Behav Res (2016) 48:909–921
DOI 10.3758/s13428-016-0766-5

Efficient *n*-gram analysis in R with `cmscu`

David W. Vinson[1] · Jason K. Davis[1] · Suzanne S. Sindi[1] · Rick Dale[1]

Published online: 5 August 2016

Abstract We present a new R package, `cmscu`, which implements a Count-Min-Sketch with conservative updating (Cormode and Muthukrishnan *Journal of Algorithms, 55*(1), 58–75, 2005), and its application to *n*-gram analyses (Goyal et al., 2012). By writing the core implementation in C++ and exposing it to R via Rcpp, we are able to provide a memory-efficient, high-throughput, and easy-to-use library. As a proof of concept, we implemented the computationally challenging (Heafield et al., 2013) modified Kneser–Ney *n*-gram smoothing algorithm using `cmscu` as the querying engine. We then explore information density measures (Jaeger *Cognitive Psychology, 61*(1), 23–62, 2010) from *n*-gram frequencies (for $n = 2, 3$) derived from a corpus of over 2.2 million reviews provided by a Yelp, Inc. dataset. We demonstrate that these text data are at a scale beyond the reach of other more common, more general-purpose libraries available through CRAN. Using the `cmscu` library and the smoothing implementation, we find a positive relationship between review information density and reader review ratings. We end by highlighting the important use of new efficient tools to explore behavioral phenomena in large, relatively noisy data sets.

Keywords Information theory · Sketch algorithms · *n*-grams · Big data · Interdisciplinary collaboration

✉ David W. Vinson
dvinson@ucmerced.edu

[1] University of California, Merced, 5200 N. Lake Rd., Merced CA, USA

Introduction

"Big data" collection and analysis are now at the forefront of modern science and business, with daily data collection equal to that of 90 % of all data collected in the past 2 years (McAfee et al., 2012), which comprises over 2.7 zettabytes (10^{21} bits). Keeping pace with the scale and speed of modern data collection necessitates the development of computational tools capable of efficiently analyzing larger and larger data sets. Hardware has rapidly evolved to enable such large-scale computations; for example, the time to assemble an entire human genome, just under one week, once took two years (Sagiroglu & Sinanc, 2013). To maximize the utilization of modern hardware, however, calculations must typically be expressed in "low-level" languages such as Fortran or C++, more or less directly exposing the hardware to the programmer at the cost of code simplicity and clarity. More expressive scripting languages such as R and Python allow scientists to more directly express their calculations and theories in a hardware-agnostic way, but at the (sometimes significant) expense of code runtime and not being able to immediately leverage the latest hardware advances.

Because of this, tools that facilitate analysis of large data sets could greatly accelerate research in behavioral science. Many (if not most) behavioral scientists are trained only in scripting languages and the relevant statistical packages. Thus the analysis of larger data sets falls to computer scientists and engineers who often lack the background and training in the behavioral sciences. Further integrating these fields through big data and analysis tools has much promise within both research and applied domains. For example, data sets continue to be released to the public

with companies benefiting from 'dataset challenges' that crowd source solutions to computational problems. Netflix famously created a dataset challenge paying out one million dollars to any team who could improve their current recommendation system by 10 %. Yelp Inc. continues to release more and more reviews from their database, and pays out $5,000 to students who simply use the data in interesting ways. In fact, there are entire websites dedicated to advertising dataset challenges (e.g., Kaggle.com). The problem is that behavioral science training rarely includes efficient programming techniques to harness the raw power of these larger data sets, often sacrificing computational efficiency for ease of programming. In order to obtain insights from larger data sets, behavioral scientists—with their important domain-specific knowledge—must be able to engage with ever larger data sets in a meaningful way.

The scripting language `R` is the preferred computational and statistical language of many behavioral scientists, having rich documentation and an accessible programming environment. However, it is not very performant relative to other solutions (Simmering, 2013). Nonetheless, the use of high-level scripting languages, and in particular `R`, has been encouraged for many current and past research agendas. Historically, this has not been a problem as the size of one's data set and complexity of analyses have typically been guided by highly controlled experimental paradigms. Such data sets require very little computational power to uncover phenomena from few participants. However, the increasing size and number of freely available data sets, as well as the desire to provide ecologically valid results (Roy et al., 2015; Lazer et al. 2009), challenges the efficacy of this trade-off. Indeed, being capable of harnessing large data sets will also help to accommodate the current needs of the behavioral sciences, such as openness and replicability (Nosek et al., 2012; Schmidt, 2009; Pashler & Wagenmakers, 2012; Zwaan & Pecher, 2012), without loss of scientific productivity (Ramscar et al., 2015). This new demand requires new tools that can be easily implemented by behavioral scientists trained to uncover interesting behavioral phenomena.

In this paper, we introduce the application of a prominent data-reduction technique for handling massive amounts of text. This technique permits efficient approximation of text information from a very large corpus. The `R` library we introduce, `cmscu`, could thus open new avenues of analysis for behavioral scientists, but the library and its application also recommend some broad methodological lessons. We aim to elucidate three key methodological observations. First, in "R-package `cmscu`", we describe how so-called "sketch" techniques help process large amounts of data while efficiently using memory resources and argue these are critical for the coming 'big data' age in the behavioral sciences (Griffiths, 2015). Next, in "Information-Theoretic structure of yelp reviews", we demonstrate a fruitful domain in which such a strategy can apply—the Information-Theoretic analysis of language structure in corpora. As an example, due to the efficiency now available with `cmscu`, we quantify the structure of a language using the sophisticated *modified Kneser–Ney smoothing* algorithm (Kneser & Ney, 1995). Finally, using a dataset from Yelp, Inc., we show that our library and its use in implementing sophisticated Information-Theoretic algorithms permit wide ranging exploration of the statistical properties of language use and its relationship to other behavioral phenomena. In the "General discussion", we revisit important messages about cross-disciplinary interactions and suggest that adopting a wide range of tools from various disciplines can help to ensure that behavioral scientists find efficient solutions for their problems, while giving computational scientists and engineers exciting behavioral problems in which to apply their research.

Here, we briefly motivate why adopting more efficient techniques is crucial to maintaining the current pace of progress in the behavioral sciences. To do this we demonstrate its application in one of the most common quantitative models of language: n-grams.

***n*-gram models of language** The study of language is a key topic in the psychological sciences, influencing and being influenced by a whole host of psychological factors at many scales including: vision (Tanenhaus et al., 1995), emotion (Nygaard & Queen, 2008; Pennebaker et al., 2001; Pennebaker, 1997; Jurafsky et al., 2014; Kahn et al., 2007), the community (Vinson & Dale, 2016; Lupyan & Dale, 2010), individuals who make up that community (Nygaard et al., 1994; Nygaard & Pisoni, 1998; Bradlow et al., 1999), gender and social status (Labov, 1972b; 1972a; Kuhl et al., 1992; Lindblom, 1990) and of course much more. n-gram models represent one of the earliest and most-used tools to uncover the statistical structure of language use. In its most basic form, an n-gram is a sequence of n items from a given collection of text, transcribed speech, genomic data, and so on (Li et al., 2001). Having originated in the early 20th century (Markov, 1913), the resurgence of n-gram models in language analysis came about in the mid 1970s and 1980s from their successful use in speech recognition systems (Jelinek, 1976; Baker, 1975; Bahl et al., 1983; Martin & Jurafsky, 2000). Such systems were heavily influence by the work of Claude Shannon whose proposed theory of communication, Information Theory (Shannon, 1948), is among the most influential frameworks of the 20th century. Shannon's key examples of Information Theory involved a n-gram analysis of written English. Information Theory posits that a word carries some *amount* of information, measured in *bits*, proportional to its $-\log_2$ probability of occurrence given some "context" (e.g., a corpus of words):

$$I(w_i) = -\log_2 p(w_i|w_1, w_2, ...w_{i-1}). \quad (1)$$

In Eq. 1 above, the number of bits in word $I(w_i)$ is dependent on the frequency of its occurrence *after* some other word(s), or its maximum likelihood estimate. In the behavioral and computational sciences, this measure and related measures are very useful for exploring language production and processing. However, computing reliable estimates of these measures requires the analysis of massive amounts of text. In addition, it may be very useful to compute these measures over ad hoc data sets of psychological relevance (e.g., education, business, etc.). Doing so requires flexible tools that allow behavioral researchers to estimate such measures in their existing computational environment. This would greatly enhance the availability of such Information Theory concepts.

Currently, programs and methods used by psychologists, specifically the `tm` package in `R`, do not scale well with increasingly large data sets or longer n-grams, thus leaving a powerful tool out of reach. Below we detail the development and use of a new n-gram package, `cmscu`, developed in part to analyze a large Yelp, Inc. dataset previously intractable with the standard `tm` package and its `DocumentTermMatrix` object in `R`. Following its description, we provide the results of an n-gram analysis motivated by current and ongoing research surrounding Information Theory.

R-package `cmscu`

The analysis of n-grams requires two fundamental operations: store and query. Given an n-gram ω, we must be able to store and update the count associated with ω, $store(\omega)$, and query the same count, $query(\omega)$. There are many possible ways to implement such functionality; however, the large nature of our intended dataset limits the feasibility of most standard approaches. In order to be fast, the data structure (or at least its "active parts") should be able to reside entirely within local memory (RAM). This means that we cannot fully store the actual n-grams themselves, but rather a compressed representation of them.

The need for compression suggests a hash table implementation, "compressing" strings by representing them as a single integer *computed* from the string itself (Cormen, 2009). For example, we may use a look-up table for a trigram such as `the_cannibal_consumes` by referencing a single integer—an index in an array—thus saving significant space in the frequency table. However, such hash tables store the string itself along with its associated data in order to resolve possible hash collisions, where two distinct strings map to the same integer—a necessary consequence of the mathematical pigeon-hole principle. As the size of the corpus increases, the space required to store all strings will exceed the size of RAM thus requiring swapping memory from the hard disk, which is often the single largest slowdown in large programs.

By relaxing the "correctness" of our stored value (in a controlled way) a host of fast and efficient algorithms become available. So-called streaming or sketch algorithms rely on *probabilistic* guarantees of accuracy: in this framework, $query(\omega)$ will return a value according to a confidence interval rather than the fixed, precise number. We chose the Count-Min-Sketch with conservative update procedure (CMS-CU) (Cormode & Muthukrishnan, 2005) for its simplicity in implementation and its proven effectiveness (Goyal et al., 2012). This approach has wide applications, and approximate or "sketch" algorithms in general have become quite common in the realm of computational and data science for summarizing and performing other computations on massive and real-time data (Cormode & Muthukrishnan, 2011), the most common example being the conceptually related Bloom filter (Song et al., 2005). By treating a corpus as a stream of text data, an approximate algorithm such as Count-Min-Sketch can provide a estimate of n-gram frequencies up to some desired accuracy depending on core parameters in the algorithm—namely the size of the table that will provide the information for the estimation of frequencies. This allows a researcher to *choose a balance* between probabilistic guarantees and available computational resources, effectively trading off statistical power for the ability to study larger datasets *on the same hardware*.

The CMS-CU sketch algorithm works in the following way: rather than representing some n-gram in a traditional hash table, containing an entry for each distinct n-gram, the count data of the n-grams are tabulated in the stream of text using a w-by-d table ("width and depth"). Every n-gram receives an entry on each row of this table, and the particular entry in each row is determined by a statistically independent hash function. Storing an n-gram consists of incrementing its associated value in each row by 1, while querying it consists of taking the *minimum* of each associated value. The conservative update limits the increment to only those entries that equal the minimum value. The fundamental idea is that if a collision of two strings under one hash function is rare, the simultaneous collision of the same two strings under two independent hash functions is extremely rare. Thus, with additional rows, it becomes increasingly unlikely that the minimum value assigned to an n-gram will over-estimate the true count of its occurrence.

The hash collision probability (or rather, the probability of colliding across all hash functions) is given in the original Cormode and Muthukrishnan (2011) paper, and is philosophically just a generalized birthday problem calculation. The more complicated calculation (and more pertinent) is, however, the probability that the count associated with a string is incorrect (too high). This could occur from a

Table 1 Methods for `cmscu`

dict <- new(FrequencyDictionary, d, w)	Initialize a dictionary with `d` rows ($d = 4$ gives > 98 % confidence) and w bins
dict$store('string')	Update the frequency count associated with string `'string'`
dict$store(c('s1', 's2', ...))	Update the counts associated with all of the strings simultaneously
dict$query('string')	Query the (approximate) frequency count associated with `'string'`
dict$query(c('s1', 's2', ...), n=1)	Query the counts of all the strings simultaneously over `n` OpenMP threads (if available)

second string colliding across all the hash tables, but more likely from multiple distinct strings each individually colliding on only a few hash tables, but collectively resulting in a net increase over all the entries for the original string. That this probability is bounded and easily estimated is what makes (Cormode & Muthukrishnan, 2011) so notable here. Below we present their confidence interval bounds. Matching intuition, increasing the number of entries per hash table achieves first-order reduction in the width of the confidence interval, while increasing the number of hash functions increases our confidence as $1 - \mathrm{e}^{-d}$.

The memory utilization of this approach offers a significant advantage—the storage of a string requires 1 byte per character, thus $(n + 1)$-grams will take more space than n-grams in memory. However, the Count-Min-Sketch offers a fixed memory data structure—assuming 4 bytes per entry (a 32-bit integer), it will consume $4 \times w \times d$ bytes in memory *independent* of the dataset being studied. When used to estimate probabilities (rather than integer counts), we have the confidence interval

$$Pr\left[p_i \leq query(\omega_i)/N < p_i + \epsilon\right] > 1 - \mathrm{e}^{-d}, \tag{2}$$

where N is the number of items stored (including repeats), p_i is the true empirical frequency of ω_i and $\epsilon \approx \mathrm{e}/w \propto 1/w$ (Cormode & Muthukrishnan, 2005).

We implemented this algorithm in `C++` using a reference MurmurHash3 hash implementation coupled with a pairwise hashing optimization (Kirsch & Mitzenmacher, 2006), and then exported our class to `R` via `Rcpp` (Eddelbuettel et al., 2011). Writing the core implementation in `C++` allows for precise, efficient, and predictable memory utilization, while the `Rcpp` binding allows for its convenient use via the `R` scripting language.

Usage

The `cmscu` library has only a few methods that wrap the entire functionality (see Table 1). We describe the three primary methods below, and refer the reader to the GitHub page[1] for the full documentation and package.

Sample usage is given below:

```
dict <- new(FrequencyDictionary, 4, 10^6); # 4 is the number of hash functions (d) and 10^6 is the width (w) # Total size (in bytes) = 4 x w x d

bigrams <- c('this is', 'is sample', 'sample usage'); dict$store(bigrams);

test <- c('this is', 'not present', 'sample usage', 'this is'); counts <- dict$query(test); # counts is c(1,0,1,1)
```

Comparison

We compare the application of our library to that of the `tm` package (Meyer et al., 2008), a common text-mining package in R. This package is frequently recommended for the analysis of n-grams in `R` specifically for its `DocumentTermMatrix` class, which counts the occurrences of each string-type per document and stores the integer values into a large matrix. It is this functionality that `cmscu` specifically offers an efficient alternative to—`tm` also provides high quality text-processing utilities that we do not attempt to replicate.

Perhaps unsurprisingly, by tailoring our data structure to the problem at hand, `cmscu` outperforms `tm`'s `DocumentTermMatrix` by orders of magnitude in the task of n-gram frequency analysis and information density calculation. The reason for this is that the `DocumentTermMatrix` solves a more general problem, which requires the storage and organization of data unnecessary for our calculation and prevents a linear scaling in the size of the corpus. While it is a powerful tool for document classification, and bundles useful functionality for cleaning and preparing raw strings, it is not optimal for the specific task of n-gram information density computation (despite its usefulness for such tasks with much smaller data sets).

We benchmark the creation, initialization, and evaluation of increasingly large datasets (the first k lines from the Yelp Inc. dataset) averaged over 10 runs in Fig. 1. We run our CMS-CU implementation with 4 rows of 2^{24} entries, using a fixed 1GB of RAM for each run. For small data sets, the cost associated with this unnecessarily large memory allocation outweigh the `tm` calculation; however, as we approach 10^4 lines of the Yelp dataset, we are 18 times faster

[1] http://www.github.com/jasonkdavis/r-cmscu.

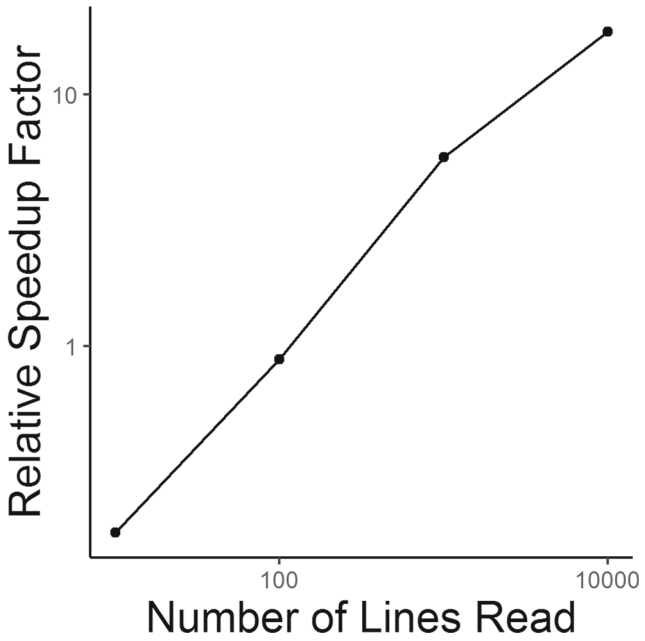

Fig. 1 Log-log plot of the calculation time of `tm` relative to the calculation time of our package, averaged over ten runs, for increasingly large data sets. Due to the nonlinear scaling of `tm`, our implementation becomes increasingly faster relative to `tm` as the dataset increases in size. The memory requirements of `tm` prevented the comparison of larger data sets

in total run-time. For larger sizes, we were unable to even run the reference `tm` code due to the non-linear scaling in its algorithmic complexity and memory use.

For a dataset of k lines, we compute k values with the `tm` result being exact and our CMS-CU result being approximate. We compute the root-mean-squared (RMS) error of our calculation in Table 2, where

$$RMS(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_i - y_i)^2}. \quad (3)$$

When $k = 10^1$ and $k = 10^2$, there is no difference (machine precision) in the output due to our large allocation which suggests the absence of any hashing collisions. As k increases, collisions expectedly occur but the average information density, $-\frac{1}{n}\sum_{i=1}^{n} log_2 p(w_i)$, of a review maintains a precision of 10^{-3}. This could be improved by using additional rows in the CMS-CU structure (going from 4 to 6, for example).

Table 2 Root-mean-squared error of CMS-CU output over increasingly large datasets

k	10^1	10^2	10^3	10^4
RMS	2.22×10^{-16}	4.31×10^{-16}	7.30×10^{-3}	8.75×10^{-3}

Information-Theoretic structure of yelp reviews

The development of efficient n-gram storage and querying techniques affords analyzing text using more sophisticated algorithms. Standard practice involves training a model on held out data and using it to predict test data. However, it is almost always the case that unseen n-grams will occur in test data, resulting in poor model performance. Applying smoothing techniques adjusts n-gram probability estimates to help account for missing data and increase model performance (i.e., decrease model perplexity).[2] The term 'smoothing' comes from the fact that these algorithms tend to make the distribution more uniform by adjusting low probabilities upward and high probabilities downward (Chen & Goodman, 1999).

To date, the most accurate models estimate the maximum likelihood of an n-gram using both higher- and lower-order n-grams. Two types of models exist: (1) *Back-off* models use lower order n-grams (such as bigrams and unigrams) to estimate the maximum-likelihood of higher order n-grams (such as trigrams), but only when data from the higher-order n-gram is missing. Thus it *backs off* to the lower-order $(n-1)$-gram until the value is defined. (2) *Interpolated* models use estimates from lower-order n-grams even when higher-order n-grams are defined. Interpolated models are defined recursively as a linear interpolation between the nth-order maximum likelihood model and the $(n-1)$-th order smoothed model (Chen & Goodman, 1999, p. 364). The most accurate interpolation models penalize higher- and lower-order n-grams using a *discount* parameter for higher n-grams and a *smoothing* parameter (often defined in part by the count of the higher-order n-gram) on lower order n-grams. It is reasonable to use both lower and higher-order n-grams together when estimating higher-order n-grams because the frequency of the $(n-1)$-gram will typically correlate with the n-gram and has the advantage of being estimated from more data. For this reason, such models tend to accurately estimate unseen data. Crucially, both types of models use lower-order n-grams to estimate *missing* higher-order n-grams (essentially interpolated models are back-off models when n-grams are missing). Similarly, in the occurrence of zero count data, where w_i is never seen (at any n- or $(n-1)$-gram) the standard procedure across both models, and one we adopt here, is to estimate its value via a uniform distribution, $p_0(w_i) = 1/|v|$, where $|v|$ is the model's vocabulary (the number of unique 1-grams in the training data).

[2] The accuracy of specific Information-Theoretic models on estimating unseen data that vary in the length of n or the complexity of the algorithm can be determined by measuring its cross-entropy or more specifically model *perplexity* (Bahl et al., 1983; Jelinek et al., 1977; Jurafsky & Martin, 2000).

At present, the most accurate smoothing technique is a variation on what is known as Kneser–Ney smoothing (Kneser & Ney, 1995). What makes Kneser–Ney smoothing more accurate than other smoothing techniques is how it estimates unigrams (1-grams). While other models simply take the frequency of the unigram, Kneser–Ney smoothing estimates this value based off how likely w_i is to occur in an unfamiliar context:

$$p_{KN}(w_i) = \frac{|\{w_i : 0 < c(w_{i-1}, w_i)\}|}{|\{(w_{i-1}, w_i) : 0 < c(w_{i-1}, w_i)\}|}, \quad (4)$$

where $|\{w_i : 0 < c(w_{i-1}, w_i)\}|$ is the total number of bigrams w_i completes, divided by the total number of unique bigrams $|\{(w_{i-1}, w_i) : 0 < c(w_{i-1}, w_i)\}|$. A common example used to explain why this is a better estimate of higher-order n-grams is to consider the unigram 'Francisco'. Suppose 'Francisco' occurs frequently throughout our dataset. However, it only ever occurs after the word 'San'. Because it is unlikely to occur in any other bigram context, its unigram probability should not be high as this would result in an inflated probability estimate for 'Francisco'.

We use a version of Kneser–Ney smoothing, *interpolated Fixed Modified Kneser–Ney* (iFix-MKN), to estimate conditional trigram and bigram probabilities (Maximum Likelihood Estimation). Our model is a direct implementation of that found in Chen and Goodman (1999, section 3). Kneser–Ney smoothing uses both a discounting parameter which subtracts some value from non-zero count n-grams, and a smoothing parameter that maximizes the probability of obtaining an accurate estimate from lower-order n-grams. Crucially, our implementation is *fixed* in that both the discounting parameter and smoothing parameters are defined prior to applying the model to test data. Both parameters can be more accurately defined when estimated using held out data prior to analysis. While models that estimate their parameters outperform their fixed counterparts, iFix-MKN outperforms all other smoothing techniques (including those that use held out data to estimate parameters), with the exception of the version of MKN that uses estimated parameters. As a result, iFix-MKN has the benefit of being both simpler and more domain-general. In what follows we leverage an implementation of iFix-MKN using our `cmscu` package to explore and quantify language use in a large natural dataset. The implementation is freely available on the `cmscu` website.

Current study

Our analyses are motivated by recent studies that show a message's Information-Theoretic structure is influenced at a variety of linguistic levels including syntactic variation and phonetic reduction (Aylett, 1999; Genzel & Charniak, 2002; Aylett & Turk, 2006; 2004; Levy & Jaeger, 2006; Jaeger, 2010; Mahowald et al., 2013). Specifically, the amount of information present across a message is shown to increase over time, abiding by the *entropy rate constancy* principle—a message's information density increases at a stable rate (Genzel & Charniak, 2002)—perhaps in an effort to provide relevant content against channel noise. That is, the cognitive agent works to structure their utterance in a way that maintains the highest rate of information without breaking channel capacity (determined by the amount of noise in the system). When at risk of breaking channel capacity, language users might add optional low-information words to high information-dense messages (Jaeger, 2010), or simply slow down their utterance (Aylett & Turk, 2006; 2004) effectively spreading an otherwise information-dense message over a longer utterance. These findings suggest that language users are sensitive to channel noise and adjust their utterances to match the channel's capacity in an effort to balance redundancy with confidence in signal transmission. To do this the theory of Uniform Information Density (UID), building off previous work (e.g., the smooth signal redundancy hypothesis: Aylett and Turk (2004)), suggests language users try to avoid "peaks and troughs" in information density. As a result, language users exhibit an inverse relationship between language redundancy and predictability in an effort to communicate efficiently.

Theories such as UID posit that the information density of one's message may be attuned to the expectations the producer holds about their intended audience. One possibility is that a more established common ground due to a larger number of shared experiences may result in a decrease in channel noise which affords more complex information-dense language use Clark and Brennan (1991). Indeed, many studies have shown that language users make assumptions about their audience and structure their own utterances with these assumptions in mind (Brennan & Williams, 1995, e.g., among many: Jaeger, 2013; Krauss & Fussell, 1990; Pate & Goldwater, 2015). For example, the information density of a language user's Yelp reviews is higher when their network of friends is more densely interconnected (Vinson & Dale, 2016). Similarly, microblog posts on Twitter about specific events, such as a baseball game series, are more information-dense toward the end of a sporting event than during (Doyle & Frank, 2015). Such findings add to growing evidence that language users are sensitive to the knowledge they share with their audience. This sensitivity is reflected in how they structure their utterances.

Method Few studies show directly how one's audience perceives the helpfulness of a producer's language use. In this example application of `cmscu`, we explore in what ways the information density of a Yelp user's review, estimated

via iFix-MKN, can predict how useful, funny and/or cool (U/F/C) it is to its reader. The `cmscu` package greatly facilitates these exploratory analyses of natural data sets. With it we are able to efficiently process massive amounts of text from the Yelp, Inc. Dataset Challenge corpus using one of the most sophisticated algorithms to date.

We estimated a total of five measures from Yelp review text: two information density measures and two uniformity measures as well as review length. We even estimate information measures over trigrams for this analysis, which `cmscu` permits with great flexibility. As this section serves only as an example analysis, we detail these measures in the Appendix below. The measures are based on analysis of bigrams, and trigrams using the iFix-MKN algorithm requiring estimation of conditional probabilities in Yelp text to the second order (trigrams). These measures are summarized briefly in Table 3.

The standard deviation of each information density measure was taken as the measure of variance (inverse uniformity). In addition to the four measures in Table 3, we included review length, giving us five variables used to predict U/F/C. Increasing n quantifies a successively more multiword estimation of information density and uniformity in language use. Given the nature of iFix-MKN, such that in the event of missing n-grams the model backs off to a lower-order n-gram, in our analysis of the Yelp dataset the correlation between trigram information and bigram information is high, $r = .93$, $t(1.03\text{ mil}) = 2527$, $p < 2.2\text{e-}16$. For this reason, we predict U/F/C using two separate models, one including bigram information/variance and length and the other including trigram information/variance and length. Information measures become more and more sparse as we increase model complexity and so there is a trade-off in using different models. Estimating higher-order n-grams gives a better measure of the actual structure of the language, while estimating lower n-grams provides a better model fit. Moreover, when modeling real language use, such as that from the Yelp community, it can vary widely from one review to the next leaving even very well trained models weak predictors. In such cases, estimating lower-order n-grams may prove to be a stronger predictor of behavioral phenomena. Finally, our variance measures will correlate with information density measures due to the presence of a true zero in information density. This will inflate the variance accounted for within our regression models. We adjust for both issues by first predicting each uniformity measure by its information density measure using linear regression (`lm`) in `R`, and then taking the residual of that model as our true estimate of information uniformity; $r\sigma(BI)$, $r\sigma(TI)$.

Higher-order n-gram models are more computationally expensive to estimate using iFix-MKN. However with `cmscu` it is relatively straightforward and easy to deploy in `R`. Though the precise n-gram model details are outside the scope of this example application of `cmscu`, we offer the reader a breakdown of our variables and modeling approach in the Appendix. This will also serve as an example strategy for deploying `cmscu` in more detail.

Though simply an example application, prior research motivates some predictions: (1) *Information density will be positively related to U/F/C ratings*. The language use within reviews is most likely already abiding by the constraints of its community. For this reason, information-dense messages should hover closer to the channel's capacity, thus providing more helpful content without too much risk of being misunderstood. (2) *Information variance will be positively related to U/F/C*. Increased variance may be associated with higher reader ratings. Specifically, more variance in information across a message may be related to higher reader ratings as it may suggest reviewer's are inserting low information-dense words to lower the risk of presenting information that may be misunderstood.

Results In all cases, a negative binomial model predicted, perhaps surprisingly, a large portion of variance. Tables 4 and 5 report the 95 % Confidence Intervals (CI_β) and associated Z-scores for each predictor variable as well as the overall R^2_{adj} for each model containing bigram and trigram measures respectively. There were no differences between the trends in either bigram or trigram model predictors. That is, all measures were highly significant positive predictors of U/F/C ratings, such that an increase in review length, information density and variance increased the probability of a review receiving more U/F/C ratings. In order, review length (*Log-Length*) was the strongest predictor followed by information density measures (*ABI* and *ATI*) and last the variance of information density, $r\sigma(BI)$ and $r\sigma(TI)$. Because there was little difference between trigram and bigram models and because trigrams are considered a stronger estimate of the structure of the language itself, Figs. 2, 3, 4 present only the trigram model's predicted U/F/C ratings by (A) *Log-Length*, (B) *Average Trigram Information* and (C) *Variance of Trigram Information* respectively.

Table 3 Summary of measures from information theory

n	Density measure	Uniformity measure
2	Average (iFix-MKN) Bigram Information: ABI,	Variance of Bigram Information: $\sigma(BI)$
3	Average (iFix-MKN) Trigram Information: ATI,	Variance of Trigram Information: $\sigma(TI)$

Table 4 Bigram by reader rating negative binomial model

Rating type	Predictor	95 %(CI_β)	Z-value	Effect size
Useful	$Intercept$	(−.169, −.163)	−104.4*	R^2_{adj} = .15
	Log-Length	(.585, .591)	367.0*	
	ABI	(.175, .182)	113.3*	
	$r\sigma(BI)$	(.040, .043)	24.5*	
Funny	$Intercept$	(−1.121, −1.111)	−425.6*	R^2_{adj} = .09
	Log-Length	(.647, .657)	254.1*	
	ABI	(.355, .365)	143.6*	
	$r\sigma(BI)$	(.075, .086)	30.8*	
Cool	$Intercept$	(−.828, −.819)	−365.8*	R^2_{adj} = .09
	Log-Length	(.616, .625)	275.5*	
	ABI	(.197, .206)	90.3*	
	$r\sigma(BI)$	(.102, .113)	46.7*	

*p < 2e-16, *r is the residual from* lm *models described in text*

Overall this exploratory analysis suggests that the composition of language use, in terms of Information-Theoretic structure, significantly predicts the impression of a review from its readers. These exploratory analyses open up interesting avenues for future research on language use. For example, showing that the variance of information is related to reader ratings as well as information density measures may suggest readers are more likely to find both high information and decreased channel noise useful for comprehension. Future research interested in understanding what aspects of online reviews readers find useful, funny or cool should be sure to investigate various other linguistic features, such as simpler lexical-level variables (e.g., curse words), which might be highly correlated with more sophisticated information measures presented here. This is outside of the scope of the current demonstration, but the tool we introduce here may permit such analyses efficiently.

General discussion

Efficiently processing large amounts of textual data is a problem at the forefront of behavioral science. One way to advance this process, as we demonstrate here, is by adapting well-known tools from computer science by developing statistical packages in programs used by behavioral scientists. This effectively broadens the number of tools that can be used by behavioral scientists while redefining the problem space wherein that tool is applicable. Here we use a sketch algorithm known for its efficiency in processing

Table 5 Trigram by reader rating negative binomial model

Rating type	Predictor	95 %(CI_β)	Z-value	Effect size
Useful	$Intercept$	(−.167, −.161)	−102.9*	R^2_{adj} = .15
	Log-Length	(.590, .597)	371.7*	
	ATI	(.161, .167)	102.4*	
	$r\sigma(TI)$	(.043, .050)	28.5*	
Funny	$Intercept$	(−1.118, −1.108)	−424.4*	R^2_{adj} = .09
	Log-Length	(.653, .663)	256.2*	
	ATI	(.352, .362)	140.8*	
	$r\sigma(TI)$	(.062, .073)	25.8*	
Cool	$Intercept$	(−.825, -.816)	−364.4*	R^2_{adj} = .09
	Log-Length	(.625, .634)	278.8*	
	ATI	(.175, .183)	79.3*	
	$r\sigma(TI)$	(.114, .123)	51.4*	

*p < 2e-16, *r is the residual from* lm *models described in text*

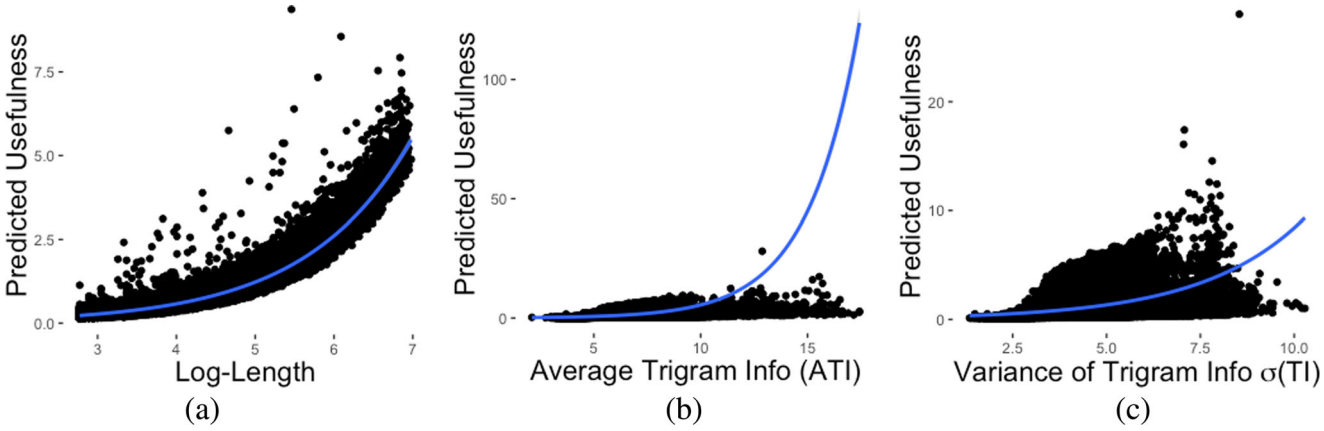

Fig. 2 Predicted Usefulness ratings by **a** Log-Length, **b** Average Trigram Information (ATI) and **c** Variance of Trigram Information $\sigma(TI)$. Predictions are provided by the full negative binomial model controlling for other variables

massive real-time data (Cormode & Muthukrishnan, 2011) to approximate the information density of words, using a sophisticated smoothing algorithm (iFix-MKN), across millions of online reviews. We show it to be a successful tool toward discovering interesting behavioral phenomena.

Our analysis shows the `cmscu` package can be used to process n-gram data at speeds and scales beyond the reach of commonly available `R` packages. In real-world applications, such as our Yelp data analysis, which used a modified Kneser–Ney implementation powered by the `cmscu` library, was able to process 5 % of the 2.2 million reviews we evaluate in under a hour, on a single core, on commodity hardware. Our analysis was configured to evaluate up to quadgrams, requiring 8 separate `cmscu` instances each configured to occupy 1gb of RAM. A much earlier (attempted) analysis on the dataset, built upon `tm` using its `DocumentTermMatrix` object, had run for over 2 months without finishing—the algorithmic and memory scaling requirements resulted in constant swapping of hard drive space, which effectively brought the computations to a standstill. Though limited in features compared to `tm`, `cmscu` is simple to use, requires few lines of code, and scales with the processing power of one's computer and size of one's dataset. That is, users may specify memory usage *a priori*, in line with their hardware's capabilities, and nonetheless obtain a useful analyses, independent of the size of the dataset under study.

Using `cmscu` we explore possible relationships among reader ratings and sophisticated estimations of Information-Theoretic structures of review text in a large Yelp, Inc. dataset, a process difficult for common scripting packages. Indeed, the sheer size of our dataset affords the possible discovery of subtle, but interesting relationships such as those between the linguistic choices of language users and their audience. For this reason our predictions are inherently exploratory. Though our predictions are broad, and somewhat intuitive perhaps, our findings build on previous work that shows language users structure their utterances with their intended audience in mind (Jaeger, 2013; Clark & Brennan, 1991; Brennan & Williams, 1995).

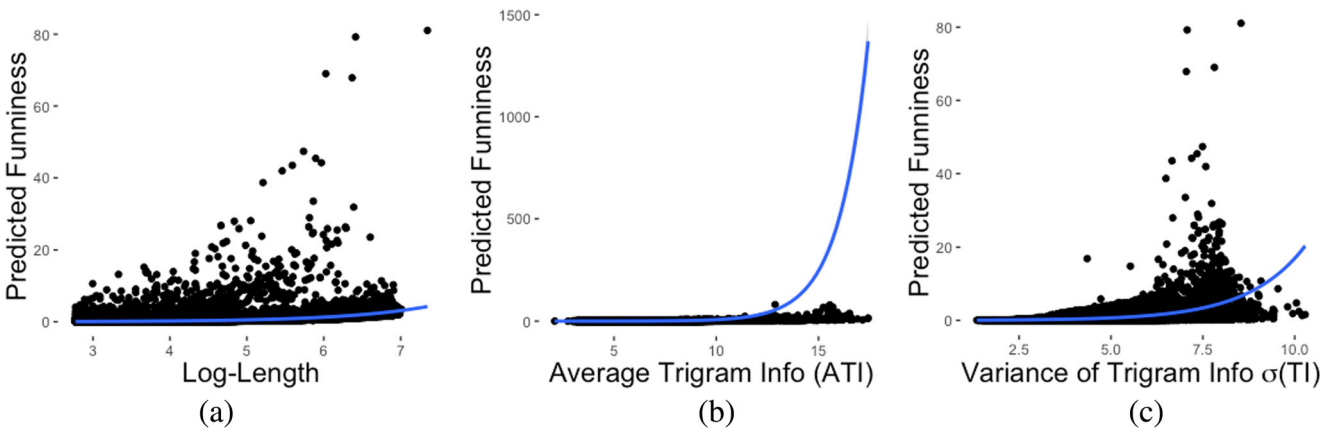

Fig. 3 Predicted Funniness ratings by **a** Log-Length, **b** Average Trigram Information (ATI) and **c** Variance of Trigram Information $\sigma(TI)$. Predictions are provided by the full negative binomial model controlling for other variables

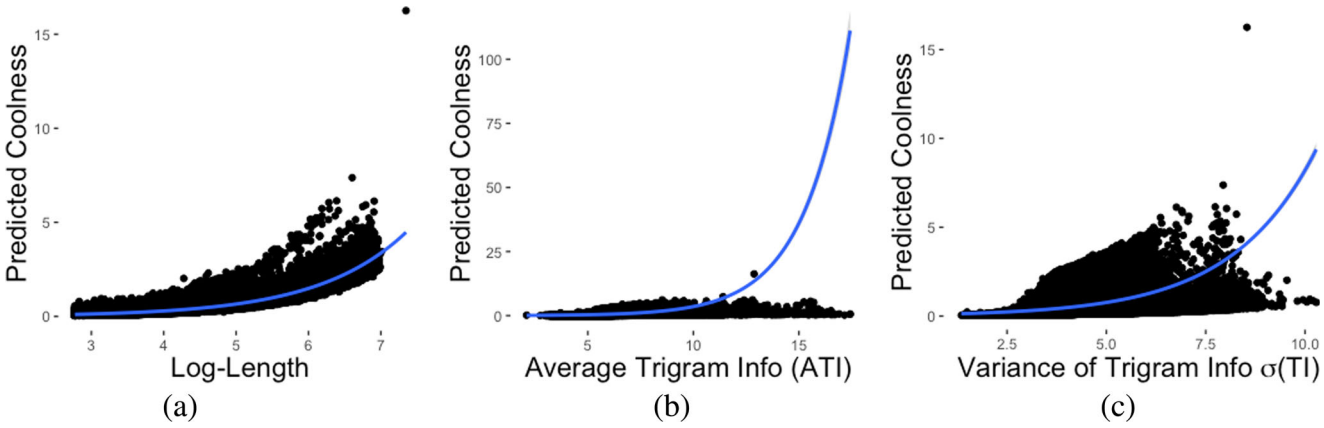

Fig. 4 Predicted Coolness ratings by **a** Log-Length, **b** Average Trigram Information (ATI) and **c** Variance of Trigram Information $\sigma(TI)$. Predictions are provided by the full negative binomial model controlling for other variables

We speculate that reader ratings mark the successful transmission of some useful information by its producer. If so, longer reviews, those with more information density and those with greater variance in information are more likely to be successfully transmitted. Interestingly, one possibility is that higher reader ratings are more probable for reviews that are more information-dense and more variable because reviewers insert low information-dense words in higher information-dense messages which may help to facilitate comprehension (Jaeger, 2010).

Another possibility worth further exploration is whether simpler linguistic factors might account for reviewer ratings more so than relatively sophisticated information measures. Specific lexical words provide a window into the semantic content that might be influencing reader ratings. This is inherently different than Information-Theoretic measures which target how likely those same words are to occur given the frequency of its context. Future studies might include both semantic and Information-Theoretic factors to determine what aspects of one's language use are considered more helpful to its audience.

These exploratory findings add to ongoing research throughout the behavioral sciences. However, further investigation is needed in order to assess the generality and accuracy of our findings. Our analysis explores new avenues for fruitful scientific data exploration. Such is the case with many exploratory studies. Yet, the exploration of interesting behavioral relationships is often inaccessible to classically trained behavioral scientists whose collection methods are often guided, justifiably, by detailed specific theoretical concerns.

Conclusions

Current problems in science and industry involve processing increasingly large and complex data sets which necessitates the development of novel scalable computational tools. Mathematical and computational scientists are naturally a good fit to discover such solutions as their training is often geared toward finding solutions to engineering problems, paying little mind to interesting behavioral phenomena discoverable at their fingertips. Yet, more and more freely available data sets such as those from the Netflix challenge, Yelp Dataset Challenge as well as Twitter's API and Yahoo!'s release of 100 million images, are overwhelmingly loaded with interesting behavioral nuances that can be harnessed to answer longstanding questions in the behavioral sciences and also increase the success of newly developing machine learning algorithms that aim to predict future behavior. Indeed, some cognitive scientists argue we are currently in the midst of a revolution: "to take back behavioral data, and - just as in the last cognitive revolution - to demonstrate the value of postulating a mind between browsing history and mouse movements" (Griffiths, 2015).

Unfortunately, taking back behaviorally relevant data is not always so straightforward. Even after obtaining the data many behavioral scientists do not know what tools are necessary to address their questions. Even when the most cutting-edge tools are freely available, few behavioral scientists are trained in methods that could enable them to harness these powerful tools. To conclude, we return to our three key methodological observations summarizing how they were successfully used to facilitate this process.

First, We described how "sketch" techniques help process large amounts of data efficiently and argue these are critical for the coming 'big data' age in cognitive science (Griffiths, 2015). This provides one instance that shows it is possible that behavioral scientists can often find more efficient data techniques for their problems (Section "R-package `cmscu`"). Second, corpus or other data analysis can make ready use of these specialized solutions (Section "Information-theoretic structure

of yelp reviews"). We demonstrate a fruitful domain in which such a strategy can apply—the Information-Theoretic analysis of language structure in corpora. We then used our library to implement a sophisticated n-gram analysis (iFix-MKN) to explore the statistical properties of language that predict successful communication. Finally, multidisciplinary cross-fertilization is crucial (General Discussion). We revisited important messages about cross-disciplinary interactions and suggest that adopting a wide range of tools for various disciplines can help to ensure that behavioral scientists find efficient solutions for their problems, while giving computational scientists and engineers exciting behavioral problems to apply their research to.

Now, more than ever, interdisciplinary collaborations among behavioral and computational scientists are required in order to successfully accomplish our goals. To this end, some argue we are in the midst of a new era of scientist - the computational social scientist - who's focus lies at the intersection of cognitive and computer science (Lazer et al. 2009). This manuscript belays the success of one such method, interdisciplinary collaboration, which can help maintain the pace of scientific discovery within the behavioral sciences. Our work fits within the broader theme of advancing discovery across the sciences illustrating that interdisciplinary collaboration—which can connect previously intractable problems with new tools and methods—is a successful approach toward accomplishing this goal.

Acknowledgments This work was in part funded by an IBM PhD fellowship awarded to David W. Vinson for the 2015-16 academic year.

Appendix

The two information density measures are the Average Bigram Information (ABI) and the Average Trigram Information (ATI):

$$ABI_j = \frac{1}{N-1}\sum_{i=2}^{N} -\log_2 P_{iFix-MKN}(w_i|w_{i-1}), \quad (5)$$

$$ATI_j = \frac{1}{N-2}\sum_{i=3}^{N} -\log_2 P_{iFix-MKN}(w_i|w_{i-1}, w_{i-2}) \quad (6)$$

where N is the number of words in the j_{th} review, $P_{iFix-MKN}(w)$ is the iFix-MKN probability of word w, and w_i is the ith word. Bigram Information is the probability of w_i given w_{i-1} and Trigram Information is the probability of w_i given the joint probability of w_{i-1} and w_{i-2}. We then obtain the information density of a review by taking the average amount of information across the review. We calculate the standard deviation, σ, of each review's average information by each n-gram and take this as an inverse measure of a message's uniformity: $\sigma(BI)$, and $\sigma(TI)$. An example implementation of iFix-MKN can be found on the website and specific details about the algorithm can be found in Chen and Goodman (1999, section 3).

We estimate both bigram and trigram information, even though estimates within the trigram model rely, in part, on bigram model estimates. We do this because maximum likelihood bigram models will typically account for more unseen data, while trigram models are a better measure of the language's actual structure (e.g., afford better word predictability when the data exists). We avoid making the decision to balance this trade off by modeling both explicitly in two distinct models and comparing those models to one another.

Independent variables We trained the information models on Yelp reviews from the United States only to avoid training on non-English reviews (though future research might look to see if different languages show similar results to those found here). Half of the Yelp, Inc. dataset was used to train each model which was tested on the remaining half. After obtaining information density and uniformity measures for each review within the test set we removed reviews shorter than 15 words in length (see Frank & Jaeger, 2008; Martin & Jurafsky, 2000; Vinson & Dale, 2014, for other possible ways to control for n-gram reliability). This reduced the total number of reviews from 1.1 million to 1.03 million (< 5 % reduction).

We anticipated the possibly of an inflated variance due to multicollinearity when including both information density and uniformity measures within the same model. The variance (inverse uniformity) naturally increases as information density increases, due to the presence of a true zero. For this reason, we took the residual of each uniformity measure, first predicted by its respective information density measure as the true measure of uniformity ($r\sigma(BI)$ and $r\sigma(TI)$). After, a variance inflation factor (`VIF`) analysis (Craney & Surles, 2002; Stine, 1995) in `R` (Library `CAR`) was used to determine whether the new predictor variables exhibit collinearity with any other predictor variable. None of our predictor variables showed signs of strong collinearity (`VIF` < 2). All variables were centered and standardized for the purpose of interpretation.

Assessing model fit U/F/C ratings are count variables associated with each distinct review. Because of this we initially use a Poisson regression model in `R` to predict each rating. We compare this model against a null intercept model using a chi-squared test of difference of log-likelihoods. We found the full Poisson model was an improvement over the null model; however, the variance of each reader rating was greater than the mean (*Useful*: $M = 1.05$, $SD = 2.17$, *Funny*: $M = .46$, $SD = 1.60$, *Cool*: $M = .56$, $SD = 1.72$) indicating possible overdispersion (larger number of zeros).

One assumption within Poisson models is that the variance equals the mean. Violating this assumption may result in underestimated standard errors or inflation in model significance. Because of this, the distribution might be better fit by another model that does not make this assumption (Scott Long, 1997). Specifically, we compare the Poisson regression against a negative binomial regression that does not require the variance equal the mean (thus adjusting for overdispersion).[3] Again using a chi-squared test on the difference between log-likelihoods, we found the negative binomial model was an improvement over the Poisson model for all (U/F/C) ratings. Crucially, the Poisson model results revealed the same exact trend (similar significant predictors) as the results from the negative binomial model, but with higher coefficient estimates. This suggests the negative binomial model is in fact adjusting for inflation of variance present in the Poisson models. Thus, we report the results from the negative binomial models.

To assess the significance of the negative binomial model, we use a likelihood ratio test comparing the deviance of a null model—predicting U/F/C using the intercept only—and the full model. We found the negative binomial model is significantly better at predicting U/F/C than the null model. Importantly, the negative binomial regression model does not allow for a straight forward interpretation of effect size. Taking after previous research, we use an adjusted likelihood-ratio-index (Abney et al., Submitted; Long & Freese, 2006) a type of pseudo R^2_{adj} (Mittlböck et al. 1996) as our measure of effect size:

$$R^2 = 1 - (L_{\text{fitted}}/L_{\text{intercept}})^{2/n} \tag{7}$$

Where L is the data likelihood.

References

Abney, D.H., Gann, T.M., Heutte, S., & Matlock, T. (Submitted). The language of uncertainty and political ideology of news sources in climate communication. *Cognitive Science*.

Aylett, M. (1999). *Stochastic suprasegmentals: relationships between redundancy, prosodic structure and syllabic duration*. San Francisco: Proceedings of ICPhS-99.

Aylett, M., & Turk, A. (2004). The smooth signal redundancy hypothesis: a functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, *47*(1), 31–56.

Aylett, M., & Turk, A. (2006). Language redundancy predicts syllabic duration and the spectral characteristics of vocalic syllable nuclei. *The Journal of the Acoustical Society of America*, *119*(5), 3048–3058.

Bahl, L.R., Jelinek, F., & Mercer, R.L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *2*, 179–190.

Baker, J.K. (1975). The dragon system—an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *23*(1), 24–29.

Bradlow, A.R., Nygaard, L.C., & Pisoni, D.B. (1999). Effects of talker, rate, and amplitude variation on recognition memory for spoken words. *Perception & Psychophysics*, *61*(2), 206–219.

Brennan, S.E., & Williams, M. (1995). The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language*, *34*(3), 383–398.

Chen, S.F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, *13*(4), 359–393.

Clark, H.H., & Brennan, S.E. (1991). Grounding in communication. *Perspectives on Socially Shared Cognition*, *13*(1991), 127–149.

Cormen, T.H. (2009). *Introduction to algorithms*: MIT Press.

Cormode, G., & Muthukrishnan, S. (2005). An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, *55*(1), 58–75.

Cormode, G., & Muthukrishnan, M. (2011). Approximating data with the count-min sketch. *IEEE Software*, *1*, 64–69.

Craney, T.A., & Surles, J.G. (2002). Model-dependent variance inflation factor cutoff values. *Quality Engineering*, *14*(3), 391–403.

Doyle, G., & Frank, M. (2015). Shared common ground influences information density in microblog texts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 1587–1596). Denver: Association for Computational Linguistics.

Eddelbuettel, D., François, R., Allaire, J., Chambers, J., Bates, D., & Ushey, K. (2011). Rcpp: seamless R and C++ integration. *Journal of Statistical Software*, *40*(8), 1–18.

Frank, A., & Jaeger, T.F. (2008). Speaking rationally: uniform information density as an optimal strategy for language production. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, (pp. 933–938).

Genzel, D., & Charniak, E. (2002). Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 199–206).

Goyal, A., Daumé III, H., & Cormode, G. (2012). Sketch algorithms for estimating point queries in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (pp. 1093–1103).

Griffiths, T.L. (2015). Manifesto for a new (computational) cognitive revolution. *Cognition*, *135*, 21–23. (The Changing Face of Cognition).

Heafield, K., Pouzyrevsky, I., Clark, J.H., & Koehn, P. (2013). Scalable modified Kneser–Ney language model estimation. In *Acl (2)*, (pp. 690–696).

Jaeger, T.F. (2010). Redundancy and reduction: speakers manage syntactic information density. *Cognitive Psychology*, *61*(1), 23–62.

Jaeger, T.F. (2013). Production preferences cannot be understood without reference to communication. *Frontiers in Psychology*, *4*.

Jelinek, F. (1976). Speech recognition by statistical methods. *Proceedings of the IEEE*, *64*, 532–556.

Jelinek, F., Mercer, R.L., Bahl, L.R., & Baker, J.K. (1977). Perplexity—a measure of the difficulty of speech recognition

[3] Other possible models include the zero inflated Poisson model and zero inflated negative binomial model. Both models assume that overdispersion is, in part, due to another process that can be modeled independently. One possibility is that certain reviews were simply never read. Though a clearly discernible independent variable, there is no such variable within the dataset that could be used to model this process using a zero inflated-Poisson or negative binomial.

tasks. *The Journal of the Acoustical Society of America*, *62*(S1), S63–S63.
Jurafsky, D., & Martin, J.H. (2000). Speech and language processing. In *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*: Prentice Hall.
Jurafsky, D., Chahuneau, V., Routledge, B.R., & Smith, N.A. (2014). Narrative framing of consumer sentiment in online restaurant reviews. *First Monday*, *19*(4).
Kahn, J.H., Tobin, R.M., Massey, A.E., & Anderson, J.A. (2007). Measuring emotional expression with the linguistic inquiry and word count. *The American Journal of Psychology*, 263–286.
Kirsch, A., & Mitzenmacher, M. (2006). Less hashing, same performance: building a better bloom filter. In *Algorithms–ESA 2006*, (pp. 456–467): Springer.
Kneser, R., & Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing, 1995. icassp-95*, (Vol. 1, pp. 181–184).
Krauss, R.M., & Fussell, S.R. (1990). Mutual knowledge and communicative effectiveness. *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, 111–146.
Kuhl, P.K., Williams, K.A., Lacerda, F., Stevens, K.N., & Lindblom, B. (1992). Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, *255*(5044), 606–608.
Labov, W. (1972a). *Language in the inner city: studies in the black English vernacular* Vol. 3: University of Pennsylvania Press.
Labov, W. (1972b). *Sociolinguistic patterns (No. 4)*: University of Pennsylvania Press.
Lazer, D., Pentland, A.S., Adamic, L., Aral, S., Barabasi, A.L., Brewer, D., et al. (2009). *Life in the network: the coming age of computational social science* (Vol. 323, p. 721). New York: Science.
Levy, R.P., & Jaeger, T.F. (2006). Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems*, (pp. 849–856).
Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., & Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, *17*(2), 149–154.
Lindblom, B. (1990). Explaining phonetic variation: a sketch of the h&h theory. In *Speech Production and Speech Modelling*, (pp. 403–439): Springer.
Long, J.S., & Freese, J. (2006). *Regression models for categorical dependent variables using Stata*: Stata Press.
Lupyan, G., & Dale, R. (2010). Language structure is partly determined by social structure. *PloS ONE*, *5*(1), e8559.
Mahowald, K., Fedorenko, E., Piantadosi, S.T., & Gibson, E. (2013). Info/information theory: speakers choose shorter words in predictive contexts. *Cognition*, *126*(2), 313–318.
Markov, A.A. (1913). Primer statisticheskogo issledovanija nad tekstomevgenija onegina'illjustrirujuschij svjaz'ispytanij v tsep (an example of statistical study on the text ofeugene onegin'illustrating the linking of events to a chain). *Izvestija Imperial Akademii Nauk*.
Martin, J.H., & Jurafsky, D. (2000). *Speech and language processing*: International Edition.
McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D., & Barton, D. (2012). Big data. The management revolution. *Harvard Business Review*, *90*(10), 61–67.
Meyer, D., Hornik, K., & Feinerer, I. (2008). Text mining infrastructure in R. *Journal of Statistical Software*, *25*(5), 1–54.
Mittlböck, M., Schemper, M., et al. (1996). Explained variation for logistic regression. *Statistics in Medicine*, *15*(19), 1987–1997.
Nosek, B.A., Spies, J.R., & Motyl, M. (2012). Scientific utopia II. restructuring incentives and practices to promote truth over publishability. *Perspectives on Psychological Science*, *7*(6), 615–631.
Nygaard, L.C., & Pisoni, D.B. (1998). Talker-specific learning in speech perception. *Perception & Psychophysics*, *60*(3), 355–376.
Nygaard, L.C., & Queen, J.S. (2008). Communicating emotion: linking affective prosody and word meaning. *Journal of Experimental Psychology: Human Perception and Performance*, *34*(4), 1017.
Nygaard, L.C., Sommers, M.S., & Pisoni, D.B. (1994). Speech perception as a talker-contingent process. *Psychological Science*, *5*(1), 42–46.
Pashler, H., & Wagenmakers, E.-J. (2012). Editors' introduction to the special section on replicability in psychological science a crisis of confidence? *Perspectives on Psychological Science*, *7*(6), 528–530.
Pate, J.K., & Goldwater, S. (2015). Talkers account for listener and channel characteristics to communicate efficiently. *Journal of Memory and Language*, *78*, 1–17.
Pennebaker, J.W. (1997). *Opening up: the healing power of expressing emotions*: Guilford Press.
Pennebaker, J.W., Francis, M.E., & Booth, R.J. (2001). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, *71*, 2001.
Ramscar, M., Shaoul, C., Baayen, R.H., & Tbingen, E.K.U. (2015). *Why many priming results don't (and won't) replicate: a quantitative analysis*: Manuscript, University of Tübingen.
Roy, B.C., Frank, M.C., DeCamp, P., Miller, M., & Roy, D. (2015). Predicting the birth of a spoken word. *Proceedings of the National Academy of Sciences*, *112*(41), 12663–12668.
Sagiroglu, S., & Sinanc, D. (2013). Big data: a review. In *2013 International Conference on Collaboration Technologies and Systems (cts)*, (pp. 42–47).
Schmidt, S. (2009). Shall we really do it again? The powerful concept of replication is neglected in the social sciences. *Review of General Psychology*, *13*(2), 90.
Scott Long, J. (1997). Regression models for categorical and limited dependent variables. *Advanced Quantitative Techniques in the Social Sciences*, 7.
Shannon, C.E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, *27*(4), 623–656.
Simmering, J. (2013). How slow is r really? [Blog]. http://www.r-bloggers.com/how-slow-is-r-really/
Song, H., Dharmapurikar, S., Turner, J., & Lockwood, J. (2005). Fast hash table lookup using extended bloom filter: an aid to network processing. *ACM SIGCOMM Computer Communication Review*, *35*(4), 181–192.
Stine, R.A. (1995). Graphical interpretation of variance inflation factors. *The American Statistician*, *49*(1), 53–56.
Tanenhaus, M.K., Spivey-Knowlton, M.J., Eberhard, K.M., & Sedivy, J.C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, *268*(5217), 1632–1634.
Vinson, D.W., & Dale, R. (2014). Valence weakly constrains the information density of messages. In Bello, P., Guarini, M., McShane, M., & Scassellati, B. (Eds.) *Proceedings of the 36th Annual Meeting of the Cognitive Science Society (p 1682–1687)*. Austin, TX.
Vinson, D.W., & Dale, R. (2016). Social structure relates to linguistic information density. In Jones, M. (Ed.) *Big data in cognitive science: from methods to insights*: Taylor & Francis.
Zwaan, R.A., & Pecher, D. (2012). Revisiting mental simulation in language comprehension: six replication attempts. *PLoS ONE*, *7*(12), e51382.