Wordify: A Tool for Discovering and Differentiating Consumer Vocabularies

DIRK HOVY
SHIRI MELUMAD
J. JEFFREY INMAN

This work describes and illustrates a free and easy-to-use online text-analysis tool for understanding how consumer word use varies across contexts. The tool, *Wordify*, uses randomized logistic regression (RLR) to identify the words that best discriminate texts drawn from different pre-classified corpora, such as posts written by men versus women, or texts containing mostly negative versus positive valence. We present illustrative examples to show how the tool can be used for such diverse purposes as (1) uncovering the distinctive vocabularies that consumers use when writing reviews on smartphones versus PCs, (2) discovering how the words used in Tweets differ between presumed supporters and opponents of a controversial ad, and (3) expanding the dictionaries of dictionary-based sentiment-measurement tools. *We show empirically that Wordify*'s RLR algorithm performs better at discriminating vocabularies than support vector machines and chi-square selectors, while offering significant advantages in computing time. A discussion is also provided on the use of *Wordify* in conjunction with other text-analysis tools, such as probabilistic topic modeling and sentiment analysis, to gain more profound knowledge of the role of language in consumer behavior.

Keywords: text analysis, natural language processing, language, sentiment analysis

The study of how consumers use words to express thoughts, feelings, and preferences is a topic of growing interest among consumer researchers (Berger et al. 2019; Humphreys and Wang 2018; Krishna and Ahluwalia 2008; Puntoni, de Langhe, and van Osselaer 2009). While work in this area varies widely in its goals, one focus of interest is how consumers choose specific words to convey ideas, and how those word choices affect behavioral outcomes. For example, research has examined how changes in the use of pronouns (e.g., "we," "you," or "I") can alter consumers' views toward service providers and brands (Packard, Moore, and McFerran 2018; Sela, Wheeler, and Sarial-Abi 2012), and how the inclusion of foreign words in ads can increase their persuasiveness among bilingual consumers (Luna and Peracchio 2005). This topic has also gained traction in consumer health care, where gaps in doctors' and patients' vocabularies when discussing illnesses are widely seen as a major barrier to treatment (Koch-Weser, Rudd, and DeJong 2010).

Paralleling the increased interest in the study of word use has been the development of automated text-analysis tools designed to facilitate such analyses. While this literature is diverse in goal and method (e.g., see Berger et al. 2019; Humphreys and Wang 2018 for recent reviews),

Dirk Hovy (dirk.hovy@unibocconi.it) is an associate professor of computer science in marketing at the University of Bocconi, Milan 20136, Italy, and the scientific director of the Data and Marketing Insights research unit (DMI) at the Bocconi Institute for Data Science and Analysis (BIDSA). Shiri Melumad (melumad@wharton.upenn.edu) is an assistant professor of marketing at the University of Pennsylvania, Philadelphia, PA 19146, USA. J. Jeffrey Inman (jinman@katz.pitt.edu) is the Albert Wesley Frey Professor of Marketing at the University of Pittsburgh, Pittsburgh, PA 15260, USA. Please address correspondence to Shiri Melumad. The authors thank Pietro Lesci for his help in the design and implementation of the tool, Jonah Berger and Ashlee Humphreys for their helpful comments, and the Wharton Behavioral Lab for its financial support. Supplementary materials are included in the web appendix accompanying the online version of this article.

Editor: Richard J. Lutz

Associate Editor: Charles F. Hofacker

Advance Access publication March 29, 2021

 • Vol. 48 • 2021
DOI: 10.1093/jcr/ucab018

most text-analysis tools seek to derive measures of the high-level features of texts, such as the topics being discussed (Tirunillai and Tellis 2012; Toubia et al. 2019) or the sentiment being conveyed (Hartmann et al. 2019; Rambocas and Pacheco 2018). While these methods use individual words as their primary inputs, fewer tools within consumer research have been developed wherein words serve as *outputs*. These tools include those that help identify the unique words that men versus women use when discussing the same topic (Johannsen, Hovy, and Søgaard 2015; Joshi et al. 2020), or how situational context affects the vocabularies that consumers use to express emotions of negative versus positive valence (Jurafsky et al. 2014).

The purpose of this article is to address this gap by introducing and illustrating a new, free online tool that allows researchers who lack advanced training in natural language processing to perform such discriminatory vocabulary analyses. The tool, named *Wordify*, uses an approach to supervised machine learning known as randomized logistic regression (Johannsen et al. 2015; Meinshausen and Bühlmann 2010; Shickel et al. 2016) to identify the words in a text that best differentiate between *a priori* categorizations of corpora. These categorizations can either be exogenous to the text, such as the gender of the writer or the device on which a review was written, or endogenous, such as whether texts in a corpus were categorized as more positive or negative based on a sentiment-scoring tool (such as LIWC; Pennebaker et al. 2015). As we illustrate, *Wordify* can be used for various research purposes, ranging from testing hypotheses about how word use differs across consumer segments, identifying the vocabularies used when discussing different products, and expanding the lexicons of commonly used dictionaries. While we focus on English text in this paper, *Wordify* supports analyses in seven additional languages.

Our presentation of *Wordify* is organized in three parts. We begin by describing how *Wordify* differs from other text-analysis tools that may be familiar to consumer researchers and the array of research problems that it can address. We next describe the statistical underpinnings of the method and report empirical comparisons of the randomized logistic regression (RLR) algorithm used by *Wordify* against alternative potential approaches for lexical-feature classification. We then illustrate the range of different possible applications of *Wordify*, including how it can be used:

1. to test hypotheses about differences in word use across contexts.
2. To expand the vocabularies of more conventional dictionary-based sentiment-analysis tools (e.g., LIWC).
3. In conjunction with such dictionary-based tools to provide deeper linguistic insights.

Word Discrimination in Text Analysis

Recent years have witnessed a rapid growth in the development of text-analysis tools in consumer research. While several methodological taxonomies have been offered (Berger et al. 2019; Humphreys and Wang 2018), all of these tools essentially achieve a common goal: to measure or extract one or more key characteristics of a corpus that are of interest to a researcher. For example, a rapid area of growth encompasses methods that can distill the topical content of a corpus. One such method is Topic Modelling, which identifies the most commonly occurring topics in a body of text by analyzing patterns of co-occurring words (see, e.g., Reisenbichler and Reutterer 2019 for a review). The most common implementation is Latent Dirichlet Allocation (LDA; Blei, Ng, and Jordan 2003; e.g., Tirunillai and Tellis 2012), though more performant neural-network-based methods now exist (Bianchi et al. 2021). Perhaps even more familiar are dictionary-scoring tools that quantify holistic features of a text, such as its sentiment (Hartmann et al. 2019; Rambocas and Pacheco 2018), concreteness (Brysbaert, Warriner, and Kuperman 2014; Humphreys, Isaac, and Wang 2020), and emotionality (Rocklage, Rucker, and Nordgren 2018).

In other contexts, however, researchers are interested in uncovering insights about the particular words used in a text. Consider, for example, a researcher who wishes to study whether consumers use different vocabularies when addressing different audiences, such as asking for product information from a trained versus an inexperienced salesperson. One approach might be to separately apply methods of keyword extraction to different samples of queries (Beliga, Meštrović, and Martinčić-Ipšić 2015; Matsuo and Ishizuka 2004; Ohsawa, Benson, and Yachida 1998), which would yield the words that most commonly arise when addressing each type of salesperson when considered separately. This approach, however, would not solve the problem of interest here, that of identifying the words that most *discriminate* between these categorizations—which becomes an even more difficult statistical problem when the number of possible word differences is in the hundreds.

Wordify allows researchers to achieve such discriminant word analyses. In table 1, we summarize the types of research problems that *Wordify* is meant to solve, and how it differs from other methods for vocabulary analysis. This table distinguishes *Wordify* from other approaches that could be used for word-identification purposes. For example, LDA is an unsupervised learning method (akin to factor analysis) that seeks to identify *ex post* the topics that most commonly arise in a corpus of text by identifying groups of words that tend to co-appear in text. *Wordify*, on the other hand, is a supervised learning method (akin to discriminant analysis) for use in cases where the researcher knows the topical classification of a text *a priori* (such as whether reviews written about a restaurant were posted in

2019 vs. 2020) and seeks the words that best discriminate between these classifications. Note that the two methods can be used in tandem, with *Wordify* being used as a confirmatory tool to validate a set of exploratory topics uncovered by a topic model. As another example, *Wordify* provides a novel method for identifying words associated with sentiment expressions that are not captured by standard dictionaries such as LIWC. As illustrated subsequently, given text that has been classified (e.g., by human judges) as conveying positive or negative sentiment, *Wordify* can be used to flag words that are most predictive of those sentiments and lie *outside* published lexicons.

Wordify also offers advantages of ease of use. One of the common barriers to solving more challenging problems in text analysis in consumer research is that available tools often require expertise in statistical programming that is outside the skillset of many researchers. *Wordify* uses machine-learning methods to conduct word discrimination analyses via a simple, free online tool that requires no advanced training in natural language processing. Importantly, this ease comes without sacrificing performance. As we show, *Wordify* achieves discrimination accuracies that are as good or superior to those yielded by more complex statistical tools (such as support vector machines) with much faster computational times.

THE *WORDIFY* ALGORITHM

Wordify solves a statistical problem in word-feature selection from text. Consider, for example, the issue explored by Jurafsky et al. (2014), who were interested in identifying the words that most commonly arise in restaurant reviews with different star ratings. If consumers had only a small vocabulary of words that they used to discuss restaurants, this would represent a multinomial discrimination problem that could be solved by common statistical methods, such as multinomial logistic regression. In this case, each word that arose in a review would be represented as a binary predictive variable. Coefficients would reflect the marginal probability that each word would arise in a review of a given star rating.

The problem, however, is that the number of words (independent variables) that arise in speech can reach several thousand, which are typically sparsely distributed within a corpus. This aspect prohibits efficient maximum likelihood estimation of marginal probabilities for each word. *Wordify* overcomes this problem by using randomized logistic regression (Meinshausen and Bühlmann 2010), a machine learning method that repeatedly fits logistic models to random samples of training sets of words/texts and then applies shrinkage or regularization methods to correct for overfitting of these models. The output of this procedure is the set of words that are best able to discriminate among the outcome categories of interest over thousands of possible regressions. By relying on repeated sampling and randomized regularization, the method approximates an "ideal" world where all possible models are estimated, and the model with the best fit is selected (something that is computationally infeasible).

The literature on classification is one of the largest in statistical machine learning, yielding various methods that could potentially be used to solve word classification problems. For example, chi-square selection, naïve Bayes classifiers (Jurafsky et al. 2014; Monroe, Colaresi, and Quinn 2008), support vector machines, random forests, and feed-forward neural networks (Schmidhuber 2015) could all potentially be used to identify words that best discriminate between different corpora. In building *Wordify*, we used randomized logistic regression (RLR) because of its relative simplicity and known robustness to the constraints that typically arise when analyzing large textual databases. Unlike naïve Bayes classifiers, for example, RLR performs well on the sparse data-design matrices that typically mark texts; because not every term occurs in every document, most cells in the term-document matrix used to represent text are empty (Manning and Schütze 2003). Likewise, logistic regression has advantages in simplicity over feed-forward (deep) neural networks, which are highly sensitive to parameterization and require specialized architecture to be trained efficiently. Logistic regression can be seen as a simplified form of feed-forward network, differing only in the lack of a hidden representation layer, but requiring substantially less data to be fitted. Finally, as we will subsequently show, RLR also has advantages over support vector machines (Cortes and Vapnik 1995; Zhou and Cristea 2016) in computational efficiency when applied to very large corpora, yielding solutions of comparable (or superior) predictive accuracy in a fraction of the computing time.

The *Wordify* algorithm is implemented in a free online software tool (available at: https://wordify.unibocconi.it/) that allows researchers to conduct lexical-feature extractions for any textual dataset containing some discrete *a priori* classification (e.g., Tweets written by male vs. female users). The tool can be used for text-analysis problems that satisfy the following properties:

1. The dependent variables are categorical (not ordinal, continuous, or measured as intervals or ratios);
2. The observations (texts) are independent, that is, they are not repeated measurements or matched data;
3. Words or terms (the independent variables) are not highly correlated;
4. The distribution of words/terms is not extremely skewed toward one category; and
5. The textual dataset is large (a general guideline is at least 10 observations for the least frequent category, and at least 1,500 total observations).

TABLE 1

COMMON WORD-IDENTIFICATION PROBLEMS AND APPROACHES

Research problem	General method	Common statistical approaches	Primary output	Available free software
Find the words that are most representative of a single document	Keyword extraction	Frequency counts, Naïve Bayes, Graph methods	Word list	GenEx, Text Rake
Find the words that best discriminate between documents or classifications of documents	Word discrimination	Chi-square, VSM, RLR, Neural Nets	Word lists	*Wordify*
Find the words that are most common across documents	Text similarity	Latent Semantic Analysis, word and document embeddings	Similarity measures	Word2Vec, Doc2Vec
Jointly identify topics and words in a set of documents	Topic modeling	Latent Semantic Analysis, Latent Dirichlet Allocation	Latent topics	Scikit-learn package in Python, *topicmodels* package in R
Add words to a lexicon	Lexicon expansion	TFIDF	List of *new* words	*Wordify*

While assumptions (2) through (4) are common to most applications of categorical data modeling (e.g., linear discriminant analysis), *Wordify* is free from several other assumptions. It does not, for example, require a linear relationship between the dependent variables (categories) and the independent variables (words); the error terms do not need to be normally distributed; and *Wordify* does not assume homoscedasticity. Although *Wordify* can, in principle, analyze corpora of any size, as with all statistical text-analysis models, the reliability of the results increases with the size of the corpus relative to the number of unique words/terms. Fortunately, because the number of words/terms increases logarithmically with the lines of text being analyzed, results for most text domains (e.g., reviews, social media posts) tend to stabilize once the number of observations in the dataset exceeds roughly 1,500–2,000 (at which point all or most of the possible terms have been observed).

The appendix provides a user's guide to the software that includes more detail regarding data requirements, recommended data-preparation steps, and adjustable parameters. In the web appendix, we describe the technical procedures used to construct the web application, and this information is also available in a publicly viewable repository (https://github.com/MilaNLProc/wordify-webapp). We next describe the statistical basis of the *Wordify* algorithm in greater detail and compare it to other methods. Readers who are less interested in the technical aspects may wish to skip ahead to the next section, where we present a series of applications of use, such as how *Wordify* can be used for hypothesis testing and exploratory analyses of word usage across different *a priori* classifications of interest.

The Technical Underpinnings of *Wordify*

The formal approach is as follows. Assume that we are given a collection $\boldsymbol{D}$ of M documents (called a corpus). To refer to an individual document in this collection, we will use d_i, with i ranging from 1 to M. A document can represent any textual data, from a short Tweet to an entire book. We define a vocabulary $\boldsymbol{V}$ of N terms. To refer to an individual term in the vocabulary, we use v_j, with j ranging from 1 to N. A term can be an individual word (unigram) or a combination of two words (a bigram, e.g., *New York*), three words (or trigram, e.g., *great white shark*), or any number of words (generically called n-gram, where n denotes the number of words in a term). In order to keep the vocabulary to a manageable size, we often omit all common words, such as *I*, *you*, *in*, *the*, *a*, *of*, etc. This set of common words is usually referred to as "stop words" and is both language- and task-dependent (e.g., in a Twitter collection, we might want to include *RT* [for *ReTweet*] in the list of stop words). To represent each document, we use word-count vectors; we map each document d_i into an N-dimensional vector in $\mathbb{Z}^N$, where the position d_{ij} represents the number of times we have seen the term v_j in document d_i. Applying this transformation to all documents in $\boldsymbol{D}$, we arrive at an M-by-N data matrix $\mathbf{X}$.

The goal of *Wordify* is to identify the most representative or indicative words that regularly occur in a given *a priori* category. It technically achieves this by applying a transformation—known as TF-IDF (term frequency–inverse document frequency)—that gives more weight to the raw words that occur "in bursts" (i.e., with elevated frequency in few documents). The rationale is that raw word frequency alone usually cannot sufficiently capture word importance. For instance, most of the stop words that we purposely remove have an extremely high frequency in texts. However, word frequency does carry some information—for example, words that occur only once in the entire dataset would likely yield very little information in discriminating among classifications. In TF-IDF, the raw word frequency d_{ij} (also called term frequency, or TF) is

weighted by the number of documents in which a word occurs (also called its document frequency, or DF). Words that occur in every document have little correlation with the document's categories, whereas words that occur in only a few documents can signal a potential correlation. To compute this weighting factor, *Wordify* uses the inverse of the document frequency or IDF (i.e., 1/DF). By multiplying the two terms together, we get the TF-IDF score of a word. Formally:

$$\begin{gathered}\mathrm{tf}(\mathrm{term},\ \mathrm{doc}) = 1 + \log(\mathrm{tf}),\\ \mathrm{idf}(\mathrm{term}) = 1/\mathrm{df}(\mathrm{w}),\\ \mathrm{TF}-\mathrm{IDF} = \mathrm{tf}(\mathrm{term},\ \mathrm{doc}) * \mathrm{idf}(\mathrm{term}) = \log\ [(1+n)/\left(1+\mathrm{df}(\mathrm{term})\right)] + 1.\end{gathered} \tag{1}$$

In essence, the logarithm in (1) "squashes" the count of very frequent words, reducing their outsized impact in large documents. Words that occur frequently in every document (such as function words) as well as words that occur very rarely throughout the documents receive the lowest TF-IDF scores. In contrast, words that are moderately frequent but occur in only a few documents get the highest score. Our goal is to identify the set of words that receive the highest TF-IDF scores for a given *a priori* text classification (e.g., high or low LIWC positive emotionality scores). Specifically, we wish to fit the general classification model:

$$\mathrm{y} = \mathrm{f}(\mathrm{X};\ \theta,\ \mathrm{b}) \tag{2}$$

where the output value y is a function $f(\cdot)$ of the TF-IDF score input matrix $\mathbf{X}$, parameterized by a set of parameters: θ (depending on the algorithm, θ can be weights, probabilities, or activation functions) and a weighted bias (or error) term b. We use this classification model, with the main difference being that we use the tool to find the subset of words from an entire vocabulary that best predict classifications in a holdout sample.

As noted above, here, we solve this classification problem using a version of randomized logistic regression that selects the words that best discriminate among classes. Briefly, *Wordify* trains the models using a bootstrap sampling approach, fitting hundreds of models on different subsets of the data (sampled with replacement), with varying penalties for overfitting (regularization strengths). Rather than trying to find some specific optimal hyperparameter values for discrimination, the method explores the effect of a large number of possible values on the outcome that are randomly drawn from a uniform range. The underlying algorithm does so by simulating different dataset conditions through two mechanisms: the sub-selection of observations (e.g., Tweets) and variation of the regularization (L1) weight. We then record how often each word from the overall corpus receives a non-zero coefficient under different subsets of observations and L1 weights.

In more formal terms, the algorithm proceeds by first collecting a predefined number K of random subsets of the data (with replacement), fitting a Lasso model on each of these samples with a randomly chosen L1-penalty strength, and collecting the coefficient vectors θ_k of the resulting k models in a coefficient matrix $\boldsymbol{\theta}$. After completion, we compute the relative frequency of all non-zero, positive coefficients for each independent variable v_j, by counting all k coefficients in column j of $\boldsymbol{\theta}$ if $\boldsymbol{\theta}_{k,\ j}>0$ and dividing the result by K. We select variables that receive a non-zero coefficient rate above a threshold set by the user. This threshold (called π_{thr} in Meinshausen and Bühlmann 2010) affects the number of selected variables but not their ranking; additionally, within reasonable bounds (i.e., above a threshold of 0.2), it always yields discriminative features. In its current implementation, the user can specify the inclusion threshold; however, in the applications, we report we used 0.3; that is, a word was identified as being associated with a category if it received a non-zero coefficient weight in over 30% of the k fitted models.[1]

When the target variable is binary, this process is also called Stability Selection (Meinshausen and Bühlmann 2010). However, in the original formulation, *any* non-zero coefficient, independent of sign, was counted, whereas we distinguish between positive and negative non-zero coefficients, which correspond to words that are either highly positively correlated or anti-correlated with the target category. For interpretation, we are mainly interested in the terms that are positively correlated.

In practice, *Wordify* fits 1,000 independent Lasso models on random subsets of the data, each sampled with replacement. For each model, we use a different weight for the L1 regularization parameter λ, controlling how aggressively coefficients are driven to 0.

Illustrative Example

To provide an intuitive example of how *Wordify* works, imagine we have the following five documents with hypothetical descriptions of wines from the United States and Italy listed in table 2 (preprocessed to remove noise words).

Wordify now draws, say, four independent samples from this data, for example: (1,3,4,5), (1,2,2,4), (1,1,2,3), and (2,3,4,4). We fit an L1-regularized Logistic Regression on each, with the United States as target class. This result in the following sparse vectors of coefficients reported in table 3 (indicators that are not present in a run are listed as 0 here):

We can now count for each indicator how many times out of the four runs it received a non-zero coefficient (the

1 The implementation in a previous version of the sklearn Python library suggested 0.25 as default, but we increase it slightly to reduce the number of selected variables to a more manageable size.

magnitude does not matter). We distinguish by positive and negative coefficients, and divide the result by the number of runs (here, four), which yields the final indicators that are positively and negatively correlated with the US wines:

The results of table 4 suggest that a wine is likely to be from the United States if its description contains any of the following words: "buttery," "chardonnay," "heavy" or "light," or "spice," and these words are similarly discriminative. In contrast, a wine is likely to *not* be from the United States if it contains the words "spice" or "cherry." It is also worth noting that "oak" and "wine," which were present for both Italian and US wines, were ultimately not selected as discriminative indicators of US wines. Finally, we would conduct an analogous analysis with Italy as the target class to determine which indicators are most and least discriminative of Italian wines.

Empirical Comparison to Alternative Methods

As noted above, RLR is but one of several statistical approaches that an analyst might use to identify the words that are most associated with different classifications of text. We utilize RLR in part because of its speed and ease of use, which allow *Wordify* to be freely accessible and usable for researchers without technical training in natural language processing. That said, ease of use is, of course, only advantageous if it does not come at the cost of accuracy. This section addresses this concern by reporting the results of a comparative analysis of *Wordify* against other possible classifiers within two contexts. One is a numerical simulation in which we study comparative accuracy in recovering words known *a priori* to be distinctly associated with different classifications, using synthetic data that allow us to control the parameters. The second compares these same approaches using a real-world dataset, which illustrates how the insights derived from *Wordify* would compare to those yielded by other tools in actual research settings.

In both analyses, we compare *Wordify* to two benchmark classification approaches for feature extraction: Chi-square variable selection (Alexandrov, Alexander, and Lozovoi 2001) and randomized support vector machine classification (Cortes and Vapnik 1995; Zhou and Cristea 2016). While other comparative classifiers could be considered, we saw these as conceptually bracketing RLR's approach. Specifically, the chi-square classifier illustrates a naïve approach to classification: it computes the relative frequencies with which a word appears in each category in a document and compares it with the value that would be expected if words were randomly assigned. While the chi-square value is thus easy to compute for each word, it is a univariate rather than multivariate approach to classification (i.e., akin to univariate vs. multivariate regression, it computes the discriminatory power of each word independently, not in the presence of others). It does not explicitly protect against chance associations between words and categories.

The other approach is randomized support vector machines (SVM), a more sophisticated non-parametric multivariate approach to classification (Cortes and Vapnik 1995). Whereas logistic regression seeks to define a hyperplane that maximizes the posterior probability that a word is a member of a given category, SVM seeks to define a boundary (hyperplane) that maximally separates the words most associated with one category versus another. As such, while it could potentially be a more flexible approach to classification than RLR, it comes at the cost of a considerable increase in computational complexity (Salazar, Velez, and Salazar 2012).

Method comparisons using simulated data. The goal of the simulation was to compare the performance of *Wordify* against chi-square and SVM classifiers along two dimensions: (1) their ability to recover a "true" vocabulary associated with a given binary classification (i.e., classification performance), and (2) their computation time (i.e., classification efficiency). Assessing classification performance for the tool is of obvious importance, as ease of use is only an asset if it does not come at the cost of a loss of performance relative to other methods. Computation time is relevant when evaluating machine-learning methods because it affects usability and it is of particular concern with SVMs, which can be difficult to apply to problems of extremely high dimensionality (e.g., a corpus with thousands of words and documents). More specifically, the computation time of an algorithm (which is a function of how many operations are necessary in order to fit a model to the data) depends on the number of observations and number of words.[2] For logistic regression, computation time increases roughly linearly with the number of observations and words, such that when the number of observations and/or words is doubled, it will take twice as long to fit the data. In contrast, for SVMs, computation time is cubic in the number of the observations and quadratic for the words, such that doubling the number of observations and/or words will take the algorithm exponentially longer to fit.[3] Note that for the chi-square algorithm, concerns about computation time do not arise, as the approach involves a series of simple linear calculations of the chi-square statistic for each word; namely, the ratio of the relative frequency of each word within each category to its expected frequency under chance assignment.

To compare the classification accuracy and speed of *Wordify* against the two alternate methods, we examined

2 For further details on the time complexity of classification algorithms, see Mohri, Rostamizadeh, and Talwalkar (2018).

3 The reason SVM takes exponentially longer is because the objective function of the SVM involves both computing a kernel matrix (i.e., the similarity of all instances with each other) and solving a quadratic function.

TABLE 2

DESCRIPTIONS OF WINES FROM THE UNITED STATES AND ITALY

	Text	Label
1	Spice light wine	Italy
2	Wine oak heavy	United States
3	Chardonnay buttery light	United States
4	Wine light cherry	Italy
5	Chardonnay wine oak buttery	United States

TABLE 3

COEFFICIENTS FOR FREQUENCY OF INDICATORS IN EACH OF THE FOUR RUNS FOR US WINES

Model	Buttery	Chardonnay	Cherry	Heavy	Light	Oak	Spice	Wine
1	0.32	3.78	−2.49	0	0	0	−2.49	0
2	0	0	0	3.62	−4.38	0	0	0
3	0	0	0	0	0	0	−6.2	0
4	0	0	−6.2	0	0	0	0	0

TABLE 4

FINAL SET OF INDICATORS THAT ARE POSITIVELY VERSUS NEGATIVELY CORRELATED WITH US WINES

Coefficient valence	Buttery	Chardonnay	Cherry	Heavy	Light	Oak	Spice	Wine
Positive	0.25	0.25	0	0.25	0.25	0	0	0
Negative	0	0	0.5	0	0	0	0.5	0

the performance of each method under each of two problem-size conditions simulating small and large corpuses. The small condition simulated a corpus that contained 2,500 documents with 300 unique words and two labels, while the large simulated the case of 5,000 documents with 2,000 unique words and three labels. This allowed us to compare the ability of each method to correctly identify the words that best discriminated between the synthetic categories (i.e., classification accuracy), after which we compared their computation times. The simulations involved four steps:

1. *Vocabulary generation*. Either 300 or 2,000 unique integer variables were generated by randomly sampling (without replacement) from a 1–100 uniform distribution. These integer variables constituted the universe of "words" used in the simulation.
2. *Category assignment*. The vocabulary words were then assigned to two or three synthetic categories. One part was assigned to each of the categories (labeled "0," "1," or "2"), and one part to all categories (these are "distractor words").
3. *Document generation*. Either 2,500 or 5,000 simulated documents (observations) were created, each having between 10 and 80 (for the small dataset) or 100 words (for the larger dataset). Specifically, a document was first created and randomly assigned to a category ("1" or "0" for small; "0," "1," or "2" for the larger condition). Its word count was then randomly drawn from a uniform distribution ranging from 10 to the maximum number of words. Given the particular word count drawn, a number of words were then sampled (with replacement) from the conditional distribution P (word|category) created in step 2. This includes both true indicator words and distractors.
4. *Replication and performance estimation*. Ten independent runs of this data-generation process were then undertaken, and performance scores for each of the three methods were computed. For each method, the average over 10 runs defined its performance for a given dictionary-size condition (smaller vs. larger).

In the simulation, we used the Python sklearn package implementations of both the Logistic Regression (used in *Wordify*) and the SVM with a linear kernel.[4] We used the SVM's default settings, for example, a tolerance of 0.001 for the stopping criterion, but with L1 regularization. To allow for an "apples-to-apples" comparison between RLR and SVM, a randomized version of SVM was programmed such that, like *Wordify*, it fit 1,000 SVM models to random subsets of the data for each run.[5]

Results for simulated data: Classification accuracy and efficiency. Since we generated the data, we knew the true correspondence between words and categories. This allowed us to compare the three methods in terms of (1) how many of the true indicators were detected ("recall"), and (2) how many of the indicators were correctly assigned to categories ("precision"). Note that if a method simply returns all words, it logically receives a "perfect recall," but its precision would be poor. To capture this tradeoff, we compute the harmonic mean of precision and recall, or the F1 score. A good method has a high F1 score (i.e., closer to 1.0).

The results of this analysis are provided in table 5, which reports for each method the average precision, recall, and F1 (as well as computation time, discussed subsequently). First, the analysis suggests that *Wordify*'s RLR offers the

4 While the sklearn package supports the estimation of SVM modes with non-linear kernels, to allow a comparison with RLR, we require a method that produces a coefficient vector for each dependent variable value. The only SVM basis function that allows for this is the linear SVM.

5 We also compared RLR to a newer version of SVM called "Thunder SVM" (Wen et al. 2018); however, Thunder SVM did not offer improvements in computation time and, because it does not support L1 (Lasso) regularization to aid in variable selection, it yielded an excessive rate of false-positive predictors. To illustrate, in our small data simulation it was almost 20 times slower than RLR (63 vs. 3.2 seconds), and had poorer precision (0.3 vs. 1.0). While recall was the same, F1 was 0.49 (ThunderSVM) versus 0.98 (RLR).

best balance between recall and precision among the three methods in recovering the words most associated with the two simulated categories. Because, by definition, the chi-square classifier yields a classification value for each word in each corpus (it has complete coverage), it displays perfect recall for both the large- and small-corpora problems. In contrast, it displays the poorest precision, resulting in the lowest F1 scores among the methods. SVM performs almost as well as RLR in the small-corpora problem (with lower precision but somewhat higher recall). However, its performance degrades severely when applied to the larger problem,[6] with SVM displaying both low precision and low recall, resulting in an F1 score only slightly higher than that provided by the chi-square classifier. In contrast, *Wordify*'s RLR still performs well for the larger problem, yielding an F1 score only modestly less than that observed in the smaller problem.

Finally, the analysis reveals a marked difference in the computational efficiency of the methods. As reported in table 5, while the chi-square selector has the poorest accuracy, its simplicity allows for extremely fast computational times—on an average less than 1 second per run. At the other extreme was SVM, which took, on an average, 8 minutes and 21 seconds to derive a solution when applied to the larger classification problem with 5,000 documents and 2,000 terms. Given that real-world text-analysis problems tend to be much larger than this—often involving tens of thousands of documents or observations (e.g., Tweets or reviews) and thousands of words/terms—such a result raises concerns about the pragmatic viability of SVM as an approach to word classification (as we elaborate on next). In contrast, no such problem arises for *Wordify*, which, while naturally slower than a simple chi-square selector, is able to solve classification problems in a fraction of the time required by SVM (e.g., for the larger problem, *Wordify* took 16 seconds while SVM took 501 seconds).

Method comparisons using field data. While the simulation results provide encouraging support for the ability of *Wordify*'s RLR to identify the words most associated with text classifications, the evidence came from a synthetic environment in which the true association pattern was known. In natural settings, of course, there is no known set of "correct" indicator terms against which the methods can be compared, such that method comparison takes a somewhat different form. In such contexts, if a researcher seeks to test a hypothesis about how word use varies across certain categories by using *Wordify*, s/he would require assurance that s/he would not have reached different conclusions had a different classifier been used.

To address this issue, we undertook a re-analysis of a corpus of TripAdvisor reviews reported in study 1 of Melumad, Inman, and Pham (2019). The corpus consisted of 61,642 reviews written about restaurants in San Francisco, CA, and Philadelphia, PA, between 2014 and 2017. In that work, the authors tested the hypothesis that reviews written on smartphones tend to be more selectively emotional than those written on personal computers. The authors found support for this hypothesis, showing, for example, that relative to PC-generated reviews, smartphone-generated reviews had a greater proportion of emotional words based on LIWC's "affect" category, and also had higher ratings of emotionality as appraised by human judges. The authors did not, however, examine whether reviews written on smartphones versus PCs actually involved distinct vocabularies that might mirror the observed difference in degrees of emotionality. *Wordify* provides us with this opportunity.

To examine this question, we subjected the TripAdvisor reviews written across devices to analysis by the same three algorithms described above: Wordify (RLR), chi-square, and SVM. While the methods cannot be compared on precision or recall (since here the "truth" is unknown), they can be compared in terms of computing time. We anticipated that because of the large nature of the database—61,635 reviews (documents/observations), 6,348 words/terms—SVM would have difficulty achieving convergence in this setting. This was indeed the case: whereas the RLR took 12 minutes and 13 seconds to fit, SVM took 11 days, 2 hours, and 40 minutes.

In table 6, we report the 50 top words identified by each method, respectively, as being most associated with smartphone- versus PC-generated reviews. For the chi-square method, the table reports the words with the highest chi-square values in discriminating between reviews written on smartphones versus PCs. Since RLR and SVM fit separate models for smartphone- and PC-generated reviews, the table reports the percentage of models in which a given word emerged as a significant predictor of a review being written on a smartphone versus PC (listed in descending order of frequency). Note that because we had to use L2 (Ridge) rather than L1 (Lasso) regularization in the SVM, it produces a larger, less selective vocabulary, with the top 50 words it identified appearing in more than 99% of all randomized models.[7]

A visual examination of the table reveals that all three methods yield words that are consistent with Melumad et al.'s (2019) hypothesis that smartphone-generated reviews tend to be more emotionally evaluative than those generated on PCs. Among the most prominent words in smartphone-generated reviews, for example, were "amazing" (identified by all three methods),

6 We note that this comparability in low-dimensionality problems is similar to that reported in prior simulation comparisons (Salazar et al. 2012).

7 Because Lasso selection in SVM yielded no discriminating words, Ridge regularization was used in estimation.

TABLE 5

SIMULATION COMPARISONS OF THE PERFORMANCE OF CHI-SQUARE, RLR *(WORDIFY)*, AND SVM CLASSIFIERS FOR SMALL- AND LARGE-CORPORA PROBLEMS

Method	Precision	Recall	F1	Comp. time (seconds)
Smaller problem (300 words, 2,500 documents)				
Chi^2	0.36	1.00	0.54	1
RLR (Wordify)	1.00	0.92	0.96	3
SVM	0.54	0.97	0.70	33
Larger problem (2,000 words, 5,000 documents)				
Chi2	0.24	1.00	0.18	1
RLR (Wordify)	1.00	0.57	0.72	16
SVM	0.18	0.33	0.24	501

"phenomenal," and "awesome" (identified by RLR and SVM). PC-generated reviews, in contrast, tended to be more descriptive of the restaurant experience, marked by words such as "restaurant," "dining," and "meal."

To analyze the outputs more systematically, we subjected the top 50 words identified by each method to analysis by human judges as well as two dictionary-based tools: LIWC (Pennebaker et al. 2015) and the Evaluative Lexicon (Rocklage et al. 2018). Specifically, following Melumad et al. (2019), we used LIWC's "affect" score—that is, the percentage of words in a text categorized by LIWC's dictionary as affective—as a measure of the degree of emotionality contained in the reviews. The Evaluative Lexicon (EL) additionally provided the average degree of emotionality as appraised by human judges for each word in the texts contained in the EL dictionary. A natural limitation of both approaches is that the measures are constrained by the scope of their dictionaries, which omitted many of the words used in the reviews; the Evaluative Lexicon's dictionary, for example, contained only 17% of the words identified by the classifiers as most discriminating between smartphone- and PC-generated reviews. To address this, we recruited 500 participants from Amazon Mechanical Turk (MTurk) to rate the emotionality of each of the top 50 words identified by one of the three randomly chosen methods (*Wordify*, SVM, or chi-square) as most discriminative of reviews written on smartphones and PCs (on a scale of 1: "Not at all emotional" to 7: "Very emotional").

The results of these analyses, reported in table 7, suggest that the three methods provide convergent support for Melumad et al.'s (2019) findings that reviews written on smartphones tend to be more emotional than those written on PCs. When averaged across methods, the words used in reviews written on smartphones were rated by human judges as significantly more emotional than those written on PCs ($M_{\text{Smartphone}}$=4.35 vs. M_{PC}=3.89; $F(1, 19{,}496)=85.19$, $p<.001$), and these words received higher emotionality scores based on both LIWC's affect measure ($M_{\text{Smartphone}}$=24.47 vs. M_{PC}=10.51; $F(1, 250)=8.24$, $p=.005$) and the Evaluative Lexicon's emotionality score ($M_{\text{Smartphone}}$=1.62 vs. M_{PC}=.44; $F(1, 229)=6.50$, $p=.011$).

In terms of the methods, the results confirmed no interaction between type of classifier and device on the emotionality of the words for each of the measures (human judgments: $F_{\text{classifier*device}}<1$; LIWC: $F_{\text{classifier*device}}(1, 250)=1.19$, $p=.277$; Evaluative Lexicon: $F_{\text{classifier*device}}<1$). That is, regardless of the method used for word identification, researchers would draw the same conclusion; here, that smartphone-generated reviews tend to be more emotional than those written on PCs. We might note, however, that when one compares the 50 words identified by *Wordify* with those identified by SVM, the former display a greater *difference* in emotionality between smartphones and PCs (across all three measures)—which implies that *Wordify* was able to identify the words that were the sharpest discriminators between the two devices in terms of emotionality. Chi-square, while useful for performance-oriented variable selection, is less useful for deriving specific insights from corpora, as it does not yield distinct words for each category.

ILLUSTRATIVE APPLICATIONS

Wordify is a tool designed to aid researchers in studying how the particular words used in textual data vary across different contexts of interest. These contexts could be exogenous to the textual corpus, such as how the words used in reviews written on smartphones differ from those written on PCs (as discussed earlier), or how the vocabulary used in wine reviews differs across varietals. Likewise, *Wordify* can be used to examine differences in contexts endogenous to a dataset, such as observing how word use differs across posts with high versus low positive emotionality as categorized by LIWC or across topics derived from LDA. As such, we see *Wordify* as having four major domains of application:

1. A *hypothesis testing* tool that allows researchers to test predictions about differences in word use across contexts (e.g., different consumer segments or devices used to generate content);
2. A *vocabulary discovery* tool that allows researchers to uncover the words that differentiate discussions of certain categories (e.g., different brands of a product or types of service experiences);
3. An *augmentation* tool that allows researchers to gain insights into the particular words that underlie automated classifications of text (e.g., the words that drive the sentiment scores provided by LIWC or topics identified by LDA); and
4. A *dictionary expansion* tool that allows users to expand the capabilities of dictionary-based text-analysis tools (e.g., to enhance the accuracy of LIWC sentiment scores in predicting human judgments of sentiment).

TABLE 6

WORDS MOST ASSOCIATED WITH SMARTPHONE- VERSUS PC-GENERATED REVIEWS AS IDENTIFIED BY *WORDIFY* (RLR), CHI-SQUARE, AND SVM CLASSIFICATION METHODS (FOR *WORDIFY* AND SVM, WORDS ARE RANKED BY FREQUENCY IN RANDOMIZED MODELS; FOR CHI-SQUARE, WORDS ARE RANKED BY CHI-SQUARE VALUE

Wordify (RLR)				Support vector				Chi square	
Mobile		PC		Mobile		PC		Mobile	
Word	Frequency	Word	Frequency	Word	Frequency	Word	Frequency	Word	χ^2
Amazing	0.59	dining	0.59	advisor	1	building	1	amazing	18.26
Recommend	0.59	restaurant	0.59	amazing	1	cook just	1	awesome	17.02
awesome	0.56	time	0.56	awesome	1	day	1	dining	12.16
delicious	0.53	meal	0.53	beautiful	1	dining	1	recommend	11.26
friendly	0.52	hotel	0.52	come	1	dislike	1	room	10.54
nice	0.52	wonderful	0.52	eggplant appetizer	1	eat outside	1	tonight	10.54
definitely	0.51	eat	0.51	family	1	entre	1	restaurant	10.16
come	0.5	not	0.5	greet	1	family run	1	hotel	9.98
family	0.48	room	0.48	look amazing	1	folk	1	concetta	9.89
thank	0.48	table	0.48	phenomenal	1	helpful not	1	definitely recommend	9.17
great	0.47	menu	0.47	recommend	1	lunch	1	amazing food	8.99
really good	0.43	special	0.43	thank	1	make point	1	pizza	8.78
visit	0.43	view	0.43	today	1	meal	1	definitely come	8.49
beautiful	0.42	area	0.42	tonight	1	menu time	1	trip advisor	8.25
greet	0.42	bit	0.42	trip advisor	1	people	1	awesome food	7.94
ok	0.42	day	0.42	visit	1	pudding	1	advisor	7.91
phenomenal	0.42	just	0.42	avocado	0.99	roast	1	highly recommend	7.85
spot	0.42	know	0.42	bring	0.99	san francisco	1	eggplant appetizer	7.77
star	0.42	lunch	0.42	business dinner	0.99	special	1	fiance	7.28
today	0.42	reservation	0.42	chilly	0.99	staff great	1	building	6.83
tonight	0.42	think	0.42	clay	0.99	think	1	table	6.76
trip advisor	0.42	year	0.42	delicious	0.99	turn	1	menu	6.64
authentic	0.41	building	0.41	fillet	0.99	twice	1	interesting	6.55
concetta	0.41	dinner	0.41	great choice	0.99	view	1	wonderful	6.45
fry	0.41	evening	0.41	jalapeã	0.99	winter	1	highly	6.43
oyster	0.41	excellent	0.41	jalapeno	0.99	breakfast	0.99	like family	6.43
salmon	0.41	interesting	0.41	little bit	0.99	constantly	0.99	definitely	6.41
starter	0.41	wait staff	0.41	min	0.99	davio	0.99	thank	6.35
avocado	0.4	breakfast	0.4	nice	0.99	dead	0.99	time	5.97
fillet	0.4	dish	0.4	not include	0.99	decade	0.99	fine	5.94
flavor	0.4	experience	0.4	quality	0.99	dessert share	0.99	preparation	5.83
hubby	0.4	item	0.4	really good	0.99	feel good	0.99	item	5.72
overall	0.4	people	0.4	recommendation	0.99	fritter	0.99	look amazing	5.66
tiramisu	0.4	pricey	0.4	saison	0.99	good staff	0.99	recently	5.65
unbelievable	0.4	street	0.4	salmon	0.99	gyro	0.99	special occasion	5.51
beer	0.39	ambiance	0.39	spot	0.99	haight	0.99	meal	5.5
boyfriend	0.39	café	0.39	starter	0.99	item	0.99	mac	5.46
bring	0.39	city	0.39	tiramisu	0.99	know	0.99	quite	5.38

TABLE 6 (*CONTINUED*)

Wordify (RLR)				Support vector				Chi square	
Mobile		PC		Mobile		PC		Mobile	
business dinner	0.39	crowd	0.39	butternut squash	0.98	manner	0.99	amazing staff	5.37
definitely come	0.39	décor	0.39	course good	0.98	menu limit	0.99	fast service	5.33
eggplant appetizer	0.39	early	0.39	restaurant friend	0.98	place come	0.99	year	5.27
fantastic	0.39	expensive	0.39	star	0.98	preparation	0.99	wait staff	5.25
filet	0.39	fun	0.39	friendly	0.98	recently	0.99	friendly staff	5.22
forget	0.39	noisy	0.39	portobello	0.98	room	0.99	jalapeno	5.17
good view	0.39	quite	0.39	sized portion	0.98	seven hill	0.99	reservation	5.16
knowledgable	0.39	recently	0.39	communal	0.98	share dessert	0.99	special	5.11
little bit	0.39	roast	0.39	definitely come	0.98	small room	0.99	think	5.02
not miss	0.39	san francisco	0.39	refill	0.98	tableside	0.99	occasion	5
recommendation	0.39	turn	0.39	business trip	0.98	usually	0.99	friendly	4.98
rude	0.39	wine	0.39	flavor	0.98	wait staff	0.99	theater	4.95
cool	0.38	preparation	0.38	philly cheese	0.98	sauteed	0.99	know	4.95

Yellow-shaded cells indicate words that were found by two or more methods to be distinctive of mobile; green-shaded cells identify corresponding words distinctive of PC).

As an illustration of this last application, imagine that a researcher wants to construct a story that manipulates whether participants are in a positive versus negative mood. To do this, a seed dictionary (such as LIWC) might be used to construct an initial scenario (e.g., a story with a greater proportion of positive vs. negative words), after which words provided by *Wordify*—for example, the words identified as most diagnostic of positivity versus negativity—might be added in to enrich this description.[8]

In this section, we describe in greater detail how *Wordify* can be used in these different capacities. Above, we showed how *Wordify* can be used for hypothesis testing, that is, determining how the use of language might differ between smartphone- and PC-generated reviews. We next illustrate how *Wordify* can be applied to the three remaining domains of use. We first show how *Wordify* can be used to discover vocabularies: here, recovering the language that experts use when discussing different wine varietals (application 1). We next demonstrate how *Wordify* can serve as a complement to sentiment-analysis tools (e.g., LIWC) to gain deeper insights into how supporters versus opponents of an ad campaign express their views on social media (application 2). In the final application, we show how *Wordify* can be used to expand the vocabulary of dictionary-based tools (such as LIWC) in order to improve the predictive performance of bag-of-words tools (application 3).

In each of these applications, we first prepared the original texts for analysis by removing duplicate entries and collapsing different forms of the same word into one form (e.g., we represent the words "go," "goes," "going," "gone," and "went" simply as "go," that is, its dictionary entry. This process is called "lemmatization"). We also removed common stop words and extracted counts for all combinations of one to three words (i.e., unigrams, bigrams, and trigrams) that occur in at least 0.1% of the documents in each dataset (this removes rare words like misspellings or user names), but in no more than 75% of all documents (since such frequent words are not correlated with any one DV and therefore have little to no discriminative power).[9] We then used the TF-IDF transformation described earlier to reduce the influence of common words and ran *Wordify* on the resulting data. Each run of the algorithm fit 1,000 independent models and kept all words and phrases that received a positive coefficient in at least 30% of all models.[10]

8 We thank an anonymous reviewer for this suggested application.

9 We follow common guidelines from text analysis for these thresholds (Hovy 2020; Manning and Schütze 2003).

10 While *Wordify* allows the user to select any percentage threshold of inclusion, we used 30% because it offers a compromise between a list of words that is reasonably exhaustive yet strongly discriminates between categories.

TABLE 7

HUMAN AND AUTOMATED ASSESSMENTS OF EMOTIONALITY IN SMARTPHONE- VERSUS PC-GENERATED REVIEWS BY METHOD

	Wordify			SVM			Chi square
	Mobile	PC	Difference	Mobile	PC	Difference	Mobile
Human (Emotionality)	4.47	3.84	0.63	4.29	3.92	0.37	4.29
LIWC (Affect)	28.81	9.26	19.55	18.46	11.76	6.7	26.15
Eval.Lexicon (Emotionality)	1.31	0.41	0.9	0.98	0.46	0.52	1.19

Application 1: *Wordify* as a Tool for Vocabulary Discovery

One potential application of *Wordify* is to gain descriptive insights about the vocabularies used by consumers to discuss different *a priori* topics of interest. Here, we illustrate this functionality by using *Wordify* to analyze a large corpus of expert wine reviews to uncover the words that professional sommeliers use most commonly to describe different varietals of wine. This particular setting is useful because there are well-established nomenclatures for describing the features of different varietals (Hogson 2009; Lehrer 2009), which can serve as a basis for a comparison with the words identified by *Wordify*. Our goal is thus to investigate the ability of the tool to uncover the essential features of this nomenclature, as well as potentially more novel words and word patterns overlooked in previously established wine vocabularies.

Note that an alternative approach to uncovering wine vocabularies might be to consider reviews for different kinds of wines independently, and then identify the most frequently used words (relative to a general background corpus) using a keyword or word-feature extraction tool (Matsuo and Ishizuka 2004). The weakness of this approach, however, is that because the vocabularies of varietals may differ in subtle ways, it would not establish the uniqueness of each vocabulary (i.e., the words that maximally *differentiate* the nomenclatures for different wines). *Wordify* is focused on achieving that goal.

The words used to describe wines are notoriously diverse and often metaphorical, designed to capture the wide array of aromas and flavors of wines that stem from differences in grapes, growing regions and climates, and even years of production (Hogson 2009; Humphreys and Carpenter 2018; Lehrer 2009). To explore the ability of *Wordify* to recover such nomenclature, we analyzed a corpus of more than 150 000 wine descriptions. These were written by sommeliers and originally published in reviews on the WineEnthusiast website, and the data were made publicly available on Kaggle.com (https://www.kaggle.com/zynicide/wine-reviews). In addition to the sommeliers' reviews, the data included the origin of the wine being described (i.e., country, province, region, winery), the grape varietal, price, and rating. For illustrative purposes, we focused our analysis on differences in language used to describe varietals. For reporting convenience, we removed all entries that lacked a label for varietal—resulting in 137,227 entries and a vocabulary of 3,969 terms—and restricted our investigation to varietal labels that occurred at least 1,000 times in our analysis. This resulted in seven varietals: Chardonnay, Malbec, Petite Shiraz, Riesling, Syrah, Tempranillo, and Viognier.

To analyze the data, we first assigned each review a label that indicated whether it was one of the seven varietals. We then used *Wordify* to identify the words that had the most non-zero coefficients with respect to each varietal. We report the results of this analysis in table 8, listing the 132 terms that were significant positive predictors of each wine varietal in 30% or more of the 1,000 randomized regression models.

To provide a benchmark of comparison, in table 9, we reproduce the features of each varietal as listed on *WineFolly.com*, a commercial site that offers novice wine drinkers basic knowledge of different wines. Note that because these vocabularies come from two distinct sources—one an automated analysis of reviews (*Wordify*) and the other from a specialized website about wine (*WineFolly.com*)—we would not necessarily expect a strong correspondence between the two. Despite this, a comparison of the two tables shows that the linguistic profiles of each varietal derived by *Wordify* aligned reasonably well with those provided by *WineFolly.com*'s descriptions of varietals. Consistent with this benchmark vocabulary, for example, citrus flavors and buttery and creamy textures were found by *Wordify* to be highly predictive of a review describing a Chardonnay, and use of the word "petrol" was found to be highly predictive of whether the wine being reviewed was a Riesling, along with other classic associations (e.g., apple flavor and high acidity/mineral flavoring). Likewise, words found by *Wordify* as being predictive of whether a review was of a Syrah—such as the flavors of bacon and blackberry and dark coloring—are also classic features of the wine according to the reference vocabulary.

Note that *Wordify* allows for different wines to share common features/descriptors; for example, "apricot" is a flavor that sommeliers use to describe a number of white wines (Rieslings, Chardonnays, and Viogniers). Table 8

TABLE 8

WORDS IDENTIFIED BY *WORDIFY* AS MOST PREDICTIVE OF DIFFERENT WINE VARIETALS IN EXPERT REVIEWS

Wordify terms									
Riesling		Malbec	Syrah		Chardonnay		Tempranillo	Viognier	Shiraz
acidity	sweet	berry	acid	smoke	acidity	oak	berry	apricot	plum
alcohol	sweetness	black	bacon	steak	apricot	oaky	black fruit	citrus	vanilla
apple	tangerine	black fruit	blackberry	tannic	banana	orange	earthy	honeysuckle	bodied
apricot	tea	blackberry	beef	tart	butter	peach	plum	peach	drink
citrus	balance	dark	berry	violet	buttered	popcorn	raspberry	tropical	finish
dry	concentrated	plum	blueberry	new	butterscotch	richness	red fruit		hint
dry style	delicate	aroma	boysenberry	vineyard	buttery	steely	aroma		texture
floral	drink	good	chocolate		caramel	structure	feel		
honey	finish	nose	coffee		citrus	toasty	finish		
lemon	hint	palate	earth		cream	tropical	nose		
lime	intensely		espresso		creamy	tropical fruit	palate		
mineral	linger		fruit		custard	unoaked			
minerality	long		meat		lemon	vanilla			
minerally	note		meaty		mango	white fruit			
peach	palate		pepper		mineral	wood			
petrol	perfume		peppery		minerality	yellow			
slate	scent		sappy		minerally	yellow fruit			
steely	style		scent		nut	vineyard			
stone									
sugar									

TABLE 9

VOCABULARIES USED TO DESCRIBE DIFFERENT WINE VARIETALS ON *WINEFOLLY.COM*

Wine Folly Terms							
Riesling	Malbec	Syrah	Chardonnay		Tempranillo	Viognier	Shiraz
apple	black cherry	allspice	almond	pineapple	cedar	honeysuckle	black pepper
apricot	black pepper	bacon fat	apple	praline	cherry	mango	blueberry
beeswax	blackberry	blackberry	apple blossom	vanilla	clove	peach	chocolate
citrus blossom	blueberry	blueberry	baked tart	vanilla bean	dill	rose	plum
diesel fuel	chocolate	boysenberry	beeswax	wet flint rocks	dried fig	tangerine	
ginger	cocoa	chocolate	butter		leather		
high acidity	coconut	clove	caramel		plum		
honey	coffee	cured meat	celery leaf		tobacco		
honeycomb	dill	herbs	citrus peel		tomato		
lanolin	gravel	licorice	coconut		vanilla		
lime	green stem	mint	creme brulee				
Meyer lemon	leather	olive	dill				
nectarine	milk chocolate	pepper	fig				
peach	mocha	rosemary	honeysuckle				
pear	molasses	smoke	jackfruit				
petrol	plum	tart	jasmine				
pineapple	pomegranate	tobacco	lemon				
rubber	raisin	vanilla	lemon balm				
	raspberry		lemon zest				
	tobacco		oaky				
	vanilla		passionfruit				
			peach				
			pear				

also illustrates how the words produced by *Wordify* are subject to the stemming decisions made by users; for example, the table reveals "acidity" and "acid" emerging as separate discriminative terms, with the former commonly used to describe Rieslings and the latter commonly used to describe Syrahs.

In addition to these comparative results, the tables reveal *Wordify*'s ability to uncover the differences that exist in

the *breadth* of vocabularies across varietals. For example, Chardonnay is the second best-selling wine in the United States (U.S.A. Wine Ratings 2019), and it has a wide variety of aromas, flavors, and textures that stem from differences in growing regions and production methods (e.g., how it is barreled). Reflective of this, the vocabulary identified by *Wordify* as being predictive of a Chardonnay is a highly diverse one—again, consistent with that found in the benchmark wine vocabulary (table 9). Likewise, the algorithm captures the more limited vocabularies that have evolved to describe Tempranillos, Viogniers, and Petite Shirazes, which are less popular within the United States and thus, unsurprisingly, have fewer words used to typically describe them.

Application 2: *Wordify* as an Augmentation Tool

In both the method-comparison analyses and application 1 discussed earlier, the categorizations under study were exogenous to the words being analyzed—they specified how word usage varies across reviews of different kinds of originating devices and different wine varietals (respectively). In some cases, however, *Wordify* could be used *in conjunction* with topic modeling or sentiment-analysis tools to provide insights into the words that drive *endogenous* classifications arising from such tools. For example, *Wordify* could be used to identify the particular words that best discriminate between high and low LIWC positive or negative emotionality scores. Here, we provide such an analysis for the words used in social media posts about a controversial ad campaign that contained mostly positive versus mostly negative emotional words (presumably marking supporters versus opponents of the ad).

The context is Nike's 2018 "Just Do It" campaign, featuring former NFL quarterback Colin Kaepernick. This particular campaign is of interest because of its polarizing nature: Kaepernick is a player who knelt during the National Anthem at the start of games in the 2016 season, a controversial act of protest that some felt contributed to his being unemployed by the league in 2017 and 2018. The ad featured a close-up of the athlete's face with the accompanying text, "Believe in something. Even if it means sacrificing everything." The ad triggered a wide range of emotional responses that were posted on social media, ranging from intense support to calls for boycotts of the Nike brand (Jennings 2018; Stillman 2018; Wang and Siegel 2018).

Naturally, we expected that supporters more frequently use positive words, and opponents more frequently use negative ones. Our interest in analyzing the words used in social media posts about the ad, therefore, lies more in the nature of the words that *accompany* such expressions of affect. That is, we examine whether supporters and opponents appear to be targeting their sentiment toward the same (or different) elements of the ad. Such an analysis can also be used in applications, such as ad testing and new product testing.

Our data were a corpus of 5,086 original Tweets associated with the #Justdoit hashtag that were posted on September 7, 2018 (Dabbas 2018; https://www.kaggle.com/eliasdabbas/5000-justdoit-tweets-dataset). The dataset included the text of each Tweet, as well as the user's username, number of friends (i.e., users they follow), and number of followers. The first required step in the analysis was to classify the sentiment being conveyed in a Tweet to determine whether the user was more likely a supporter or opponent of the ad. While several sentiment-analysis tools could be used for this purpose (Rambocas and Pacheco 2018), for illustrative purposes, we used the popular dictionary tool LIWC (Pennebaker et al. 2015). LIWC was useful for this illustration due to its pervasiveness in consumer research (Berger and Milkman 2012; Ludwig et al. 2013; Melumad et al. 2019) and its ability to provide separate measures of the percentage of negative ("negemo") and positive ("posemo") words that appeared in a Tweet (dimensions that might co-exist in the same Tweet). To prepare the data for analysis within *Wordify*, we classified each Tweet as being either above or below the mean negemo or posemo scores for the entire corpus of Tweets.

The results of this *Wordify* analysis are presented in table 10, where we partition the most common words that most discriminate among the categories into two groups: those words that *do* appear in the LIWC vocabulary of negative or positive emotionality, and those that lie outside. Words that lie outside the LIWC vocabulary are of interest for two reasons: first, they shed light on language differences that exist beyond the scope afforded by LIWC's dictionary; second, they potentially provide some insight into the *substantive* focus of emotional expression. For example, were expressions of negativity mostly in reference to specific aspects of the campaign or to the surrounding political issues?

The table yields a number of insights into how consumers expressed their negative and positive sentiment about the ads. First, of the 37 words that most commonly arose in Tweets containing high LIWC negative-emotionality scores, 15 (40%) were outside the LIWC dictionary. This set of words external to LIWC's dictionary tended to include substantive references either to the protagonist in the ad ("Kaepernick"), the cause that the player was protesting ("police"), or patriotism ("flag"). Of the negative words identified that *were* included in LIWC's dictionary, most tended to accompany these specific references (e.g., "hater," "brutality," "cut," and "savage"). In contrast, of the 29 words identified as most commonly arising in Tweets with high *positive* emotionality scores, none were outside the LIWC dictionary. Among them, the most commonly used words in highly positive Tweets included "fantastic," "thank," "support," "bless," and "encourage."

These findings suggest that while both negative and positive emotions were expressed, there was a notable

TABLE 10

WORDS MOST PREDICTIVE OF TWEETS ABOVE THE MEAN ON LIWC NEGATIVE AND POSITIVE EMOTIONALITY FOR NIKE'S 2018 *JUST DO IT* CAMPAIGN, AND ASSOCIATED LEVELS OF CONCRETENESS ("NONE" INDICATES THAT THERE WERE NO WORDS IDENTIFIED THAT WERE NOT CONTAINED IN THE LIWC POSITIVE EMOTIONALITY DICTIONARY)

High negative emotionality			High positive emotionality		
Words within LIWC dictionary		Words outside LIWC dictionary	Words within LIWC dictionary		Words outside LIWC dictionary
crazy	brutality	police	thank	definitely	None*
Idiot	cut	company	great	impressive	
Bitch	seriously	make	support	bless	
fear	threaten	hat	hilarious	enjoy	
stupid	outrage	people	super	important	
savage	lie	Don	positive	okay	
worry	tear	work	lmao	welcome	
fake		Kaepernick	winner	joke	
Hater		flag	greatness	challenge	
ignorant		face	strong	passion	
afraid		job	ok	supporter	
scream		man	proudly	encourage	
upset		national	fantastic	safety	
Suck		message	pretty	like	
Fail		amp	better		
Brysbaert Concreteness Score=3.276 MRC Concreteness Score=410.312			Brysbaert Concreteness Score=2.315 MRC Concreteness Score=335.137		

difference in how the ad was being discussed by the two groups of Tweeters: more concretely by opponents, more abstractly by supporters. Specifically, expressions of negative sentiment were accompanied by references to more concrete elements of the ad itself (e.g., "Kaepernick," "police"). On the other hand, expressions of positivity referred more to abstract, holistic expressions of support (e.g., "like," "fantastic") without concrete elements. To confirm this, we subjected the Tweets to analysis by two dictionary-based tools that provide measures of the degree of concreteness of the words: the Brysbaert score (Brysbaert et al. 2014), which is based on a dictionary of 40,000 words rated for concreteness, and the MRC score (Paetzold and Specia 2016), based on an analogous lexicon of 8,228 words; both calculations were provided by the concreteness-scoring tool developed by Humphreys et al. (2020). As shown in table 10, the results confirm the pattern suggested by the *Wordify* results: The words identified by *Wordify* as most predictive of negative emotionality received higher concreteness scores than those most predictive of positive emotionality.

In sum, while LIWC was helpful in categorizing the Tweets in terms of sentiment, *Wordify* provided deeper insight into the potential motivations underlying expressions of positivity versus negativity toward the ad.

Application 3: *Wordify* as a Dictionary-Expansion Tool

In our final application, we demonstrate an additional domain of potential usage of *Wordify:* a tool for the expansion of linguistic dictionaries commonly used in automated text analysis. One of the well-known limitations of dictionary tools (such as LIWC) is that their insights are restricted to the scope of their lexicons. For example, LIWC measures the extent of negative emotionality in a corpus by computing the relative frequency of any one of 744 words contained in the LIWC negative emotionality dictionary (Pennebaker et al. 2015). While this is expansive, application 2 showed how negative expressions of emotion may often be conveyed by words that take on negative meanings only in specific contexts (e.g., "police") and hence lie outside standard dictionaries. While these non-dictionary words are useful in providing insights into the context in which negative emotionality is being expressed, they could also be used to *refine* the sentiment tool itself by expanding its lexicon.

To illustrate how *Wordify* can be used to refine the lexicons of bag-of-words sentiment-measurement tools, we used a corpus of more than 12,000 Tweets about e-sports games, collected as part of a class project. These Tweets were retrieved using the Tweepy library, which queried the Twitter API for Tweets mentioning e-sports games. The 12,000 Tweets were then randomly assigned to eighteen judges who were asked to classify the content as positive, negative, or neutral. Most Tweets were reviewed by a single judge, but about 5% were reviewed by multiple judges. An independent research assistant then removed Tweets that contained irrelevant content (not related to games) or were simply URLs, leaving a total of 9,771 unique Tweets. The total size of the vocabulary is 1,642 distinct terms.

We first input the texts into *Wordify* to identify the most frequently occurring words in content classified as "positive" by human judges, and then did the same to identify the most common words for content judged as "negative." We excluded neutral words from the analysis since there is no corresponding LIWC category. Next, we used the words identified by the *Wordify* analysis to estimate two separate predictive models for each of the two human-judged sentiment classifications (positive and negative).[11] One classification model used as independent variables the subset of positive (or negative) *Wordify*-identified words that were also included in LIWC's "positive emotionality" (or "negative emotionality") vocabulary, and the dependent variable was whether the texts were rated by judges as positive (or negative). For the second classification model, we used as independent variables *all* words identified by *Wordify*—those within *and* outside LIWC's dictionary—as most frequently occurring in positive (negative) texts and, again, used as a dependent variable whether the texts were judged as positive (negative).

In order to quantify these differences, we used five-fold cross-validation (Hartmann et al. 2019). The data were split into five equal-sized parts, and a model was fitted on four of these parts and then evaluated on the fifth part. By iterating five times over the data to make each subset the target of the evaluation, we ended up with five performance scores. Averaging over these provides a robust estimate of the predictive performance of the model on the hold-out data. As a performance metric, as in our simulation, we use macro-F1, the harmonic mean between the precision and recall of the model. We evaluated the statistical significance among the predictive performance scores of the different conditions using a non-parametric bootstrap-sampling test with 10,000 iterations. This test is commonly used in text-classification tasks (Berg-Kirkpatrick, Burkett, and Klein 2012; Søgaard et al. 2014).

The differences in predictive performance highlight the drawback of relying exclusively on a predefined set of dictionary words. For example, if we predict human-judged negative sentiment using only the subset of 43 *Wordify*-identified negative words that are included in the LIWC negative emotionality dictionary, we get a five-fold cross-validation macro-F1 score of 0.67 (the best possible score is 1.0). If instead, we use the full set of 69 words identified by *Wordify* as most discriminative of negativity—including also terms outside LIWC's dictionary—to predict human-judged negativity, we get an F1 score of 0.74. This difference is not only significant ($p<.001$ from nonparametric bootstrap test) but also it represents a 10% increase in F1 score.

When predicting positive sentiment, LIWC ran into issues by misclassifying words like "war" and "kill" as negative (in the context of e-games, these often refer to positive events). Consequently, the difference is even more striking. Using the subset of 84 *Wordify*-identified positive words contained in LIWC's "positive emotionality" dictionary, we get a performance of 0.49 F1 score. In contrast, using the full set of 109 words identified by *Wordify* as most discriminative of positivity, we get a score of 0.60. Again, this difference is statistically significant at $p < .001$ and represents a 22% improvement in score. These results illustrate the value of expanding a dictionary beyond the words contained in LIWC to the words specific to the domain of interest—a benefit that *Wordify* can provide.

As a final caveat, we should emphasize that expanding dictionaries with *Wordify* will enhance predictive performance only if the domain or target of the prediction is largely similar to that used for word identification and model fitting. If used in too different a context—for example, using *Wordify*-identified positive and negative words for Tweets about e-sports to predict the valence of Tweets about a TV—then expanding the dictionary might not help. In fact, it might degrade predictive performance, since it focuses the model too much on the original domain (here, e-sports). However, if the focus of prediction can be assumed to be reasonably similar, then using the tool for dictionary expansion can provide meaningful value.

DISCUSSION

Recent years have witnessed a rapid growth of interest in the development and use of automated text-analysis tools. In this paper, we contribute to this stream of work by describing and illustrating a new, easy-to-apply method for identifying the words that are most associated with different *a priori* classifications of text. The tool differs in its purpose and ease-of-use from the large catalog of existing text-analysis methods with which some researchers may be familiar. Whereas the most common goal of text analysis is to extract the higher-level meaning of words in a corpus—such as categorizing texts as more versus less positive—*Wordify* tries to solve what might be seen as the reciprocal problem: given *a priori* categorizations of text, what specific words most discriminate one category from the other? By being available as a free, easily usable online tool, it allows researchers to solve this problem with modern machine-learning methods—methods that might otherwise be beyond the reach of scholars who lack training in more advanced natural language processing techniques (Sebastiani 2002).

11 We used L2-regularized Logistic Regression models over TF-IDF vectorized presentations as classifiers. These are similar to the model in *Wordify*, except for the regularization. However, in *Wordify*, we use Logistic Regression with L1 regularization (Lasso), which encourages sparse coefficient vectors, so we can select discriminative terms. In prediction, we want the model to spread predictive mass across all available indicators, so we use Logistic Regression with L2 regularization (Ridge), which generalizes better to unseen data and has better predictive performance.

Range of Applications

In this paper, we illustrated the types of research contexts in which *Wordify* can be useful. Perhaps the most widely applicable for consumer researchers is use of the tool for testing hypotheses of how word usage differs between contexts. We showed, for example, how *Wordify* can uncover differences in the vocabularies that consumers use to write restaurant reviews on their smartphone versus PC, or differences in the words used by experts to describe distinct wine varietals. We might note that while the analyses reported here measured vocabularies at a given point in time, the tool can likewise be used to measure changes in vocabulary use over time, such as on firms' websites or Facebook pages. We also showed how the tool can be used in conjunction with dictionary-based sentiment analysis tools (such as LIWC). After categorizing texts as predominantly positive or negative in affect, for example, *Wordify* can be used to provide deeper insights into the particular words most associated with each type of text. When applied to an analysis of Tweets about Nike's 2018 Colin Kaepernick "Just Do It" campaign (application 2), for example, we showed that Tweets classified by LIWC as more negatively emotional tended to reference substantively different topics than those categorized as positively emotional; Tweets categorized as more negatively emotional often referenced more concrete topics, targeting specific aspects of the ad or the surrounding political issue, whereas those categorized as more positively emotional tended to make more abstract references, such as expressions of general liking and support. Finally, we showed how *Wordify* could be used to expand the indicator sets of "bag-of-words" tools for sentiment analysis to obtain higher descriptive validities. For instance, the list of words identified by *Wordify* as most discriminative of positive and negative Tweets about e-sports was more expansive than the list derived using LIWC's predefined dictionary of words alone (application 3).

We note that the above applications also illustrate another dimension of how the tool can be used. While *Wordify* can serve as a standalone method for vocabulary discovery—our application to wine varietals is one such example—*Wordify* is also well-suited for use in conjunction with other text-analysis methods. In particular, *Wordify* can be used as the first of a two-step process for analyzing differences in consumer vocabularies across contexts. After *Wordify* identifies the vocabularies that differ across contexts (e.g., Tweets about different political candidates), another tool can be used to characterize these vocabularies on dimensions of interest (e.g., sentiment, style, topic). The reported study of language differences between smartphone- and PC-generated reviews serves as an illustration of this. We first used *Wordify* to identify the specific words that most differentiated reviews across devices, and then used a combination of human judges and dictionary tools (LIWC and the Evaluative Lexicon) to quantify differences in the emotionality of these words. Similarly, in the analysis of consumer responses to Nike's Kaepernick ad, we used *Wordify* to identify the words that most differentiated between positive and negative Tweets, and then scored the words using the concreteness-scoring tool developed by Humphreys et al. (2020).

Limitations

The use of *Wordify* comes with a set of caveats and cautions. First, because the goal of *Wordify* is to uncover differences in word usage across two (or more) categories, it serves a different function than keyword extraction tools, which are designed to identify the words that are most distinctive (relative to a background corpus) in a single document. Likewise, if one were interested in testing for differences in summary measures (e.g., sentiment) between two contexts—rather than the words that gave rise to those measures—then the use of *Wordify* would be unnecessary. Note, though, that researchers using such methods must trust that summarization tools are able to provide accurate measures of the constructs they are interested in; for example, when testing whether men (vs. women) use more positive language in movie reviews, one must trust that LIWC's "positive emotionality" scores actually capture positivity within that context. In such cases, *Wordify* can serve a useful function by giving researchers an easy way to "look under the hood" and see the individual words that are driving the sentiment measure—an insight generally not provided by most popular applications (e.g., LIWC).

Wordify also comes with technical caveats. Perhaps most important, because *Wordify* is agnostic to the meaning of words, vocabularies may occasionally include words that are only incidentally associated with an outcome of interest. For example, in our analysis of wine reviews, one of the most predictive words of a review being a Riesling or a Petite Shiraz was "drink"—a word that just happened to be used more in reviews of those two varietals but is clearly not unique to the normative vocabularies for those wines over others. It is important to emphasize, however, that this issue is a generic one arising in almost all text-modeling approaches (e.g., LDA). As is the case with LDA, we recommend that prior to using *Wordify*, users strip text of words that might strongly co-occur with a category but would be of limited scholarly interest.

Future Directions for *Wordify*

As discussed earlier, the current implementation of *Wordify* uses randomized logistic regression because it offers distinct efficiency advantages over other extant methods (such as support vector machines) when applied to problems of word-variable selection. That said, it is important to note that the field of computational methods in

natural language processing is a rapidly changing one that witnesses constant improvements in the speed and accuracy of algorithms. It is for this reason that *Wordify* was designed with a modular architecture that allows for continual upgrading of the classification algorithm if and when faster or more accurate inference engines become available. That is, we emphasize that *Wordify* is agnostic to the statistical classification that is used: as long as a tool supports L1 regularization, it could be a viable alternative to RLR if it offers improvements in speed and accuracy. With this in mind, we are closely following recent computational advances in support vector machines (Wen et al. 2018) and other algorithms, to explore whether they might offer a useful means for improving the tool.

Likewise, in future work, we will explore the feasibility of strengthening the inference engine underlying *Wordify* through the use of a suite of learning models rather than one algorithm (Lee, Hosanagar, and Nair 2018; Netzer, Lemaire, and Herzenstein 2019).[12] While, as noted, the randomized logistic regression method that drives *Wordify* outperforms other single modeling alternatives (specifically SVMs with L1 regularization and chi-square classifiers) in speed and accuracy, intuition suggests that a suite of tools working in concert would perform better than one working alone.

As a final note, one of the challenges in developing text-analysis tools is maintaining a balance between technical sophistication and usability. While machine-learning algorithms have been found to provide measures of sentiment that predict human assessment to a much higher degree than relatively naïve dictionary tools (Hartmann et al. 2019), their adoption has been tempered by the absence of easily usable turnkey software. It was with an eye toward achieving a balance between technical sophistication and usability that *Wordify* was developed. While the technology underlying the software may be complex, it can be used by any researcher armed simply with textual data and a classification scheme. Our hope is that *Wordify* can be seen as an illustration of how the researcher's text-analysis toolkit might evolve going forward, marked by a constellation of methods tailored for different purposes, and coupled with software that makes their use accessible to a wide breadth of researchers.

DATA COLLECTION INFORMATION

Five studies are reported in the paper. The numerical simulation was designed and analyzed by the first author. The data for the TripAdvisor reviews, Nike Tweets, Wine Spectator Reviews, and E-Sports Tweets were all drawn from public sources as described in the manuscript. The TripAdvisor and Nike Tweets were jointly analyzed by the first and second authors. The Wine Spectator and E-Sports data were analyzed by the first author. Data are currently stored in a project directory on the Open Science Framework.

12 We thank an anonymous reviewer for raising this suggestion.

APPENDIX

Accessing and Using *Wordify* online

A fully functional beta version of the Wordify algorithm can be accessed at https://wordify.unibocconi.it/index (temporary user password 0981). Using the platform is straightforward:

DATA REQUIREMENTS

Wordify is a tool that can be used to identify the words that best discriminate between two or more *a priori* categories of text. *Wordify* currently supports texts:

- English
- German
- Dutch
- Spanish
- French
- Portuguese
- Italian
- Greek
- Danish
- Japanese
- Lithuanian
- Norwegian
- Polish
- Romanian
- Russian
- Chinese

It does not, however, support multi-language texts, therefore your texts should be in one language.

For reliable results, we suggest providing at least 2,000 lines of text. If you provide less, we will still *Wordify* your file, but caution should be taken in analyzing/interpreting the results. The approach assumes that the texts being analyzed satisfy these additional criteria:

a. The observations (texts) are independent, that is, they are not repeated measurements or matched data;
b. Words or terms (the independent variables) are not highly correlated; and
c. The distribution of words/terms is not extremely skewed toward one category.

STEP 1: PREPARE YOUR DATA

a. **Data cleaning.** We strongly recommend that users remove unwanted "word noise" from the text prior to analysis. While not required for application, data cleaning helps ensure more reliable and

interpretable results. A number of easy-to-use automated cleaning tools are available, though all require some rudimentary knowledge of programming. The most common is the Natural Language Toolkit (NLTK), which is written in the Python programming language and freely available. We recommend two cleaning steps:

i. *Lemmatization:* If possible, remove duplicate entries and collapse different forms of the same word into one form (e.g., we represent the words "go," "goes," "going," "gone," and "went" simply as "go").
ii. *ii. Remove stop words:* Words that are rare or too common (called "stop words") typically have low discriminatory power. In our applications, we focus the analysis on all combinations of one to three words (i.e., unigrams, bigrams, and trigrams) that occur in at least 0.1% of the documents in each dataset (this removes rare words like misspellings or usernames), but in no more than 75% of all documents (since such frequent words are not correlated with any one dependent variable and therefore have little to no discriminative power).

b. **Data formatting.** Create an Excel (.xlxs) file with two columns. The first should be labeled **"text"** and contain all your documents (e.g., Tweets, reviews, patents, etc.), one per line. The second column should be labeled **"label"** and contain the dependent-variable label associated with each text (e.g., rating, author gender, company, etc.).

Once you have prepared your Excel file, open *Wordify*, click the "Get started" button, and scroll down to the file-upload screen below:

STEP 2: CHOOSE A THRESHOLD FOR WORDS TO INCLUDE IN THE OUTPUT (DEFAULT IS 30%)

Wordify works by estimating a large number of models that use different subsets of words in your text to discriminate between categories. The output is a list of words that consistently emerge as significant predictors in these models. Because words that are rarely used (e.g., appear in less than 5% of models) probably reflect noise, we report only those that are significant in a certain percentage of models. The default threshold is 30%, which we have found offers a good compromise between coverage and accuracy (the original paper on Stability Selection gives little guidance but generally suggests higher thresholds, which can be too restrictive and return no results). However, you can change this threshold to be more (or less) than 30% if you want to be more (or less) selective.

STEP 3: UPLOAD YOUR FILE

Locate the file that you wish to have analyzed, choose the language of your texts, and provide your email address.

STEP 4: CHECK YOUR EMAIL FOR THE RESULTS

We will process your data, and you will receive your *Wordified* file by email. Depending on the number of requests, it can take up to 30 minutes (but usually 3–4 minutes are enough). No data are stored on our server.

Below is a sample input and output file:

REFERENCES

Alexandrov, Mikhail, Gelbukh Alexander, and George Lozovoi (2001), "Chi-Square Classifier for Document Categorization," in *Computational Linguistics and Intelligent Text Processing, CICLing 2001*. Lecture Notes in Computer Science, Vol. 2004, ed. A. Gelbukh, Berlin, Heidelberg: Springer.

Beliga, Slobodan, Ana Meštrović, and Sandra Martinčić-Ipšić (2015), "An Overview of Graph-Based Keyword Extraction Methods and Approaches," *Journal of Information and Organizational Sciences*, 39 (1), 1–20.

Berger, Jonah, Ashley Humphreys, Stephan Ludwig, Wendy Moe, Oded Netzer, and David Schweidel. (2019), "Uniting the Tribes: Using Text for Marketing Insights," *Journal of Marketing*, 84 (1), 1–25.

Berger, Jonah, and Katherine L., Milkman (2012), "What Makes Online Content Go Viral," *Journal of Marketing Research*, 49 (2), 192–205.

Berg-Kirkpatrick, Taylor, David Burkett, and Dan. Klein (2012), "An Empirical Investigation of Statistical Significance in NLP," in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Jeju Island, Korea, 995–1005.

Bianchi, Federico, Silvia Terragni, Dirk Hovy, Debora Nozza, Dand Elisabetta Fersini. (2021), "Cross-Lingual Contextualized Topic Models with Zero-Shot Learning," in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Kiev, Ukraine.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003), "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, 3, 993–1022.

Brysbaert, Marc, Amy Beth Warriner, and Victor Kuperman (2014), "Concreteness Ratings for 40 Thousand Generally Known English Word Lemmas," *Behavior Research Methods*, 46 (3), 904–11.

Cortes, Corinna, and Vladimir Vapnik (1995), "Support-Vector Networks," *Machine Learning*, 20 (3), 273–97.

Dabbas, Elias. (2018), "5000 'Just Do It' Tweets Data Set," www.kaggle.com/eliasdabbas/ [last accessed December 2019].

Hartmann, Jochen, Juliana Huppertz, Christina Schamp, and Mark Heitmann (2019), "Comparing Automated Text

Classification Methods," *International Journal of Research in Marketing*, 36 (1), 20–38.

Hogson, Charles. (2009), *History of Wine Words: An Intoxicating Dictionary of Etymology and Word Histories of Wine, Vine, and Grape from the Vineyard, Glass, and Bottle*. Canada: P2peak Press.

Hovy, Dirk. (2020), *Text Analysis in Python for Social Scientists: Discovery and Exploration*. Cambridge, UK: Cambridge University Press.

Humphreys, Ashlee, and Gregory S. Carpenter (2018), "Status Games: Market Driving through Social Influence in the U. S. Wine Industry," *Journal of Marketing*, 82 (5), 141–59.

Humphreys, Ashlee, and Rebecca Jen-Hui Wang (2018), "Automated Text Analysis for Consumer Research," *Journal of Consumer Research*, 44 (6), 1274–306.

Humphreys, Ashlee, Isaac Matthew, and Wang Jen-Hui (2020), "Construal Matching in Online Search: Applying Text Analysis to Illuminate the Consumer Decision Journey," *Journal of Marketing Research*, September. doi: 10.1177/0022243720940693.

Jennings, Rebecca. (2018), "How Nike's Colin Kaepernick Ad Explains Branding in the Trump Era," https://www.vox.com/2018/9/4/17818222/nike-colin-kaepernick-ad [last accessed November 2019].

Johannsen, Anders, Dirk Hovy, and Anders Søgaard. (2015), "Cross-lingual Syntactic Variation over Age and Gender," in Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China, July 2015, 103–12.

Joshi, Priyanka D., Cheryl J. Wakslak, Gil Appel, and Laura Huang (2020), "Gender Differences in Communicative Abstraction," *Journal of Personality and Social Psychology*, 118 (3), 417–35.

Jurafsky, Dan, Victor Chahuneau, Bryan R. Routledge, and Noah A. Smith (2014), "Narrative Framing of Consumer Sentiment in Online Restaurant Reviews," *First Monday*, 19 (4).

Koch-Weser, Susan, Rima E. Rudd, and William DeJong (2010), "Quantifying Word Use to Study Health Literacy in Doctor–Patient Communication," *Journal of Health Communication*, 15 (6), 590–602.

Krishna, Aradhna, and Rohini Ahluwalia (2008), "Language Choice in Advertising to Bilinguals: Asymmetric Effects for Multinationals versus Local Firms," *Journal of Consumer Research*, 35 (4), 692–705.

Lee, Dokyun, Kartik Hosanagar, and Harikesh S. Nair (2018), "Advertising Content and Consumer Engagement on Social Media: Evidence from Facebook," *Management Science*, 64 (11), 5105–31.

Lehrer, Adrienne. (2009), *Wine and Conversation*, 2nd ed. New York, NY: Oxford University Press.

Ludwig, Stephan, Ko de Ruyter, Mike Friedman, Elisabeth C. Brüggen, Martin Wetzels, and Gerard Pfann (2013), "More than Words: The Influence of Affective Content and Linguistic Style Matches in Online Reviews on Conversion Rates," *Journal of Marketing*, 77 (January), 87–103.

Luna, David, and Laura A. Peracchio (2005), "Advertising to Bilingual Consumers: The Impact of Code-Switching and Language Schemas on Persuasion," *Journal of Consumer Research*, 31 (4), 760–5.

Manning, Christopher D. and Hinrich Schütze (2003), *Foundations of Natural Language Processing*, Cambridge, MA: MIT Press.

Matsuo, Yutaka and Mitsuru Ishizuka. (2004), "Keyword Extraction from a Single Document Using Word Co-occurrence Statistical Information," *International Journal on Artificial Intelligence Tools*, 13, 157–69.

Meinshausen, Nicolai, and Peter Bühlmann (2010), "Stability Selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (4), 417–73.

Melumad, Shiri, J. Jeffrey Inman, and Michel Tuan Pham (2019), "Selectively Emotional: How Smartphone Use Changes User-Generated Content," *Journal of Marketing Research*, 56 (2), 259–75.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018), *Foundations of Machine Learning*, Cambridge, MA: MIT Press.

Monroe, Burt L., Michael P. Colaresi, and Kevin M. Quinn (2008), "Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict," *Political Analysis*, 16 (4), 372–403.

Netzer, Oded, Alain Lemaire, and Michal Herzenstein (2019), "When Words Sweat: Identifying Signals for Loan Default in the Text of Loan Applications," *Journal of Marketing Research*, 56 (6), 960–80.

Ohsawa, Yukio, Nels E. Benson, and Masahiko Yachida. (1998), "KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor," in Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98, Beijing, China, 12–18.

Packard, Grant, Sarah G. Moore, and Brent McFerran (2018), "(I'm)Happy to Help (You): the Impact of Personal Pronoun Use in Customer–Firm Interactions," *Journal of Marketing Research*, 55 (4), 541–55.

Paetzold, Gustavo and Lucia Specia. (2016), "Inferring Psycholinguistic Properties of Words," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Santa Barbara, CA, 435–40.

Pennebaker, James W., Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. (2015), *The Development and Psychometric Properties of LIWC 2015*, Austin, TX: University of Texas at Austin.

Puntoni, Stefano, Bart de Langhe, and Stijn van Osselaer. (2009), "Bilingualism and the Emotional Intensity of Advertising Language," *Journal of Consumer Research*, 35, 1012–25.

Rambocas, Meena, and Barney G. Pacheco (2018), "Online Sentiment Analysis in Marketing Research: A Review," *Journal of Research in Interactive Marketing*, 12 (2), 146–63.

Reisenbichler, Martin, and Thomas Reutterer (2019), "Topic Modeling in Marketing: Recent Advances and Research Opportunities," *Journal of Business Economics*, 89 (3), 327–56.

Rocklage, Matthew D., Derek D. Rucker, and Loran F. Nordgren (2018), "The Evaluative Lexicon 2.0: The Measurement of Emotionality, Extremity, and Valence in Language," *Behavior Research Methods*, 50 (4), 1327–44.

Salazar, Diego A., Jorge I. Velez, and Juan C. Salazar (2012), "Comparison between SVM and Logistic Regression: Which One is Better to Discriminate?," *Revista Colombiana de Estadistica Numero especial en Bioestadistica*, 35 (SPE2), 223–37.

Schmidhuber, Jürgen (2015), "Deep Learning in Neural Networks: An Overview," *Neural Networks : The Official*

Journal of the International Neural Network Society, 61, 85–117.

Sebastiani, Fabrizio (2002), "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, 34 (1), 1–47.

Sela, Aner, S. Christian Wheeler, and Gülen Sarial-Abi (2012), "We Are Not the Same as You and I: Causal Effects of Minor Language Variations on Consumers' Attitudes toward Brands," *Journal of Consumer Research*, 39 (3), 644–61.

Shickel, Benjamin, Martin Heesacker, Sherry Benton, Ashkan Ebadi, Paul Nickerson, and Parisa. Rashidi (2016), "Self-reflective Sentiment Analysis," in Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology, San Diego, CA, June 2016, 23–32.

Søgaard, Anders, Anders Johannsen, Barbara Plank, Dirk Hovy, and Héctor Martínez Alonso. (2014), "What's in a p-value in NLP?" in Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Baltimore, MD, 1–10.

Stillman, Jessica. (2018), "Here's the Data That Proves Nike's Colin Kaepernick Ad Is Seriously Smart Marketing," https://www.inc.com/jessica-stillman/heres-data-that-proves-nikes-colin-kaepernick-ad-is-seriously-smart-marketing.html [last accessed November 2019].

Tirunillai, Seshadri, and Gerard J. Tellis (2012), "Does Chatter Really Matter? Dynamics of User-Generated Content and Stock Performance," *Marketing Science*, 31 (2), 198–215.

Torsten (2011), Australia vs. Germany: "Wine Rambler Blind Tasting Madness part 8," https://www.winerambler.net/blog/australia-vs-germany-wine-rambler-blind-tasting-madness-part-8 [last accessed November 2019].

Toubia, Olivier, Garud Iyengar, Renée Bunnell, and Alain Lemaire (2019), "Extracting Features of Entertainment Products: A Guided LDA Approach Informed by the Psychology of Media Consumption," *Journal of Marketing Research*, 56 (1), 18–36.

U.S.A. Wine Ratings (2019), "Top Wine Varietals of the USA in Terms of Sales," https://usawineratings.com/en/blog/insights-1/top-wine-varietals-of-the-usa-in-terms-of-sales-208.html [last accessed January 2020].

Wang, Amy and Rachael Siegel (2018), "Trump: Nike 'getting absolutely killed' with Boycotts over Colin Kaepernick's 'Just Do It' Campaign," https://www.washingtonpost.com/business/2018/09/04/people-are-destroying-their-nike-gear-protest-colin-kaepernicks-just-do-it-campaign/?noredirect [last accessed November 2019].

Wen, Zeyi, Jiashuai Shi, Qinbin Li, Bingsheng He, and Jian Chen (2018), "ThunderSVM: A Fast SVM Library on GPUs and CPUs," *Journal of Machine Learning Research*, 19 (21), 1–−5.

Zhou, Yiwei and Alexandra I. Cristea. (2016), "Towards Detection of Influential Sentences Affecting Reputation in Wikipedia," in Proceedings of the 8th ACM Conference on Web Science, Hannover, Germany, 244–48.