming languages, such as C++ (rccp) or the C interface that is part of the R-core. Especially the substitution tasks in the cleansplit function could benefit from implementation in those languages. A third possible way to optimize performance is making use of the compiler package and its Just In Time Compiler (JIT), which does not require code revision. The second possible addition to the proposed techniques is extending the existing process to address more of the specific challenges that sentiment analysis faces. An obvious addition would be to make use of the sentiment strength scores available in some sentiment dictionaries. In addition, the pre-processing techniques employed by the cleansplit function could be improved, in example by including a spell-checking and stemming stage to improve the chances of matching a word with the dictionary. Furthermore, the introduction of word sense disambiguation could improve result accuracy. Furthermore, automated translation of corpora and dictionaries could enable cross-language use of the process. Finally, the need for domain-specific dictionaries remains an issue. One approach to solving this problem is using a digital dictionary like WordNet [32] to bootstrap a dictionary for each dataset, using some of the domains most prominent sentiment-laden terms as seeds. Such bootstrapping approaches have been shown to effective [33] and a WordNet interface for R is already available. Apart from achieving scoring-accuracy, this research intended to create a reproducible and open framework for sentiment analysis, which enables researchers to produce peer-reviewable results. The HSentiR process relies on simple R-scripts, which can be shared with both reviewers and the public, ideally making the reproduction of results as easy as pressing as pressing a button. This combination of openness of method and ease of producing results for validation can help the scientific community to build public trust in empirical research. Furthermore, public validation of results can help researchers to correct mistakes, thus improving the quality of future publications. It is with these goals in mind, that the use of methods, such as the HSentiR framework, should be encouraged.

References

1. Speed, J.G.: Do Newspapers now give the News? Forum Fam Plan West Hemisph 15, 704–711 (1893)
2. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. 5, 1–167 (2012)
3. Muhammad, A., Wiratunga, N., Lothian, R., Glassey, R.: Domain-Based Lexicon Enhancement for Sentiment Analysis. In: BCS SGAI SMA 2013 BCS SGAI Work. Soc. Media Anal., pp. 7–18 (2013)
4. Aue, A., Gamon, M.: Customizing Sentiment Classifiers to New Domains: a Case Study. In: Proc. Recent Adv. Nat. Lang. Process RANLP, vol. 49, pp. 207–218 (2005), doi:10.1111/j.1745-3992.1984.tb00758.x
5. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. J. Informetr. 3, 143–157 (2009), doi: 10.1016/j.joi, 01.003
6. Gentleman, R., Temple Lang, D.: Statistical Analyses and Reproducible Research. J. Comput. Graph Stat. 16, 1–23 (2007), doi:10.1198/106186007X178663

7. Peng, R.D.: Reproducible Research in Computational Science. Science 334, 1226–1227 (2011), doi:10.1126/science.1213847
8. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proc. 42nd Annu. Meet. Assoc. Comput. Linguist., p. 271 (2004)
9. Blair-Goldensohn, S., Hannan, K., McDonald, R., et al.: Building a sentiment summarizer for local service reviews. WWW Work. NLP Inf. Explos. Era (2008)
10. Arnold, I.J.M., Vrugt, E.B., Arnold Ivo, J.M., Vrugt Evert, B.: Fundamental uncertainty and stock market volatility. Appl. Financ. Econ. 18, 1425–1440 (2008), doi:10.1080/09603100701857922.
11. Zhang, X., Fuehres, H., Gloor, P.A.: Predicting stock market indicators through twitter "I hope it is not as bad as I fear". Procedia-Social Behav. Sci. 26, 55–62 (2011)
12. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Comput. Intell. 22, 110–125 (2006)
13. Baron, D.P.: Competing for the public through the news media. J. Econ. Manag. Strateg. 14, 339–376 (2005), doi:10.1111/j.1530-9134.2005.00044.x
14. Ludvigson, S.C.: Consumer confidence and consumer spending. J. Econ. Perspect. 18, 29–50 (2004)
15. Steele, S.: Past and irrealis: just what does it all mean? Int. J. Am Linguist. 41, 200–217 (1975)
16. Taboada, M., Brooke, J., Tofiloski, M., et al.: Lexicon-based methods for sentiment analysis. Comput. Linguist. 37, 267–307 (2011)
17. Hatzivassiloglou, V., McKeown Kathleen, R.: Predicting the semantic orientation of adjectives. In: Proc. 35th Annu. Meet. Assoc. Comput. Linguist. Eighth Conf. Eur. Chapter Assoc. Comput. Linguist., pp. 174–181 (1997)
18. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. Inf. Syst. 21, 315–346 (2003)
19. Pang, B., Lee, L., Rd, H., et al.: Thumbs up?: sentiment classification using machine learning techniques. In: Proc. ACL 2002 Conf. Empir. Methods Nat. Lang. Process., vol. 10, pp. 79–86 (2002)
20. Salvetti, F., Reichenbach, C., Lewis, S.: Opinion polarity identification of movie reviews. In: Comput. Attitude Affect Text Theory Appl., pp. 303–316. Springer (2006)
21. Everitt, B.S.: The Cambridge Dictionary of Statistics, 2nd edn. Cambridge University Press, Cambridge (2002)
22. Held, L.: Methoden der statistischen Inferenz. Likelihood und Bayes. Heidelb. Spektrum Akad. Verl. (2008)
23. Mayo Deborah, G., Cox David, R.: Frequentist statistics as a theory of inductive inference. Lect. Notes-Monograph Ser. 77–97 (2006)
24. Read, J., Carroll, J.: Weakly supervised techniques for domain-independent sentiment classification. In: Proc. 1st Int. CIKM Work. Top. Anal. Mass Opin., pp. 45–52 (2009)
25. Li, S., Huang, C.-R., Zhou, G., Lee, S.Y.M.: Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In: Proc. 48th Annu. Meet. Assoc. Comput. Linguist., pp. 414–423 (2010)
26. Berelson, B.: Content analysis in communication research. Society 44, 220 (1952), doi:10.1086/617924
27. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proc. Conf. Hum. Lang. Technol. Empir. Methods Nat. Lang. Process., pp. 347–354 (2005)

28. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilive, D.M.: The General Inquirer. The M.I.T. Press, Cambridge (1966)
29. Feinerer, I., Hornik, K., Meyer, D.: Text Mining Infrastructure in R. J. Stat. Softw. 25, 1–54 (2008), doi: citeulike-article-id:2842334
30. Loughran, T., McDonald, B.: When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. J. Finance 66, 35–65 (2011)
31. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 168–177 (2004)
32. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM 38, 39–41 (1995)
33. Baccianella, S., Esuli, A., Sebastiani, F.: SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proc. Lr. Seventh Int. Conf. Lang. Resour. Eval., pp. 2200–2204 (2008)