Strength in Numbers: Using Big Data to Simplify Sentiment Classification

Apostolos Filippas
IOMS Department, NYU Stern School of Business, afilippa@stern.nyu.edu

Theodoros Lappas
School of Business, Stevens Institute of Technology, tlappas@stevens.edu

Sentiment classification, the task of assigning a positive or negative label to a text segment, is a key component of mainstream applications such as reputation monitoring, sentiment summarization, and item recommendation. Even though the performance of sentiment classification methods has steadily improved over time, their ever-increasing complexity renders them comprehensible by only a shrinking minority of expert practitioners. For all others, such highly complex methods are black-box predictors that are hard to tune and even harder to justify to decision-makers. Motivated by these shortcomings, we introduce `BigCounter`: a new algorithm for sentiment classification that substitutes algorithmic complexity with Big Data. Our algorithm combines standard data structures with statistical testing to deliver accurate and interpretable predictions. It is also parameter-free and suitable for use virtually "out of the box," which makes it appealing for organizations wanting to leverage their troves of unstructured data without incurring the significant expense of creating in-house teams of data scientists. Finally, `BigCounter`'s efficient and parallelizable design makes it applicable to very large datasets. We apply our method on such datasets toward a study on the limits of Big Data for sentiment classification. Our study finds that, after a certain point, predictive performance tends to converge and additional data have little benefit. Our algorithmic design and findings provide the foundations for future research on the data-over-computation paradigm for classification problems.

1. Introduction

The ability to automatically label the sentiment of a given text segment as positive, negative, or neutral, is a fundamental component of mainstream applications such as reputation monitoring,[1] sentiment summarization,[2] review mining,[3] recommender systems design,[4] and modeling consumer behavior.[5] Relevant literature typically refers to this task as *sentiment classification* or *sentiment analysis*.[6,7]

The popularity of sentiment classification has motivated a significant body of work and has led to the design of numerous algorithms.[8,9] A study of the relevant literature in chronological order reveals that these algorithms are becoming significantly more complex with time. Early work primarily focused on simple lexicon-based approaches, which were then extended by incorporating basic linguistic features.[10,11] The next stage brought about the use of increasingly complex machine learning algorithms that formulate sentiment classification as a supervised learning task.[6,12] The use of Natural Language Processing (NLP) techniques for feature engineering further improved the results of this approach, while also increasing its complexity.[13,14] Today, the state-of-the-art tries to unlock the power of deep learning: a branch of machine learning based on modeling abstract relationships in unstructured text via multi-layer graphical models, such as artificial neural networks[15] and advanced language constructs, such as word embeddings.[16]

Even though new techniques are consistently pushing the performance boundaries in sentiment classification, their ever-increasing complexity has multiple drawbacks that we address in our work:

- **Interpretability:** Complex machine learning methods typically have very limited interpretability: even though we can prove that they perform well by testing them on new data, their non-linear nature makes their outcomes hard to explain to decision makers.[17]
- **Parameter Tuning**: Complex algorithms are notoriously hard to tune, as the vast number of parameters prohibits manual tuning and requires exhaustive or "intelligent"—but still computationally expensive—methods for automatic tuning.[18,19]
- **Specificity**: Existing methods are designed for classifying entire documents. As a result, their accuracy suffers when they have to classify smaller segments, such sentences or phrases, that bear very little evidence in terms of vocabulary or context.
- **Infrastructure Cost:** Algorithmic complexity translates to steep rises in computational costs, as well as in software and hardware costs.[20–22]
- **Recruitment Cost:** Firms willing to accept the high infrastructure costs must inevitably also invest in top quality talent that is able to build, manage, and utilize this infrastructure. In practice,

Table 1 Sentences of easy, medium, and high difficulty for sentiment classification

Easy	Medium	Hard
The food was great.	We will stay longer next time.	Minutes to downtown.
Best experience of my life.	The location cannot be beat.	They have shuttles if needed.
The most boring game ever.	Hats off to the chef.	I was told the food would take 5 min.
We loved the food.	The place has an eclectic feel.	This was our fourth stay here.
The soup was yummy.	Million buck view.	They leave little chocolates on your pillow.
Easily accessible and very clean.	They go above and beyond.	Parking was extra.
Very reasonable prices.	It's full of cockroaches.	They gave us extra time to check out.
This place is a hidden treasure.	The food is not worth the wait.	I called reception twice, no response.
The rooms were clean and sunny.	Parking is not very conveinant.	We asked for a different room.
The view was definitely a bonus.	Did not live up to standards.	Jet planes also landing nearby.

however, limited funds and steep competition in the talent market have turned the acquisition of technical talent into a challenging endeavor, especially for small and medium-sized firms.[23–26]

The negative consequences of the increasingly complex algorithms for sentiment classification motivate us to consider a different approach. Table 1 presents a set of sentences that are easy, medium, and hard to classify as positive or negative. Even a simple, lexicon-based approach that counts the number of known positive or negative words in each sentence (e.g. *amazing, horrible*), and reports the label with the majority count, would perform satisfactorily for the examples in the easy group. However, this approach would not work for the examples in the medium group, as they do not include such obvious leads. Instead, these sentences express their true sentiment via (i) words that are unlikely to be found in standard lexicons (e.g. *cockroach*, *eclectic*, *meh*, the misspelled word *conveinant*), (ii) expressions that consist of words that carry no sentiment on their own (e.g. idioms such as *above and beyond*, *hats off*), and (iii) expressions that change the sentiment of known opinion words (e.g. *not worth*). A supervised approach could address occurrences of type (i) provided that the training corpus includes enough occurrences of these atypical words to affect the prediction. This is more likely to happen if the the corpus includes documents from the same domain (e.g. the word *cockroach* is more likely to appear in restaurant reviews than in book

reviews). On the other hand, given that standard supervised approaches ignore the order of the words (i.e. the *bag-of-words* approach), they would fail to address types (i) and (ii). To address such cases, the practitioner would have to engineer advanced features (e.g. entities, n-grams, noun phrases) that can potentially capture at least some of these occurrences.

Finally, the hard group includes a set of a very challenging examples. In addition to lacking both typical and rare opinion words, these examples do not include obviously positive or negative expressions. Instead, the polarity of these sentences is highly *context-dependent*. For instance, the sentence *This was our fourth stay here*, which we extracted from a hotel review, informs us that the reviewer has repeatedly visited the hotel in the past. Given that a customer is highly unlikely to return four times to a business that she is not pleased with, we intuitively expect this sentence to carry positive sentiment. Another example is the phrase *Minutes to downtown*, which reveals the hotel's close proximity to a specific location. The fact that a city's downtown area is typically a popular destination (especially for hotel guests), implies a positive sentiment. At a glance, the sentence *Jet planes also landing nearby* does not carry any sentiment. However, its negative polarity becomes apparent if we consider that it comes from a hotel review. Specifically, the noise that inevitably comes with the hotel's proximity to an airport is clearly undesirable for the hotel's guests. One could argue that proximity to the airport enhances the hotel's accessibility and could motivate positive comments. However, such a comment would be unlikely to include the terms *jet planes* and *landing*. Instead, we would expect a positive proximity comment to be similar to the one that we discussed previously (e.g. *Minutes to the airport*).

The dependency of the polarity of these examples on context-specific information makes them particularly challenging to address. Standard lexicon-based or supervised methods are unlikely to be effective, even if they are extended via the use of advanced feature engineering. Further, the state-of-the-art in deep learning has only recently started to explore textual representations that take context into account.[27,28] Even though the initial findings of such efforts are very promising in terms of performance,[29,30] they also introduce yet another layer of complexity, and demand

familiarity with advanced concepts such as artificial neural networks and word embeddings.[16,31] These obstacles limit the (informed) use of such methods to the minority of practitioners and firms that possess the necessary experience, infrastructure, and skills. In addition, the complexity of such methods essentially eliminates the interpretability of their results, especially for non-experts.

The examples in Table 1 verify that more challenging sentences require more complex algorithms. As we get closer to simulating the way in which humans process and generate natural language constructs, it is reasonable to expect that our algorithmic machinery will get larger and more complex. What if we could take a radically different, purely agnostic approach that is seemingly oblivious to the context or thought process behind a statement? Let us again consider the phrase *Minutes to downtown*. Suppose that we have a random sample of 5 English hotel reviews that include the phrase *Minutes to downtown*. Out of these 5 reviews, 4 are positive and 1 is negative. This small sample provides some evidence that the phrase is positive. However, some of the occurrences of this phrase may be random and have little to do with the review's overall rating. Our confidence is thus limited due to the small size of the sample. If, for instance, our sample consisted of 9700 positive and 300 negative reviews, then our confidence would be much higher. Similarly, if our sample consisted of 5050 positive and 4950 negative reviews, a safer prediction would be that the statement is neutral. What if we had a sample of $100,000$ or even a sample of 1 million reviews? Previous work has explored the value of Big Data for predictive tasks.[32–34] Intuitively, we expect convergence to a confident prediction to occur after a certain sample size.

In this work, we apply this simple count-based paradigm to design the `BigCounter` algorithm for sentiment classification. Given a very large corpus that includes both positive and negative documents of arbitrary length (e.g. customer reviews), `BigCounter` predicts the sentiment of a given short text sequence s, such as a sentence or phrase, by using a simple statistical test to compare the positive and negative counts of s in the corpus. In order to account for the fact that some sequences might not occur frequently enough to allow for a confident test, we extend `BigCounter` to count the frequency of flexible wildcard patterns extracted from s. The algorithm delivers an accuracy that competes and often surpasses the state-of-the-art.

The simple design of the `BigCounter` algorithm provide it with the following advantages:

1. It adopts a simple algorithmic approach based on standard data structures and statistical testing, leading to minimal software requirements and a very lightweight implementation.

2. It is parameter-free and suitable for use virtually "out of the box," which makes it appealing for organizations wanting to leverage their troves of unstructured data without incurring the significant expense of creating in-house teams of data scientists.

3. It produces easily interpretable predictions that can be traced back to actual examples from the input dataset.

4. It can accurately classify sentences, even if it is trained on data labeled just at the document level. Our comparisons with two benchmark algorithms demonstrate its advantage on real datasets.

5. It is naturally parallelizable and thus scalable to very large datasets.

Our methodology has implications for practitioners in both academia and industry, as it offers a simple and effective alternative to the increasingly complex methods for sentiment classification. In addition, our methodology lowers the barrier to entry for organizations that want to mine their growing text repositories but cannot afford the infrastructure and talent required by state-of-the-art machine learning algorithms. Further, our study on the limits of Big Data can help managers make informed decisions about how much data their firm needs in order to achieve accurate classification results. Finally, our work lays the foundations for future research on the use of similar methods for multi-label classification tasks in other domains.

2. Background and related Work

We begin this section with an overview of extant methods for sentiment classification. We then discuss the motivation and theoretical background of our own approach.

2.1. Sentiment classification

Sentiment analysis (also known as opinion mining) is a broad field that covers multiple tasks relevant to extracting opinions from different types of unstructured text, such as customer reviews, articles, and blog posts. Arguably the most prevalent of these tasks is that of classifying a given text segment as positive or negative,[8,9] which is also the focus of our own work.

The underlying theme of all previous relevant methods is their effort to simulate the way human authors generate text to encode their sentiment. Lexicon-based approaches were the earliest successful attempts in this direction. These methods predict the polarity of a text by counting the number of known positive and negative words that it includes. At its core, this approach utilizes two lexicons of positive and negative words, which the practitioner needs to provide as input.[3,35,36] Several extensions were subsequently proposed, such as domain-specific[37] and automatically constructed[38] lexicons. Natural Language Processing (NLP) techniques can be used to capture linguistic constructs that cannot be addressed by simple lexicons or extract semantic information that can improve an algorithm's prediction. For instance, previous work has combined lexicons with linguistic constructs such as negation rules (e.g. "not good"), rules that enhance or change a word's sentiment (e.g. "very good"), intra- and inter-sentence conjunctions, synonyms, and antonyms.[10,11,39] Lexicon-based approaches are intuitive, easy to implement, and can also be competitive if properly customized for the domain of application. However, they require considerable tuning and fail to predict statements that do not include leads from the underlying lexicons, such as those that we showed in the second and third columns of Table 1.

A second family of methods formulates sentiment classification as a standard supervised learning problem. In this setting, the input to the method consists of a set of training instances with known labels. Given the training data, a machine learning algorithm builds a predictive model that is then used to classify new instances. One of the main benefits of this approach is that it allows us to experiment with a wide range of established classification algorithms, such as Naive Bayes,[40] Logistic Regression,[41] and Support Vector Machines.[6,14] In the absence of sufficient manually annotated data, the interested practitioner can use a semi-supervised algorithm to extend the training set with automatically labeled instances.[42]

Even though supervised learning techniques can be very competitive in the context of polarity prediction, their simplifying assumptions prevent them from accurately predicting challenging instances that use context and complex linguistic constructs to express sentiment. Arguably the

most influential assumption is that the order of the words in a document does not affect its overall sentiment, which allows algorithms to treat the document as a bag of words. This assumption greatly simplifies the process of building predictive models, but also sacrifices the valuable information that comes with the order of the words, such as idioms with a clear positive or negative sentiment. This information can be partially salvaged via the use of NLP techniques to improve feature engineering. Specifically, rather than using single words as tokens, one could define complex features such as n-grams (e.g. *New York*, *buffalo chicken wings*), or combinations of neighboring words (e.g. a binary feature that encodes whether or not the words *airport* and *noise* appear in the same sentence). Even though such features can indeed lead to performance improvements, feature engineering is a complex task that involves manual tuning and the consideration of an arbitrarily large number of candidate features.

The state-of-the-art from the domain of machine learning includes methods from the exciting area of deep learning. These methods use advanced graphical models, such as different variants of neural networks, to model various levels of abstractions over the input data. Artificial neural networks are inspired by the neuron structure in the human brain, and their ultimate goal is to model the brain's decision making and processing functions.[43] One of the benefits of deep learning methods is the utilization of advanced word representations that go far beyond single words or even simple linguistic features. Arguably the most characteristic example is the use of word embeddings: continuous word representations based on the assumption that words in similar contexts have similar meanings.[31,44] Deep learning research has evaluated neural networks in the context of various domains, including sentiment classification. For instance, convolutional neural networks (CNNs) have been used for polarity prediction[30] Such models introduce a single layer of convolution over a set of word representations obtained via previously proposed unsupervised neural language models.[15,31] For the same task, good performance is obtained by a variant of the standard recursive neural network (RNN), referred to as a *recursive neural tensor network*,[29] that allows for direct interactions between the continuous representations (embeddings) of the words in the considered

vocabulary. This facilitates the detection of meaningful linguistic constructs, such as negation. Neural network have also been customized for sentiment classification by incorporating sentiment scores into word embeddings.[45] Additional graphical models for polarity prediction include gated recurrent neural networks[46] and adaptive recursive neural networks.[47]

Previous work has repeatedly verified the superiority of deep learning methods for sentiment classification. However, their performance comes at the cost of significant complexity and decreased interpretability.[48] Neural networks are typically treated as "black boxes," as the large number of interacting non-linear parts make it difficult to understand exactly how they function and to interpret their results.[17] This can create resistance to adoption of these techniques in business settings, especially in highly regulated industries with decision makers that lack the expertise required to fully comprehend the inner workings of such complex models.[49] This type of criticism precedes the emergence of deep learning, as it dates back to the early days of artificial neural networks.[50–52]

2.2. Our Approach: Memory Over Computation

The shortcomings and ever-increasing complexity of existing methods motivate us present a novel and much simpler algorithm for sentiment classification. Consider a random English sentence S that we need to classify as positive, negative or neutral. Our training corpus $\mathcal{D}$ consists of documents (e.g. customer reviews) that have been manually annotated as positive or negative. Rather than try to reverse engineer S, we opt to treat S as a single object and simply count the number of times that it appears in a positive and in a negative document from $\mathcal{C}$. If the difference in the two counts is statistically significant, we mark S with the majority label. Otherwise, we mark it as "neutral". Admittedly, if the frequency $N_{S,\mathcal{C}}$ of S in C is low, then our confidence in the prediction will also be low. Our confidence rises as $N_{S,\mathcal{C}}$ becomes larger, and we expect convergence to occur after a certain point. For instance, $N_{S,\mathcal{C}} = 1,000,000$ likely does not lead to more accurate predictions than $N_{S,\mathcal{C}} = 100,000$ or even $N_{S,\mathcal{C}} = 10,000$.

Our "memory over computation" approach is motivated by the usage-based paradigm for language learning, which posits that children develop their language skills by initially memorizing and

gradually refining simple patterns.[53–55] For example, after memorizing the pattern "Where is the X?," the child can then customize it by substituting X for other terms or constructs that represent meaningful objects such as "book" or "cookie jar." In this setting, children learn how to properly select which constructs to insert into each pattern based on frequency of usage. For instance, while the child is likely to hear the phrase "Where is the cookie?" often, she is unlikely to hear the phrase "Where is the eat?." The first phrase will thus be validated via observation and repetition, while the second one will be rejected. This mechanism allows the pruning of the endless patterns and combination that can theoretically exist in a language.

The usage-based paradigm has strong ties to extensive theoretical work on formulaic sequences and their effects on language learning.[56,57] A formulaic sequence is a continuous or non-continuous sequence of words that is stored and retrieved from memory at the time of use, and is not subject to generation or analysis by the language grammar.[58] Formulaic sequences offer processing efficiency because single memorized units, even if made up of a sequence of words, are processed more quickly and easily than the same sequences of words which are generated creatively.[59,60] In other words, it is easier to memorize prefabricated chunks of language that can be used on-demand, rather than to build a new sequence by considering vocabulary and grammar rules.[61–63]

The success of our approach comes down to a simple question that we address in this work: *do we have enough data to confidently predict the polarity of any possible formulaic sequence?* At a glance, the response to this question is negative, as the number of possible sentences is simply overwhelming and its would require an unrealistically large training corpus. However, as we demonstrate in our study, we can drastically reduce this data dependency and achieve highly accurate predictions via the use of flexible patterns that allow the replacement of words with wildcards. For instance, if the frequency of the pattern "*the * cannot be beat*" in the input corpus is large enough to allow for a confident prediction of its polarity, then we can memorize this pattern and use it to classify any matching sentences (e.g. "*the location cannot be beat*," "*the food cannot be beat*").

This flexible representation is similar in spirit to that of the "schema theory" that was introduced by Holland,[64] and served as the basis of numerous follow-up works on genetic algorithms.[65–67]

According to schema theory, one can cover a large part of a multidimensional space via a schema that defines constraints on the defining dimensions. For instance, a general schema would restrict one of the dimensions to a specific value while allowing the others to assume any value. While this schema would cover a very large part of the search space, it is likely too general to represent a meaningful pattern that applies to many valid points. Hence, by balancing the restrictiveness-applicability tradeoff, we can discover interesting rules that accurately and succinctly represent our data.[66] As we discuss in detail in Section 3, our algorithm faces a similar tradeoff, as it utilizes textual patterns that include both fixed terms and wildcards and can therefore match multiple sentences. In the context of sentiment classification, the goal is then to mine patterns that can match many sentences of the same (negative or positive) polarity and few or no sentences of the opposite polarity.

3. The `BigCounter` Algorithm

In this section we present the `BigCounter` algorithm for sentiment classification. Our method belongs to the broad class of supervised machine learning algorithms and operates in three steps: preprocessing, training, and prediction.

3.1. Preprocessing

The input to the preprocessing phase is a large collection of *weakly annotated* documents, i.e. documents that are annotated as positive or negative at the document level but not necessarily at the sentence level. `BigCounter` begins by segmenting each document D into its sentences. For each sentence S, the algorithm then computes $\mathcal{P}_S$: the set of all possible patterns generated if we replace every possible subset of words in S with wildcards ($*$) that represent *any* word. As we discuss in the following section, the use of wildcards allows `BigCounter` to predict the sentiment of sentences that have a very low or even zero frequency in the training corpus.

If D has a positive (negative) label, then we increment the positive (negative) count of every pattern $P \in \mathcal{P}_S$ by 1. We demonstrate this with an example in Figure 1. In this example, we focus on the sentences "*I would definitely return*" and "*I would never return*," mined during the

preprocessing phase from a positive and negative review, respectively. The figure presents all the possible flexible patterns that can be generated from these two sentences. The middle column includes the patterns that the two sentences have in common. For each of these common patterns, we increment their respective positive and negative counts by 1. For the patterns in the left and right columns we increment only the positive and only the negative count, respectively.

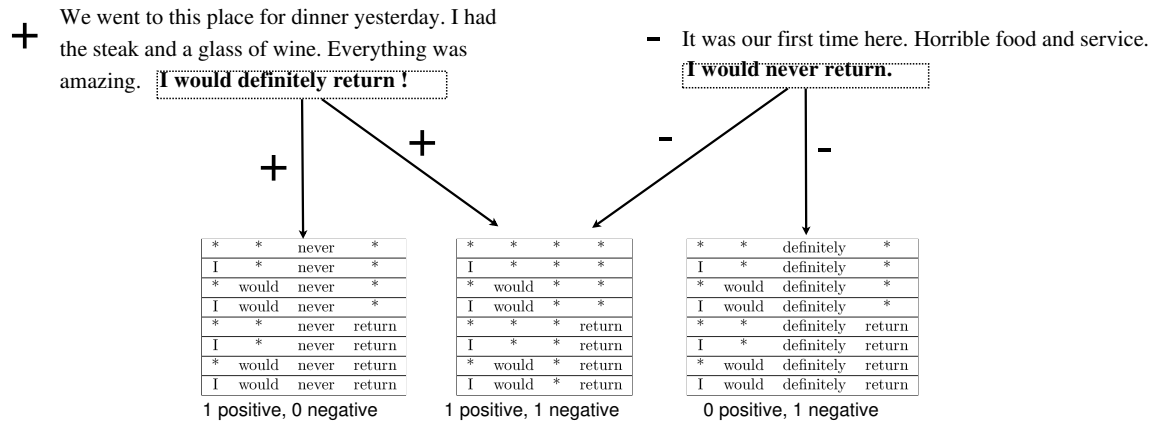


Figure 1 Patterns generated from the sentences: "I would definitely return" and "I would never return".

3.2. Training

The input to the training phase consists of a collection of distinct wildcard patterns, as well as a positive and negative count for each pattern. The training phase uses the two counts to assign a positive or negative label to the pattern.

We formally model the assignment as a two-sided coin toss, where the possible outcomes are positive or negative. Given a pattern P, let p_P denote the true probability of a positive outcome and let N_P^+, N_P^- be the pattern's positive and negative counts, respectively. We want to decide whether the two counts provide enough evidence to reject the *neutrality hypothesis* $H_0 : p_P = \frac{1}{2}$, which states that P is equally likely to occur in positive and negative texts, and hence bears no sentiment. The appropriate statistical test for this task is the binomial test, which examines whether the deviations of the distribution of observations with two possible classes from the theoretically expected distribution are statistically significant. We also account for the fact that positive reviews

are known to be much more common than negative reviews[68] by updating p_P to reflect the class proportions in the dataset. *

If our test rejects the neutrality hypothesis, then we assign the majority label to P and record it in a simple key-value (pattern-label) store, which we refer to as the `Polarity-Index`. The algorithm repeats this process for all the patterns extracted from all the documents in the training corpus. We present the pseudocode for `BigCounter`'s preprocessing and training phases in Algorithm 1.

3.3. Prediction

Prediction with `BigCounter` is a straightforward task. Let S be a sentence whose sentiment we want to determine. The prediction process begins by extracting the set of flexible patterns $\mathcal{P}_S$. The algorithm then retrieves the polarities of the patterns from $\mathcal{P}_S$ that are contained in the index $\mathcal{I}$. If the index includes multiple matching patterns, `BigCounter` simply labels S with the majority label. If the numbers of positive and negative matching patterns are equal, or if no matching patterns exist, then there is insufficient evidence that S bears sentiment, and it is labeled as neutral. Essentially, `BigCounter`'s prediction phase is based on the same principle that ensemble methods utilize: the predictions of multiple weak classifiers are aggregated to produce a final prediction.[70,71]

One can consider different aggregation policies by assigning weights to the predictors that match a sentence. For example, matching flexible sequences with fewer wildcard characters may be assigned higher weights. In our experiments on the algorithm's predictive accuracy we use unweighted majority voting, as we found alternative policies to be less competitive.

3.4. Discussion

One of the main benefit of `BigCounter` is that it can deliver accurate sentence-level predictions even when trained on instances with document-level annotations. The large volume of product reviews on online marketplaces is an inexpensive source of such data. In contrast, other state-of-the-art approaches, such as recursive neural networks, often require sentence-level or even phrase-level annotations.[29]

* For large values of $N_P^+ + N_P^-$ we can use the faster χ^2 test to closely approximate the binomial test.[69]

Algorithm 1 `BigCounter` Training.

Input: Corpus of training docs $\mathcal{D}$, confidence level α

$\mathcal{U} = \emptyset$ ▷ Set of unique wildcard patterns

$pos = \{\}$, $neg = \{\}$ ▷ Key-value stores for the pos and neg counts of each pattern

for each doc $D \in \mathcal{D}$ **do**

  for each sentence S of D **do**

    Generate the set of wildcard patterns $\mathcal{P}_S$ from S.

    for each pattern $P \in \mathcal{P}_S$ **do**

      Add P to $\mathcal{U}$

      if D is positive **then** $pos[P] += 1$ ▷ $pos[P]$, $neg[P]$ are initialized to 0

      else $neg[P] += 1$

`Polarity-Index`={} ▷ An empty key-value store

Set p equal to the percentage of positive docs in $\mathcal{D}$

for each pattern in $\mathcal{U}$ **do**

  if $BinomialTest(pos[P], neg[P], p) < \alpha$ **then**

    if $pos[P] > neg[P]$ **then** `Polarity-Index`$[P] = +$

    else `Polarity-Index`$[P] = -$

Return `Polarity-Index`

On the other hand, `BigCounter` naively assumes that all the patterns mined from an overall positive (negative) document also carry positive (negative) sentiment. While positive reviews will generally include positive patterns, there are some cases where this assumption is violated. For instance, only part of the text in a customer review actually carries sentiment while the rest covers objective information, positive sentences are used sarcastically in negative reviews, or an otherwise positive review includes a single negative comment. Such inevitable occurrences are likely to contaminate the positive and negative counts of each pattern. Despite such contaminations,

given a very large dataset of reviews with many occurrences of a truly positive pattern, the pattern will appear in more positive than negative reviews (the opposite is the case for negative patterns). Finally, the difference between the positive and negative counts of neutral patterns will not be statistically significant (as determined by the binomial test) and thus, the pattern will not be added to the `Polarity-Index`.

4. Scalability Enhancements

In this section we describe three techniques that enable us to efficiently apply our method on very large datasets.

4.1. Efficiently performing many statistical tests

The `BigCounter` algorithm performs a large number of statistical tests during training. By exploiting a simple structural property of the binomial test we can significantly speed up this phase. Consider a pattern P with $N_P^+ + N_P^- = n$. The binomial test will label P as positive if N_P^+ is sufficiently large. Intuitively, there should exist some number $\overline{x}$ such that we can reject the neutrality hypothesis and label the sequence as positive if and only if $N_P^+ > \overline{x}$.

We prove that such a number exists and that we can efficiently compute it[†]. Let α denote the confidence level which we are using in our statistical tests and let $p = \frac{N_P^+}{n}$. The bound $\overline{x} = \overline{x}(N_P^+, n, \alpha)$ the solution to the following problem:

$$\begin{aligned} \min_{x} \quad & x \\ \text{s.t.} \quad & \sum_{k=x}^{n} \tbinom{n}{k} p^k (1-p)^{n-k} \leq \alpha \\ & x \in \{0, 1, \ldots, n\}. \end{aligned} \tag{$P1$}$$

The optimization problem $(P1)$ is non-linear and discrete. However, observe that as the value of x decreases, the left-hand side of the inequality constraint strictly increases since more positive terms are included in the summation. Therefore, the optimal solution can be obtained following a greedy procedure: search through the solution space starting at $x = n$ and decrease the value of

[†] The bound for labeling the pattern as negative can be computed in the exact same way.

x until a non-feasible solution x^* is reached, at which point set $\overline{x} = x^* + 1$. In the case that $P1$ is infeasible, the proposed procedure outputs $\overline{x} = n + 1$, i.e a bound that never rejects the neutrality hypothesis.

For a given class probability p and confidence level α, we are now able to pre-compute these bounds for different values of n, and store them in a simple key-value store, such a dictionary or hash map. This substitutes statistical tests (which in the case of the binomial test involve computationally cumbersome factorials) with fast memory lookups, thus dramatically speeding up the training phase. Further, it is straightforward to apply our analysis to obtain similar bounds for the χ^2 test, which can be used instead if the number of observations n is sufficiently large.

4.2. Stopword Homogenization

The notion of *stopwords* is used in linguistics to refer to the most commonly used words in a language. For the English language, the list of stopwords include words such as "*a*," "*the*," "*this*," "*be*," and "*and*," which account for close to 40% of the words of all written text.[7][2]

We utilize stopwords to reduce the memory requirements of the `BigCounter` algorithm. During preprocessing, every stopword is replaced by the special token *š*. We refer to this step as *stopword homogenization*. The homogenization step has significant computational impact, as it greatly reduces the number of generated patterns. For example, the sentences "*The price cannot be beat*" and "*This price cannot be beat*" would both be homogenized to "*š price cannot š beat*" and would thus generate the exact same set of patterns. Stopword homogenization has a significant computational impact, as it reduces the memory footprint of `BigCounter` by about 75%.

4.3. Parallelizability

Building the `Polarity-Index`can be a computationally strenuous task, especially if we want to scale up to real-life Big Data applications. Next, we demonstrate that the index-building process is highly parallelizable and can be completed via parallel threads on any distributed infrastructure.

Consider a pattern S of length n. The first observation is that only sequences of the same length as S may contribute in computing the positive and negative counters N_P^+, N_P^- for the pattern.

Therefore, we can build the index for each distinct length independently and in parallel. The second observation is that two sentences of the same length can generate the same pattern only if they include stopwords in the exact same positions. For example, the sentence $S =$"*Great price for the quality*" has stopwords in the third (*for*) and fourth (*the*) positions. Therefore, the set of patterns $\mathcal{P}_S$ that it generates will have the stopword token $\check{s}$ in these positions. It follows that a sentence with a different allocation of stopwords cannot generate any of the patterns in $\mathcal{P}_S$. We conclude that sequences with different stopword allocations are independent with respect to the statistical tests that `BigCounter` conducts during its training phase. Based on these two observations, our implementation of `BigCounter` assigns a separate thread for each batch of sentences that share the same length and the same stopword structure, efficiently parallelizing the training phase.

5. Evaluation

In this section we present the experiments that we conducted toward the evaluation of the `BigCounter` algorithm. We begin with a brief overview of our datasets. We then evaluate the predictive accuracy of `BigCounter` by conducting extensive tests that include comparisons with the state-of-the-art. We conclude our evaluation with a study on the limits of Big Data for sentiment classification. All the datasets and software implementations used in this section are openly accessible or can be made immediately available upon request.

5.1. Datasets and Setup

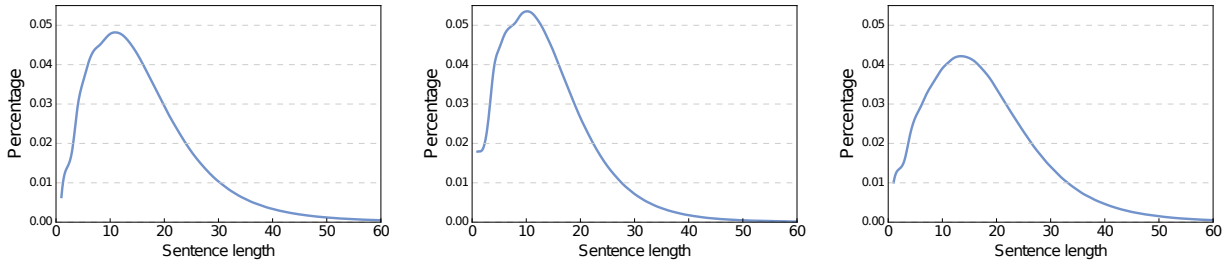
Raw Data: We utilize three datasets of reviews from the TripAdvisor, Yelp, and Amazon websites, which we crawled over the period between September 2015 and December 2015. TripAdvisor reviews pertain to the hotel industry, Yelp reviews to the restaurant industry, and Amazon reviews are

Table 2 Basic Dataset Statistics - Ratings Distribution

Dataset	Pos	Neg	Neutral	Total
Hotels	4,048,966	619,954	727,662	5,396,582
Restaurants	2,873,114	236,121	537,497	3,646,732
Books	1,166,161	185,000	134,705	1,485,866

focused on books. The challenge in the book domain stems from its highly subjective nature, as well as the richness of the language used in book reviews. The main benefit in utilizing three different datasets is that we can assess the generalizability of our results, across both websites and review domains. Our datasets include the text and a rating from 1 to 5 stars for each review. We assume that reviews with 4 or more stars are positive and reviews with 2 or less stars are negative.

Table 2 provides an overview of our datasets. Positive reviews are by far the most frequent category in all three datasets, confirming the existence of a strong positive bias.[68] We also report the distribution of sentence length (measured in number of words) for all three datasets in Figure 2. We observe a similar distribution across datasets, with a clear peak at approximately 10 words. As the sentence length increases, our data becomes sparser. Given that `BigCounter` relies on the availability of a large training corpus to deliver accurate predictions, we will refer to these plots to help us interpret any variation in our results.



(a) Hotels (b) Restaurants (c) Books

Figure 2 Distribution of sentence length (measured in number of words) for the three datasets.

Ground Truth Data: We construct 6 ground-truth datasets with sentence-level sentiment annotations as follows: First, for each of the three review domains, we sample 100 sentences uniformly at random for reviews belonging to each star rating category (1 to 5 stars). This ensures that the test sets are sufficiently large (500 sentences) and balanced with respect to ratings. We then ask 10 human annotators hired via Amazon's Mechanical Turk platform to manually annotate each sampled sentence as *positive*, *negative*, or *neutral*. If the majority label does not have the support

of at least 7 of the 10 annotators, then the sentence is discarded. ‡ This first sampling phase leads to a corpus 3 datasets (1 per domain) with 500 annotated sentences per dataset. We refer to the first corpus as `SentenceCorpus`.

We then repeat the process for all 3 domains domain, but this time restrict the sampling process to sentences that do not include known positive or negative words (e.g. *great, amazing, horrible*). The identity of such words is verified via their presence in the established lexicons introduced in.[35] § The elimination of sentences with obvious sentiment cues leads to a much more challenging test set, wherein the authors use more complex linguistic patterns to express sentiment. This second sampling phase leads to 3 additional datasets (1 per domain) with 500 annotated sentences per dataset. We refer to this second corpus as `SentenceCorpus-Hard`. In total, our evaluation includes $(3 \times 500) + (3 \times 500) = 3000$ annotated sentences.

5.2. Sentiment Classification

In this experiment, we assess the ability of `BigCounter` to predict the sentiment of sentences, and compare it with other state-of-the-art techniques. We compare our algorithm against two competitive baselines: support vector machines (SVMs) and recursive neural tensor networks (RNNs). SVM-type algorithms are based on an intuitive idea: given the representation of the training set examples in the feature space, the goal is to find the hyperplane that maximizes a distance measure between the different classes. SVMs have met success across a wide range of real life applications, including sentiment classification.[73–75] We use the multi-class SVM implementation of Scikit-learn, which we tune by performing an extensive grid search over the hyperparameter space.¶

The second baseline belongs to the class of deep learning algorithms. More specifically, we employ the Stanford CoreNLP sentiment analysis implementation.[29] The algorithm is a sentence-level model, which begins by utilizing a pipeline of NLP techniques that include lexical and syntactical

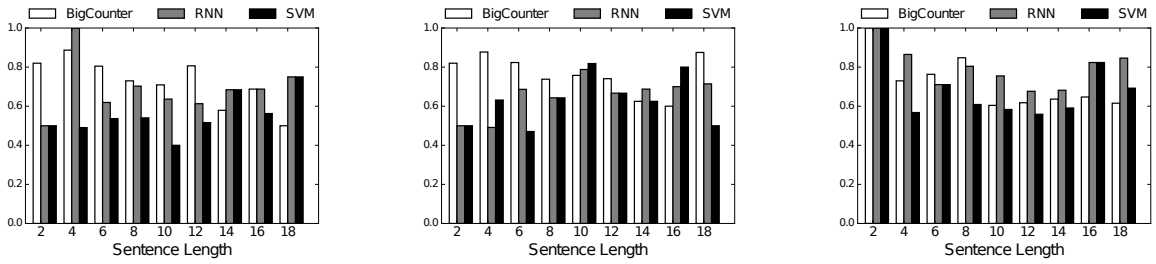
‡ Due to the apparent triviality of the annotation task for human annotators only a handful of sentences were actually discarded

§ `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`

¶ See also `goo.gl/bZgCeg`, last accessed on September 1st, 2016.

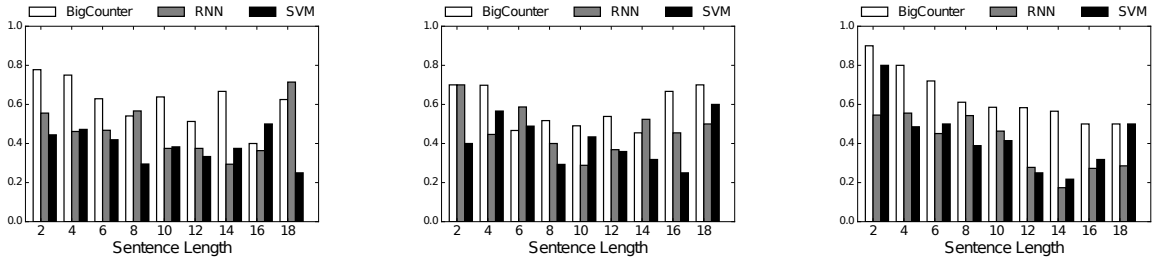
sentence parsing, tagging, and construct identification, to construct a tree representation of the sentence structure. In this implementation, the parse trees are constructed from a publicly available movie review dataset,[76] all unique phrases are extracted from the trees and require manual annotation. A recursive neural tensor network (RNN) is then trained on top of the extracted linguistic structure and annotations.

All three algorithms are evaluated on the `SentenceCorpus` and `SentenceCorpus-Hard` corpora, which include sentences with three different labels (positive, negative, neutral). We present the results of our experiments on the `SentenceCorpus` and `SentenceCorpus-Hard` corpora in Figures 3 and 4, respectively. The y-axis represents the achieved accuracy, while each bar on the x-axis represents a different sentence length (i.e. number of words). We report the results for each length separately in order to study the consistency of the three algorithms.



(a) Hotels (b) Restaurants (c) Books

Figure 3 Predictive accuracy as a function of sentence length, evaluated on the `SentenceCorpus` corpus.



(a) Hotels (b) Restaurants (c) Books

Figure 4 Predictive accuracy as a function of sentence length, evaluated on the `SentenceCorpus-Hard` corpus.

For the SentenceCorpus corpus, BigCounter consistently outperforms both baselines for most sentence lengths in the hotels and restaurants domains. For all three domains, BigCounter is very competitive and often the winner for sentences that include up to 8-10 words. The RNN demonstrates a slight advantage for longer sentences, although this advantage is not consistent and the top approach tends to vary across datasets and sentence lengths. We anticipated a decrease in BigCounter's accuracy for longer sentences, as the algorithm depends on the size of the training set, which becomes increasingly smaller for longer sentences. This is also supported by Figure 2, which verifies that the availability of sentences steadily decreases after their length surpasses 8-10 words. We examine and quantify the effect of data availability in the following section, where we discuss our study on the limits of Big Data for sentiment classification.

Even though, as anticipated, the accuracy of all three algorithms is lower for the SentenceCorpus-Hard, our algorithm performs at a high level and often surpasses the two baselines by a wide margin. This verifies that BigCounter does not require obvious sentiment cues to deliver accurate predictions, as it utilizes its own Polarity-Index: an extensive and diverse library of both obvious and non-obvious sentiment-bearing patterns. On the other hard, both the SVM and RNN baselines perform significantly worse for this dataset, across domains and sentence lengths. Table 3 reports the accuracies of the three classifiers.

Table 3 Summary of the classification accuracies.

Domain	Type	BigCounter	RNN	SVM
Hotel	Easy	0.743	0.654	0.504
Hotel	Hard	0.620	0.443	0.409
Restaurants	Easy	0.759	0.656	0.658
Restaurants	Hard	0.577	0.444	0.436
Books	Easy	0.683	0.736	0.587
Books	Hard	0.671	0.467	0.409

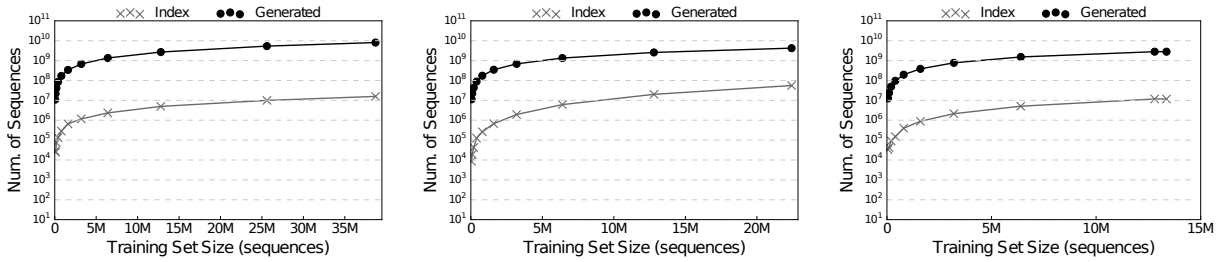
5.3. Testing the Limits of Big Data for Sentiment Classification

In this section, we attempt to contribute to the body of research that examines performance gains from bigger data by focusing on how the size of the training corps affects the of predictive performance of `BigCounter`. Our approach builds on previous work that has utilized data from different domains to conduct a learning curve analysis and gauge the benefits of larger training corpora for predictive tasks.[33,34] In our own evaluation, we examine the effect of increasing the size of the training set on:

1. The training set's linguistic diversity, as encoded by the the number of unique wildcard patterns, as well as by the number of patterns that are accepted into `BigCounter`'s `Polarity-Index`.
2. The predictive performance of `BigCounter`.

We generally expect that adding more data should increase the number of unique and indexed patterns. However, we hypothesize that as the training set gets larger, it becomes harder to locate never-before-seen patterns, and even harder to locate new sentiment-bearing patterns that make it into the `Polarity-Index`. This would lead to a convergence in terms of accuracy, as the algorithm would not have additional ways to classify new sentences. The purpose of this experiment is to verify the existence of such a convergence point. We begin by randomly sampling 50000 sentences to serve as the initial training set. We then iteratively double the size of the training set by adding a new random sample of sentences. After each addition, we re-compute the number of unique patterns, the size of the `Polarity-Index`, and the accuracy of `BigCounter` on both the `SentenceCorpus` and `SentenceCorpus-Hard` corpora.

Linguistic Diversity: We report the results of the first two Figure 5. We observe that, for all three domains, additional data have a major effect on both the number of generated patterns and the size of the index early in the process. This effect dwindles for larger training sets, as large amounts of additional data are required for only small increases in the number of generated patterns. This verifies that the majority of distinct language patterns are already present in smaller datasets. In addition, the index size exhibits the same diminishing returns behavior, with most of



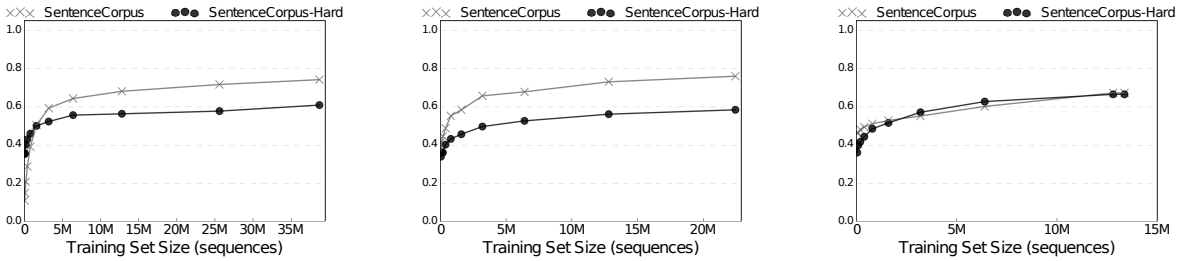
(a) Hotels (b) Restaurants (c) Books

Figure 5 Number of unique patterns and the size of the `Polarity-Index` as a function of the training set size.

the sentiment-bearing patterns having been detected early in the process. This provides us with strong evidence that, with respect to discovering new sentiment-bearing patterns, there exists a threshold above which additional data quickly becomes less valuable.

A second observation is that, for all three domains, there exists a considerable difference (3 orders of magnitude) between the size of the `Polarity-Index` and the number of unique patterns mined from the training set. The same holds for the rate at which these two quantities grow. We conclude that even though there is a great number of patterns in the reviews, the number of sentiment-bearing patterns is a lot smaller and converges a lot faster. This finding implies that the practice of additional acquiring additional data does not deliver additional gains after a certain point in terms of linguistic diversity.

Accuracy: We report the results of the first two Figure 6. We first observe that, as expected, the effect of additional data is most pronounced during the first steps of the learning curve analysis: relatively small increases in the training set size result in large performance gains. Increases in predictive performance persist throughout the learning curve analysis, but soon start to diminish to the point that they become marginal. The fact that this is the case for a highly non-linear and well-performing algorithm such as `BigCounter` reveals that, after some point, additional data has little value, at least with respect to sentiment classification. It is worth noting the overlap of the two curves for the books domain; this is expected due to the fact that reviews from this domain contain less obvious leads, and hence they are more challenging to classify.



(a) Hotels (b) Restaurants (c) Books

Figure 6 Accuracy as a function of the training set size, evaluated on the `SentenceCorpus` and `SentenceCorpus-Hard` corpora.

6. Implications and Directions for Future Work

Our methodology for sentiment classification departs from the standard approach of trying to mathematically explain natural language. Instead, we demonstrate how the ever-increasing complexity of state-of-the-art methods can be replaced by mining weakly annotated big data. Our experimental evaluation against competitive baselines verifies the efficacy of this new and much simpler approach. In addition, we utilize our methodology toward a detailed study on the limits of big data for sentiment classification. Our study is motivated by the hypothesis that, after a certain point, adding more data to the training set does not increase performance. Our findings provide strong evidence in support of this hypothesis, and deliver valuable insight on the connection between data size and performance.

Implications: Our work has implications for practitioners in both academia and industry, as it presents an intuitive alternative to the increasingly complex and computationally expensive algorithms for sentiment classification. In addition to being much simpler while achieving highly competitive results, our approach delivers interpretable predictions that can be easily communicated to managers and decision makers, even if they do not possess an extensive technical background. This is a significant advantage over state-of-the-art algorithms, such as recent advancements in deep learning, that are typically treated as black boxes and mystify non-experts.

Further, our methodology lowers the barrier to entry for firms that want to incorporate sentiment classification into their product or data analysis tasks. This is especially important for firms that

cannot afford the hardware/software infrastructure and talent that is required to train and tune complex computational models. Lowering a firm's dependency on technical talent can be a significant advantage, especially for smaller firms that do not have the resources to be competitive in the ongoing talent wars. Such firms can greatly benefit by our data-over-computation paradigm and utilize data that they already have (or can easily acquire) rather than try to design and implement an algorithmic engine that surpasses their capabilities.

Finally, our study on the limits of Big Data can help managers make informed decisions about how much data their firm needs in order to achieve accurate sentiment-classification results. The ability to make such decisions is valuable, as additional data typically comes with additional acquisition and management costs, measured in both monetary terms and work hours. Therefore, a firm can achieve significant savings by not trying to crunch more data than it actually needs to. Our study can help managers and team leads strategically design their data acquisition efforts by revealing the type of data that they need to acquire (e.g. in terms of the origin domain, vocabulary, polarity) in order to complement their training set, cover previously uncovered cases, and achieve more accurate results.

Directions for future work: Future research can consider applying our approach to different domains. Even though the focus of our work is on sentiment classification, our methodology can also be applied to any document classification task, as well as to the task of labeling specific sentences within a larger document. For instance, consider the problem of assigning topic labels to tweets. Rather than depending on elusive training datasets with tweet-level annotations, a practitioner could utilize our approach on weakly annotated data, such as batches of tweets from users with known topical interests (e.g. we expect politicians to tweet about politics and athletes to tweet about sports).

It is our hope that our findings and methodological contributions will inspire and support relevant research in this domain and will motivate the design of simple but effective algorithms that can mine actionable insights from big data.

References

[1] Anindya Ghose, Panagiotis G Ipeirotis, and Arun Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *annual meeting-association for computational linguistics*, volume 45, page 416, 2007.

[2] Ivan Titov and Ryan T McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer, 2008.

[3] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.

[4] Vivek Kumar Singh, Mousumi Mukherjee, and Ghanshyam Kumar Mehta. Combining collaborative filtering and sentiment classification for improved movie recommendations. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 38–50. Springer, 2011.

[5] Yi Zhao, Sha Yang, Vishal Narayan, and Ying Zhao. Modeling consumer learning from online product reviews. *Marketing Science*, 32(1):153–169, 2013.

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

[7] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

[8] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[9] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[10] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.

[11] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[12] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 327–335. Association for Computational Linguistics, 2006.

[13] Michael Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics, 2004.

[14] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418, 2004.

[15] Cícero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[16] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

[17] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[18] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

[19] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[20] Bryan Catanzaro. Deep learning with cots hpc systems. 2013.

[21] Dong Yu Li Deng. Deep learning: Methods and applications. Technical report, May 2014.

[22] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[23] Jerry Luftman and Rajkumar M Kempaiah. The is organization of the future: The it talent challenge. *Information Systems Management*, 24(2):129–138, 2007.

[24] S Murray. Talent wars: The struggle for tomorrow's workforce. *The Economist*, pages 1–20, 2008.

[25] Tom Stein Tim Devaney. The talent wars: Todays toughest startup challenge. Technical report, 2012.

[26] Boris Groysberg and Deborah Bell. New research: Where the talent wars are hottest. *Harvard Business Review: Blog Network*, 2013.

[27] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[28] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[29] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[30] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[32] Vasant Dhar. Can big data machines analyze stock market sentiment?, 2014.

[33] Brian Dalessandro, Claudia Perlich, and Troy Raeder. Predictive modeling with big data: is bigger really better? *Big Data*, 2(2):87–96, 2014.

[34] Enric Junqué de Fortuny, David Martens, and Foster Provost. Predictive modeling with big data: is bigger really better? *Big Data*, 1(4):215–226, 2013.

[35] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[36] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics, 2004.

[37] Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 111–120. ACM, 2010.

[38] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics, 2006.

[39] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[40] Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. Adapting naive bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval*, pages 337–349. Springer, 2009.

[41] Yousef Alhessi and Richard Wicentowski. Swatac: A sentiment analyzer using one-vs-rest logistic regression. *SemEval-2015*, page 636, 2015.

[42] Vikas Sindhwani and Prem Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *2008 Eighth IEEE International Conference on Data Mining*, pages 1025–1030. IEEE, 2008.

[43] Jonathan Laserson. From neural networks to deep learning: zeroing in on the human brain. *XRDS: Crossroads, The ACM Magazine for Students*, 18(1):29–34, 2011.

[44] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[45] Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212, 2014.

[46] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, 2015.

[47] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*, pages 49–54, 2014.

[48] Zachary C Lipton, David C Kale, Charles Elkan, Randall Wetzell, Sharad Vikram, Julian McAuley, Randall C Wetzell, Zhanglong Ji, Balakrishnan Narayaswamy, Cheng-I Wang, et al. The mythos of model interpretability. *IEEE Spectrum*, 2016.

[49] SAS. Deep learning what it is and why it matters. `http://www.sas.com/en_us/insights/analytics/deep-learning.html`, 2016. Accessed: 2016-08-30.

[50] Mark W Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, pages 24–30, 1996.

[51] Alan B Tickle, Robert Andrews, Mostefa Golea, and Joachim Diederich. The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998.

[52] Thorsteinn Rögnvaldsson and Liwen You. Why neural networks should not be used for hiv-1 protease cleavage site prediction. *Bioinformatics*, 20(11):1702–1709, 2004.

[53] Adele E Goldberg. *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand, 2006.

[54] Michael Tomasello and Michael Tomasello. *Constructing a language: A usage-based theory of language acquisition*. Harvard university press, 2009.

[55] Joan Bybee. *Language, usage and cognition*. Cambridge University Press, 2010.

[56] Norbert Schmitt. *Formulaic sequences: Acquisition, processing, and use*, volume 9. John Benjamins Publishing, 2004.

[57] Alison Wray. *Formulaic language: Pushing the boundaries*. Oxford University Press, 2008.

[58] Alison Wray. Formulaic sequences in second language teaching: Principle and practice. *Applied linguistics*, 21(4):463–489, 2000.

[59] Andrew Pawley and Frances Hodgetts Syder. Two puzzles for linguistic theory: Nativelike selection and nativelike fluency. *Language and communication*, 191:225, 1983.

[60] Geoffrey Underwood, Norbert Schmitt, and Adam Galpin. The eyes have it. *Formulaic sequences: Acquisition, processing, and use*, 9:153, 2004.

[61] Kathy Conklin and Norbert Schmitt. Formulaic sequences: Are they processed more quickly than nonformulaic language by native and nonnative speakers? *Applied linguistics*, 29(1):72–89, 2008.

[62] Joan Bresnan. Linguistic theory at the turn of the century. In *Plenary address to the 12th World Congress of Applied Linguistics. Tokyo, Japan*, 1999.

[63] Koenraad Kuiper. *Smooth talkers: The linguistic performance of auctioneers and sportscasters*. Lawrence Erlbaum, 1996.

[64] John H Holland. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*, 1975.

[65] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[66] Vasant Dhar, Dashin Chou, and Foster Provost. Discovering interesting patterns for investment decision making with glowera genetic learner overlaid with entropy reduction. *Data Mining and Knowledge Discovery*, 4(4):251–280, 2000.

[67] David E Goldberg, Kelsey Milman, and Christina Tidd. Genetic algorithms: A bibliography. *IlliGAL Report*, 92008, 1992.

[68] Nan Hu, Jie Zhang, and Paul A Pavlou. Overcoming the j-shaped distribution of product reviews. *Communications of the ACM*, 52(10):144–147, 2009.

[69] John Rice. *Mathematical statistics and data analysis*. Nelson Education, 2006.

[70] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[71] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[72] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics, 2011.

[73] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[74] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, 21(8):879–886, 2008.

[75] Dino Isa, Lam H Lee, VP Kallimani, and Rajprasad Rajkumar. Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge and Data engineering*, 20(9):1264–1272, 2008.

[76] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.