Non-parametric Information Flow Estimation in Social-Media News

Tobin South

February 28, 2021

Thesis submitted for the degree of
Masters of Philosophy
in
Applied Mathematics
at The University of Adelaide
Faculty of Engineering, Computer and Mathematical Sciences
School of Mathematical Sciences

THE UNIVERSITY
of ADELAIDE

Contents

List of Tables

List of Figures

Abstract

News media has long been an ecosystem of creation, copying and critique, where news outlets break stories and add new information and opinions to ongoing stories. Understanding the dynamics of how news information is created and spread is important to accurately ascribe credit to influential work and understand how societal narratives develop. These dynamics can be modelled through a combination of information-theoretic natural language processing and networks, and parametrised using large quantities of textual data.

In this thesis, new comparative techniques are developed to estimate textual information flow between pairs of news-media outlets. To achieve this, Chapter 3 outlines the collection and cleaning of data sourced from the Twitter accounts of news-media organisations collected for the duration of 2019. Chapter 4 introduces two non-parametric entropy estimators and validates their convergence. These estimators are then extended to produce several measures of information flow in Chapter 5, which are compared via simulation models using both synthetic and real text data. The resulting best estimator produces a reliable measure of textual information flow that captures aspects of grammar and word choice in its calculation.

Resulting information flows are constructed into a network of news-media organisations, which Chapter 6 uses to examine approaches of ranking influence using examples from centrality, sport ranking and network topology. The interconnected nature of the information flow ecosystem proves resistant to simplification, demonstrating implicit complexity in the flow dynamics.

In total, this work provides a new methodology for examining the information transmitted between content producers in any connected system of natural language, a toolkit with applications to the many networked discourses of our online world.

Signed statement

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed: ..... Date: .....

Acknowledgements

An immense thanks is owed to Dr Lewis Mitchell and Professor Matthew Roughan for supporting and pushing me to be the best I can be. The coffees, advice and banter made for a fantastic two years.

To my friends, who have brought me regular sleep deprivation and constant laughs for many years, thanks.

Finally, thanks to my parents; you may not understand this thesis, but your pride and faith in me for the last 22 years has been limitless.

Dedication

To my parents, for helping me become the best person I could be; and to Thomas and Nicole, for all the late night O'Connell St bakery visits and the long drives to loud music.

List of papers arising from this thesis

- "Complex contagion features without social reinforcement in a model of social information flow" [66], published in *Entropy*, used the code developed in Chapter 4 to efficiently estimate cross entropy rates as a measure of information flow in an contagion model.
- "Popularity and centrality in Spotify networks: Critical transitions in eigenvector centrality" [80], published in *Complex Networks*, expands on the discussion in Chapter 6 for a collaboration network of musical artists from Spotifty.

Chapter 1

Introduction

1.1 Motivation

Journalism has always been a cornerstone of democracy. It both shapes and is shaped by public sentiment, influencing the social sphere through its choices of content and its framing of issues. This 'fourth estate' has turned the tides of history, with countless examples from the role of newspapers in distributing information during the American Revolution [83] to the exposé of the Panama Papers [62]. This influence is a perilous task; during the late 1990s the news-media played an important role in rapidly proliferating the misinformation linking the MMR vaccine to autism – but also played a key role in exposing the fraudulent science afterwards [31]. News-media can act as an powerful tool in boosting morale during wartime, and can equally serve as a machine of propaganda [45]. News-media must fight many challenges to maintain credibility, profitability and journalistic integrity.

The rise of the internet and online social media sites has increased some of these challenges. New mediums for online engagement have induced a rapid change in the ways citizens consume content. Roles the media has historically played have become decentralised, with social movements such as the Arab Spring and Occupy Wall Street arising in large part through collective action of citizens online [77]. This decentralised collective system of news is a powerful connector, but also prone to increasing polarisation [7]. Despite these changes, traditional news-media has adapted to, and in many cases thrived in, this new environment. Many popular news-media organisations are primary funded through digital advertisements and digital-only subscriptions, including the New York Times where digital revenue overtook print for the first time in 2020 [82]. Traditional news-media continues to produce a significant amount of content, which it typically attempts to disseminate

over social media and through online platforms as well as offline.

This huge volume of content, both traditionally produced and user generated, makes analysis difficult. Qualitative approaches can study impactful events and articles, but cannot encompass the deluge of content that played smaller roles in a story's development. Conversely, quantitative tools are often capable of spanning the totality of relevant content, but often focus on single aspects of a news story, such as counts of words or sets of words over time [55, 35, 64]. These quantitative approaches are important lenses through which to view news, but need to be combined with context. Where context and quantitative analysis of news are combined, novel perspectives can be formed; exemplifying this is the usefulness of hyperlinks in revealing the cross-linguistic communication between bloggers during the Haitian earthquake in 2010 [38].

In particular the study of *information diffusion* has benefited from the internet and online social media, with studies into topics such as information propagation in the blogspace [36] and the spread of true and false news through social media [87]. The study of information diffusion has a longer history using other contexts such as the spread of messages in organizations [91], and the information flow within research and development laboratories [2]. These studies of diffusion are often accompanied by models of flow, usually in the form of epidemiological or statistical physics models [14]. These models and empirical studies share a common theme; they focus on the transmission and diffusion of singular ideas or packets of information.

The question we study here is of information flow more generally. Rather than constraining the analysis to individual stories or discrete packets of information, we use tools from information theory that capture the complexity of language in this flow. Similar approaches have used information-theoretic tools on the temporal data from social media [84] and more recent work has used non-parametric entropy estimators to study information flow in a Twitter user's local social network [6]. These approaches have subsequently been extended into a model of information flow built from these simple estimators [5, 66].

This thesis significantly extends these approaches by speeding up and validating the convergence of these non-parametric entropy estimators on text, extending the estimators into several measures which better estimate and represent information flow, and applying these tools to the novel context of news-media data. These new additions to the literature contribute to the study of information flow in news, but also provide significant improvements to the toolkit of information flow analysis in any systems containing connected producers of textual information.

1.1.1 Preliminary background

The rise of modern news is intricately linked to the rapid changes in our tools of information technology. These technologies are useful not only in providing information to the public, but also in analysing it [71, 20]. In contrast to traditional studies of news, techniques built from fundamental engineering tools can provide an investigation into news independently of the broader qualitative context, using only the data.

This notion of information is drawn from the field of information theory established in the first half of the 20th century [71, 61, 39]. Information in this context takes human interpretable content and quantifies it into useful computational objects. This approach allows for an analysis of these objects through measuring uncertainty, predictability and transmission. These quantities, although summary statistics of the full informational context, can be done at speeds orders of magnitude faster than traditional human comprehension, allowing for large quantities of information to be analysed quickly.

In this work, these large quantities of information will come in the form of human readable text. To convert human-style language into discrete quantified information we draw upon several techniques from natural language processing. These techniques allow us to both convert natural language text into sequences of consistent 'tokens', but also to generate synthetic text that allows for model evaluation using simulations.

Combining these two tools allows us to create networks – mathematical constructions where objects called 'nodes' are connected by directed relationships called 'edges' [58]. These constructed networks allow us to begin examining the system as a whole using analysis tools on networks. The tools will include a variety of ranking methods drawn from centrality measures, sports ranking, and graph-topological arrangement.

1.1.2 Outline of thesis

This thesis is separated into five main chapters. We begin Chapter 2 by providing a background of the three fields mentioned above: information theory, networks, and natural language processing. We introduce several important concepts, such as the notion of entropy rate, cross entropy rate and maximal predictability.

In Chapter 3 we introduce our data and describe how we collected and cleaned it. This data, representing a year's worth of tweets from 154 news-media organisations, is representative across the US political spectrum and contains a total of 2,977,980 tweets. We included an exploratory data analysis of the corpus.

In Chapter 4 we develop a new modified **cross entropy rate estimator** by extending the work of Kontoyianni *et al.* [48] and Bagrow *et al.* [6]. The convergence of this estimator is studied using both simulated and real text to confirm it as a useful tool for finding time-based entropy relationships between text sequences. This estimator is optimized and released in an open source Python package listed in Appendix C.

In Chapter 5 we evaluate the use of the cross entropy rate estimator as a measure of information flow and discover it is incomplete. We hence extend the estimator into a collection of possible flow measures that combine both directions of cross entropy with self-entropy rates and local network information to formulate a set of measures to estimate information flow. These measures are evaluated using simulated models of networked information flow which encapsulate the complexity of natural language combined with chains and cycles of artificial quoting. Measures that normalised differences in cross entropy flow by target information content prove most effective, with a single measure based on local neighbour information consistently outperforming all other measures.

Finally, the highest performing measure is then applied the news data previously collected and cleaned to construct a network of information flows in news. This network is analysed in Chapter 6 to examine the feasibility of a definitive ranking. Several ranking methods are used including network centralities, a sports ranking and a graph topological approach. The network of flows is shown to have centralities and ranks that are sensitive to small changes in the network, and the diverse rankings are highly discordant. This diversity of results highlights key difficulties of simplifying such a network.

We conclude with a discussion and outlook for further work in Chapter 7.

Chapter 2

Background

This chapter will introduce core concepts from the three main fields on which this thesis draws: information theory, graph theory, and natural language processing. Throughout this thesis these three areas will be interwoven to produce new methods and results. When more specific tools from these fields are used in later chapters, they will be introduced during their discussion and this chapter referenced accordingly.

2.1 Information theory

Information theory is concerns the quantitative analysis of storage and transmission of information. This field underpins much of modern information and communication technology, with applications in signal processing, data compression [70, 9], statistical inference [13], linguistic analysis [17, 34], cryptography [54] and even human perception [22]. An import concept at the heart of information theory and these applications is *entropy*.

2.1.1 Entropy

Entropy is a measure of the uncertainty of a random variable. This version of entropy, commonly defined by Equation 2.1, is referred to as Shannon entropy, named after Claude Shannon for his work in 1948 studying the quantities of information in transmitted messages [71]. The definitions hereafter are sourced from "The Elements of Information Theory" by Thomas and Cover [20].

Definition 2.1.1 (Shannon Entropy)**.** Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = P(X = x), x \in \mathcal{X}$.

The entropy $H(X)$ of X, is

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x). \tag{2.1}$$

The entropy of the random variable is measured in bits. A bit can have two states, 0 or 1. The entropy of a random variable is the number of bits on average that are required to describe the random variable in question. To measure the entropy in bits, we use a logarithm of base 2, and all logarithms throughout this work are assumed to be in based 2, unless otherwise specified.

To give a typical example of entropy, when an unbiased coin is tossed there are two equally probable outcomes, giving an entropy of 1 bit.

Remark. We use the convention of $0 \log 0 = 0$, which sensibly means that adding a state with 0 probably to the random variable does not change its entropy. This adds several useful properties including that the entropy is always non-negative.

Lemma 2.1.1. The entropy of a random variable is strictly non-negative, $H(X) \geq 0$.

Proof. $0 \leq p(x) \leq 1$ which implies that $\log \frac{1}{p(x)} \geq 0$, hence the sum of products of strictly non-negative terms will always be non-negative. ■

Remark. The entropy of the random variable X can also be described in terms of the *expected surprise*, where the surprise of a state is $\log \frac{1}{p(x)}$.

$$H(X) = \mathbb{E}\left[\log \frac{1}{p(x)}\right]. \tag{2.2}$$

An important concept we draw on throughout this work is the broader notion of *complexity*. When the entropy of a system is higher, it is often said to be *more complex*. While complexity itself has many varied definitions in science, we map our quantitative calculation of entropy onto this complexity, helping us discuss the impact of entropy on real systems. In essence, a more complex system is one in which one expects to be surprised more often.

2.1.2 Conditional entropy and cross entropy

The definition above for a single random variable is extended by introducing a second discrete random variable Y. This opens us to a variety of information theoretic quantities, only a few of which we will examine here.

We first introduce the concept of joint entropy. Joint entropy, although not directly used in the rest of this work, is a core concept when extending

from a single random variable to two random variables. In essence, this joint entropy is measuring the complexity of a two dimensional probability space.

Definition 2.1.2 (Joint Entropy)**.** The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ and state spaces $(\mathcal{X}, \mathcal{Y})$ is

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y). \tag{2.3}$$

This definition of joint entropy is then useful in describing conditional entropy.

Definition 2.1.3 (Conditional Entropy)**.** The conditional entropy $H(X|Y)$ of random variable X given random variable Y is defined as,

$$\begin{aligned} H(X|Y) &= \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y) && (2.4) \\ &= -\sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log p(x|y) \\ &= -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log p(x|y) \\ &= -E \log p(X|Y). && (2.5) \end{aligned}$$

The conditional entropy can also be defined using the joint entropy by

$$H(X|Y) = H(X, Y) - H(Y). \tag{2.6}$$

Indeed, this hints at an interpretation of conditional entropy. If the complexity of X can be mostly explained by conditioning on Y, then the conditional entropy is very low. In a sense, this conditional entropy is hinting at a directed information relationship between the variables X and Y.

Subtly different from the *conditional entropy* is the *cross entropy*. Whereas the *conditional entropy* is the amount of information needed to describe X given knowledge of Y, the *cross entropy* is the amount of information needed to describe X given a optimal coding scheme built from Y. An additional subtle complication of language here is that in conditional entropy we normally refer to X and Y as random variables whereas cross entropy usually is directly describing relationships between *distributions*. As such, we shift our notation to the distribution of $X \sim q(x)$ and $Y \sim p(y)$.

Definition 2.1.4 (Cross Entropy)**.** The cross entropy $H(p||q)$ between two probability distributions, p and q, defined over the same state space, is defined as,

$$H(p||q) = -\sum_{x} p(x) \log q(x). \tag{2.7}$$

Remark. Importantly, note that $H(X|Y) \neq H(Y|X)$ and $H(p||q) \neq H(q||p)$, properties we will exploit later.

The conditional entropy is a useful information-theoretic measure, but can often be difficult to compute without knowledge of the conditional probability distribution. As such, the cross entropy provides a more practical tool for us to use in our analysis.

Although cross entropy has the common notation $H(p, q)$ elsewhere in the literature, in this thesis we will use an alternative $H(p||q)$, reminiscent if the Kullback–Leibler divergence, so as to not confuse the cross entropy with the joint entropy and to provide an important notion of the direction of information flow from q to p.

Throughout this work, entropy will be referred to in two ways. If the entropy measure in question is theoretical in nature we will use the symbols H, $H(X)$, or $H(X||Y)$. Where the entropy measures are estimates derived from data, we will prefer the notation $\hat{h}$, $\hat{h}(X)$, and $\hat{h}(X||Y)$.

So far, these entropy measures have looked at discrete random variables. There are several ways this can be extended, but as we work with sequences of random variables later, we look towards random *processes* and their analysis using entropy rates.

2.1.3 Entropy rates

Rather than examining the entropy of a single random variable, we can turn our inquiry toward a *sequence* of random variables. These random variables can be generated by some *stochastic process*, which at each point in the sequence draws a realisation from a probability distribution than can depend on the previous realisations of the sequence. Consider a simple random walk where the probability distribution of the next location is centred tightly around the previous location. The future of this sequence can be said to be dependent on its past, in this case by only one time step.

In such a self-dependent stochastic process, tools such as Shannon entropy can be ineffective at fully describing the complexity. Consider two stochastic processes drawing from the set $0, 1$: A which always alternates, and B which selects values randomly.

0101010101010101... 0001011011011100...
Sequence A Sequence B

With knowledge of the process, we can predict perfectly that the next element of sequence A is 0; whereas sequence B cannot be predicted better than a coin toss. However, describing both sequences using Shannon entropy would result the same 1 bit per symbol.

The *entropy rate* provides a slightly different lens for processes. The entropy rate of a stochastic process describes the amount of information required to describe the future state of a process, conditioned on the information in the history of the process.

Definition 2.1.5 (Entropy Rate)**.** Let $\mathcal{X} = \{X_i\}$ be a ergodic stochastic process with a finite alphabet. The entropy rate can be defined as,

$$H(\mathcal{X}) = \lim_{n\to\infty} H\left(X_n | X_{n-1}, X_{n-2}, \ldots, X_1\right). \tag{2.8}$$

Given the assumption of stationarity, this can be expressed as,

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H\left(X_1, X_2, \ldots, X_n\right). \tag{2.9}$$

This entropy rate is closely connected with the ideas of compression. As noted by Kontoyiannis *et al.* [48], 'the entropy rate is almost surely an asymptotic lower bound on the per-symbol description length when the process is losslessly encoded'. Algorithms such as Lempel-Ziv [96] exploit these lessons to efficiently compress sequences as close to their size lower bound as practical. However, these algorithms are not a robust method of measuring the entropy rate of a sequence because implementations can differ across systems. Indeed, while this entropy is a useful theoretical tool, calculating it for real examples can prove difficult. We discuss methods of estimating entropy rates on large quantities of data in Chapter 4.

In order to assist in these later efforts, we will occasionally exploit a useful simplification of the entropy rate. If the sequence generated by a stochastic process is made from independent and identically distributed (i.i.d.) random variables then the entropy rate of the process is simply the Shannon entropy of the random variables.

Lemma 2.1.2. The entropy rate of an i.i.d. stochastic process, $\mathcal{X} = \{X_i\}$, is the Shannon entropy of each individual member of the process, $H(X_i)$.

Proof.

$$H(\mathcal{X}) = \lim_{n\to\infty} H\left(X_n | X_{n-1}, X_{n-2}, \ldots, X_1\right), \tag{2.10}$$

and by conditional independence,

$$\begin{aligned} H(\mathcal{X}) &= \lim_{n\to\infty} H\left(X_n\right) \\ &= H(X_0). \end{aligned} \tag{2.11}$$

■

2.1.4 Predictability

Using these entropic measures we can create other quantities as well. One such useful quantity is the maximal predictability.

Predictability is the probability π that an ideal theoretical predictive algorithm could correctly predict the next state of a process. For a process with $\pi = 0.3$, we could expect to predict this process correctly 30% of the time.

This π is often difficult to obtain, however an upper bound, $\pi \leq \pi^{max}$, is possible through the use of Fano's inequality [27]. For a process with $\pi^{max} = 0.3$, we could expect to predict this process correctly *no more than* 30% of the time, no matter how good our predictive algorithm [78].

Definition 2.1.6 (Maximal Predictability)**.** For a process $X = \{X_i\}$ with entropy $H(X)$ and state space $\mathcal{X}$, Fano's inequality in the context of our maximal predictability gives,

$$H(X) = H(\pi^{max}) + (1 - \pi^{max}) \log(|\mathcal{X}| - 1) \tag{2.12}$$

The entropy of the maximal predictability $H(\pi^{max})$ is substituted with the binary entropy function [78],

$$H(\pi^{max}) = -\pi^{max} \log(\pi^{max}) - (1 - \pi^{max}) \log(1 - \pi^{max}). \tag{2.13}$$

Which finally gives us a form that can be solved numerically for the fundamental limit of the process predictability, π^{max},

$$-H(X) = \pi^{max} \log(\pi^{max}) + (1-\pi^{max}) \log(1-\pi^{max}) - (1-\pi^{max}) \log(|\mathcal{X}|-1). \tag{2.14}$$

Throughout this thesis, maximal predictabilities will found by solving Equation 2.14 using the Powell's conjugate direction method, implemented in Python using SciPy [85], with a starting estimate for the root at $\pi^{max} = 0.5$.

These maximal predictabilities will be used throughout this work and will be extended to a notion of maximal cross predictability in Subsection 2.1.3.

2.2 Networks

The second major area of mathematics used in this work is the theory of networks. Networks are powerful tools for encapsulating the relationships between a collection of objects [58]. These objects of interest are referred as *nodes*, where each node represents a distinct object. The relationships between these nodes are called *edges*.

Both nodes and edges can have properties. We differentiate here between a *graph*, which is exclusively a collection of nodes and edges without extra properties, and a *network*, which is a graph where nodes and edges can have properties assigned to them such that they can represent real objects. For example, a node could be a stand-in for a random variable, bundle of data or a news-media organisation. This node could have a variety of properties such as a name or Shannon entropy.

Similarly the edges can take several forms. Two nodes can have a symmetrical relationship in which they have an *undirected* edge or the relationship could be *directed* from one node to the other. Another important property of an edge that we use here is a *weight* which can describe the magnitude of the relations between the nodes, such as the amount of flow from one node to the other.

Definition 2.2.1. A network is a collection of nodes that are connected by edges. Nodes can contain meta-information and properties such as a name. Edges can have an optional direction and weight.

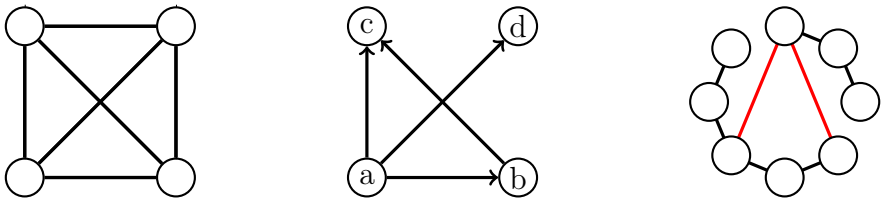
Networks throughout this work will either be generated by using the relationships extracted from data, or relationships will be generated through a random graph model for experimentation. A random graph model is a procedure for creating a graph by following a specific algorithm, sometimes using a set of chosen parameters. Here we outline three random graph models that will be useful in this work.

- Although not random in its simplest form, we start with a clique graph [58]. A clique graph is a graph where every pair of nodes has a single edge between them. When these edges are undirected, there is no randomness in this graph. However, throughout this work we will modify this clique graph such that each edge has a random direction and weight assigned to it.

- One of the simplest and most widely used random network models is the Erdős–Rényi (ER) random graph model [26, 30]. This ER(n, p) model creates a graph with n nodes and links each pair of nodes with

probability p. These graphs have an expected $n(n-1)p/2$ number of edges. Again these can take both directed and undirected forms, of which we mainly use the latter by randomly choosing a direction for each edge at its creation.

- Finally we introduce the more complicated Watts-Strogatz 'small world' model [88]. The version of the model we use imagines a set of n nodes in a ring. In this ring each node has an edge connecting it to the nearest k neighbours. Each of these edges is then rewired with probability β, meaning that one end of the edge is relocated to connect to a different node at random. At $\beta = 0$ there is no rewiring and at $\beta = 1$ the graph is approximately an ER(n, p) graph where $p = k/(n-1)$. This β parameter has several interesting properties; as β increases, the average path length (the average number of hops required to get from one node to any another) decreases rapidly, the clustering coefficient decreases slowly, and the degree distribution becomes more skewed [88]. As such we can change the structure of the network, without altering the total number of nodes or edges, a property we will exploit for our analysis in Chapter 5.



(a) (b) (c)

Figure 2.1: Three examples of graphs and networks. (a) is an undirected clique graph with four nodes. (b) is a directed Erdős–Rényi network with four labelled nodes and $p = 2/3$. (c) is a undirected Watts-Strogatz graph with $n = 8, k = 2$ and $\beta = 0.25$, where rewired edges are coloured red.

While the literature is full of many diverse and useful random graph models, we choose these three models for their simplicity and ability to control specific factors. In particular, the three graphs allow us to individually control the network size, edge probability and rewiring complexity independently, while holding other parameters constant. This allows us to examine the relationship between graph properties in several settings and their flow dynamics, which we study in Chapter 5.

Some example graphs and networks are shown in Figure 2.1. By combining these tools with information theory, we can construct networks from

data and simulate information flows through networks, but a final tool-kit is needed to process the raw data we are interested in.

2.3 Natural language processing

While information theory and networks will form a backbone of our analysis, the data we are interested in comes in a difficult form. Natural language is the non-rigid, semi-unstructured style of language used in everyday conversation and writing. This natural language is the type of text that makes up the social media news of interest to us. In order to study this and convert it to useful forms for quantitative analysis we need an additional tool-kit.

Natural language processing (NLP) is the area of study in which unstructured human language is examined via machine. Formally, natural language refers to either spoken or written language, designed to be understandable to a human listener or reader. This language is not explicitly designed to be machine understandable, and machine comprehension of this language is a challenging problem [4].

NLP is a broad term covering many models and techniques for computationally processing meaningful information from text, ranging from the simple identification of individual words, to the extraction of deeper semantic meaning.

Early work in NLP focused around simple grammatical rules and small vocabularies, such as the work of Georgetown-IBM [43] to translate 60 sentences from Russian to English in 1954. With the rapid increase in computational power and the size of digital text corpuses, modern NLP has focused on deeper challenges of extracting meaning from text with tools such as Word2Vec [57, 56] or deep learning methods such as Google's BERT [23].

These methods face a daunting challenge: language is not only complex and often duplicitous, but contextual and ever-changing. In this work, we present a new comparative method of natural language processing, which attempts to implicitly build this context into the network it creates.

2.3.1 Tokenization

The first step of any natural language processing is tokenization. Tokenization is the fundamental building block of NLP and can often prove deceptively hard. In simple terms, tokenizing a piece of text is the process of breaking a long sequence of characters into smaller chucks of characters called tokens. This often means breaking a sentence into words, e.g. `'This thesis is great'` becomes `['This', 'thesis', 'is', 'great']`.

The task becomes more complex when we introduce contractions. In the case of the word, `‘‘that’s’’`, should we introduce a new token to represent the compound, or break it into two new tokens, `[‘that’, ‘is’]`? There are many such choices that can be made during tokenization and several will be discussed when they are applied in Chapter 3.

While each choice may seem trivial, a process of matching segments of text is highly dependent on tokenization choices such as these, which will have carry on effects to the rest of this work. While there are many aspects of tokenisation that could be elaborated upon here, we focus on one very simple and important concept, the vocabulary size, as it will be critically important throughout this thesis.

Vocabulary size

The vocabulary size is simply the number of unique tokens in any corpus. In general, the vocabulary size of a corpus grows sub-linearly with the total number of tokens [40]. This is deeply important to our work as an increased vocabulary size can often result in a larger complexity. Given the differences in data collection and content production, we sometime have need to observe the complexity of the language independent of the vocabulary size or token count. As such, vocabulary is a fundamental properly which we referenced throughout this work.

2.3.2 Text generation

Text generation, sometimes referred to as natural-language generation, is an algorithmic process of generating natural-looking text. Text generation has many applications from chat-bots [52] and article writing bots to text generation for modelling and analysis purposes [44].

The techniques range from dictionary approaches, such as those first used by ELIZA in 1964 [89], to language models which can range from simple probabilist generators [69] to large machine learned models trained on huge quantities of text [65]. Each generation approach has trade-offs, especially with regards to explainability. Large complex models based on deep learning can prove extremely effective and powerful, but post risks in scientific settings when models are stochastically trained “black boxes” with results that are not reproducible.

In scientific settings such as this work, it is often best to apply simple models, even at the expense of realism. One such model we use throughout this work is a simple independent and identically distributed (i.i.d.) generation approach. This approach repeatedly draws from a bag of tokens with

replacement according to a chosen distribution.

Zipf's law

One such example is drawing from a Zipf distribution [93, 94]. This distribution is based on results from quantitative linguistics showing that in many corpora of natural language the frequency of words is inversely proportional to a power of their rank in the list of most frequent words. This is closely related with a power-law distribution, and can be stated mathematically as the probability of a word with rank k occurring next being,

$$P(k) = \frac{1}{k^{\alpha}} \frac{1}{H_{N,\alpha}}, \tag{2.15}$$

where N is the total vocabulary size, α is scaling parameter characterizing the distribution and $H_{N,s}$ is the Nth generalised harmonic number which acts as a normalising constant.

The choice of α in such a distribution is usually derived from fitted data. As various results for α exist between different corpses (usually ranging from between 0.5 to 2), we will fit the scaling parameter on our own data in Subsection 3.1.2.

Alternative approaches

While many other approaches for synthetic text generation exist we restrain ourselves to these i.i.d. models listed here.

More complex models can produce very realistic text, for instance through deep learning techniques, however it is impossible analytically compute entropy rates for these models. Conversely, simpler models such as Markov chain text generation do have analytic entropy rates, but require more assumptions to be made about the text process in creating the transition probabilities.

We find here that the Zipf i.i.d. model sufficiently balances realism for our context while requiring only a single parameter to be fit. Indeed, when we use this text generation we will also apply the same techniques to real text data in Chapter 4 and Chapter 5 and find similar results.

In this chapter, we have introduced three mathematical fields separately. However, moving forward – and especially in Chapter 5 – we will be combining these areas to produce hybrid tools (*e.g.* natural language generation on networks to produce information flows). First, these tools require data, a topic we now turn out attention to in Chapter 3

Chapter 3

Data collection and cleaning

"Without news to feed it, the biggest story starves."

Emlyn Williams, 1968

In order to analyse information flow in news, we first need a dataset that is: comprehensive, containing most popular news sources; textual, as methods developed in this work are based on text data; and relevant, as the analysis of news should happen where people actually consume it.

In the 1950s, this source would be household television, the widespread popularity of which allowed TV broadcasting to become the primary tool for influencing public opinion in developed nations [1]. The rise of the internet and social media sites in the last two decades has reshaped this paradigm. Moving beyond regularly scheduled television news and structured newspapers, the conveniences of the modern developed world allow individuals to consume news any time, anywhere. As of 2019, 55% of US adults get their news from social media either 'often' or 'sometimes' and 88% state that 'social media companies have at least some control over the mix of news people see' [73].

Given the importance of a free press and the role of social media sites in the delivery of news, this work aims to study the news on social media. To begin this task, we first define some common terms for clarity.

Social media: The platforms used to consume and share information by individuals in the public. E.g. Twitter, Facebook, Reddit.

News-media: The organisations that are producing information about a broad range of current events and sharing that information with the public.

News: The *content* produced by news-media organisations.

Consumers: Individuals in the public that willingly seek out and consume news from news-media organisations.

3.1 Data

Using the media analysis source AllSides[1], a collection of news-media organisations was found. The purpose of AllSides is to provide an open analysis of political leanings of news sources [29], and to aggregate news allowing consumers to view articles from different sides of the political spectrum. Each news source is labelled into one of 5 categories; Left, Lean Left, Center, Lean Right, or Right. For each news source the ratings are determined internally using 'blind surveys of people across the political spectrum, multi-partisan analysis, editorial reviews, third party data, and tens of thousands of user feedback ratings' [29]. News sources are only assigned to a single category, but do have an attached confidence rating that is provided from users selecting if they agree or disagree with the rating. An example of the ratings can be seen in Figure 3.1.

From the website, a list of possible news sources was collected on February 1st, 2019. In this collection was an organisation name, political bias, the number of user feedback ratings of the political bias, and, if available, the Twitter handles associated with the organisation. These collected news sources were broad, containing not just news-media organisations but authors, pundits and think tanks.

We performed the following filtering: a source was only considered if it was a organisation (not an individual), that produced news content of a diverse range of topics. Many news sources were connected to think tanks or opinion groups, and only created news of a single topic or campaign. Further, if a news-media organisation has no Twitter account or had less than 10,000 followers, then it was removed from the pool. This mainly removed inactive organisations and news organisation from very small rural towns.

Finally, a single source was removed as it was not in English (*@univision*), and a single source (*@theMRC*) was removed as its content was a subset of its sister site, CNSNews. The result of this filtering process is 170 news-media organisations with associated Twitter accounts and categorised political biases. A list of all news-media organisations under analysis can be seen in Appendix A and all removed sources and the removal justification can be seen in Table A.2.

Collection

Using the Twitter user handles associated with each news-media organisation, the history of all tweets for each account was collected using the Twitter

[1] www.allsides.com

Figure 3.1: An example collection of News-Media sites that have been classified into biases; sourced from Allsides website [29].

application programming interface (API)[2] and web-scraping tools[3]. Of interest in this work are the tweets each news organisation made between January 1st, 2019 to January 1st, 2020, which is the largest practical time-frame possible for analysis during this research project given data collect constraints.

Over this period major news-media organisations tweeted pieces of news multiple times throughout the day. The manner in which each organisation does this can differ and no standard format is used. The tweets often come in the form of a single line description of articles, alongside a link to the article on the news-media organisation's website. The primary purpose of using social media sites to post these stories is to drive traffic to the organisation's website, wherein they can earn revenue from ad impressions. As such, the format of such news tweets is to extract core concepts from articles and frame them in their most essential and appealing way; often they are trying to create so-called 'click-bait' [67] but even standard reporting is often reformatted to a clear tweet length summary. This format is desirable for our work as we want to explore how the language news-media organisations use to appeal to consumers differs between organisations.

Twitter also serves as a tool for breaking news. The modern 24 hour news cycle has had many effects on journalism, including a pressure to produce breaking news at a lightning fast pace. The use of social media as a near instant tool for global public communication means that often no time is wasted in publishing a story, potentially while it is unfolding. Indeed, research has explored Twitter's role in for breaking news in the cases of the 2011 UK summer riots [86], through providing real time updates over the four days, and in the case of the death of Osama Bin Laden in 2011 [42], where the news was leaked and spread virally through social media before any news-media organisations could fully verify and publish stories on the claim. This rapid information exchange can prove dangerous, as in the case of the 2013 Boston Marathon bombing [81], in which that 29% of the most viral content on Twitter was rumors and fake content [37].

Account removal

Using this collection method a total of 3,221,769 tweets were collected from the 170 news-media organisation official Twitter accounts in the 2019 calendar year. This represents an average number of tweets per day of above 50 tweets for each news organisation. However, the activity level and consistency of output variety greatly between organisations. Figure 3.2 shows the

[2] https://developer.Twitter.com/en/docs
[3] https://github.com/twintproject/twint

distribution of average daily number of tweets for each news-media organisations, showing that some news organisations produce very little content on average. This can be explained through two mechanisms.

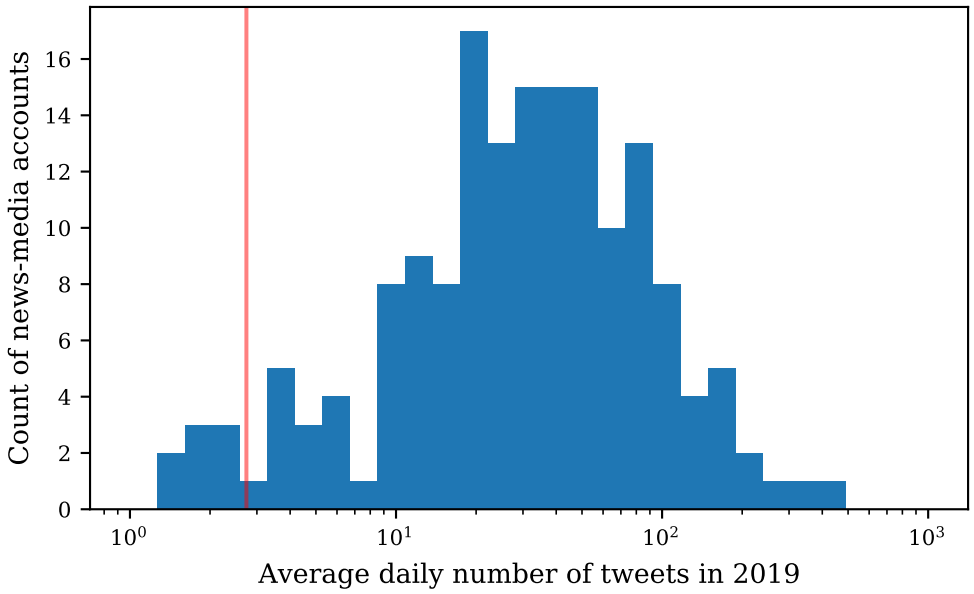


Figure 3.2: The average number of tweets produced each day during the 2019 calendar year for all 170 news-media organisations. The red line is the chosen threshold of 1000 tweets in the year, an average of 2.74 tweets per day.

Firstly, some organisations are not very active on social media. In particular, smaller organisations, which are typically less well resourced, place a lower priority on social media posting. This lower tweet volume presents a challenge for this work. In particular, the use of the non-parametric entropy estimator in Chapter 4 require a substantial amount of text to converge. As such, organisations that produced less than 1000 tweets in 2019 were removed from further consideration. This removed a total of 11 news-media organisations, listed in Table 3.1.

Secondly, five news-media organisations, for reasons unknown, had large periods of time in which they did not post any tweets. These periods of time spanning a few months present key challenges to our investigation. Time is an important aspect of news, especially in the context of news, and this aspect is incorporated into our entropy estimation tools in Chapter 4. As such, news organisations that take long breaks will not have fair information theoretic comparisons. These five organisations, listed in Table 3.2 were removed from further analysis.

News-media Organisation	Bias	Number of tweets in 2019
RealClearPolitics	Center	532
IJR	Lean Right	777
WND News	Right	709
PRI	Center	346
EurekAlert!	Center	610
FAIR	Center	697
Crowdpac	Center	521
Inside Philanthropy	Center	781
Diplomatic Courier	Center	750
Peacock Panache	Left	198
Independent Voter	Center	303

Table 3.1: Table of news-media organisations that were removed from data due to a low number of tweets in the 2019 calendar year.

News-media Organisation	Bias
American Thinker	Right
Pacific Standard	Lean Left
Philly.com	Lean Left
Splinter	Left
ThinkProgress	Left

Table 3.2: Table of news-media organisations that were removed from data due to long periods of inactivity.

We examined the daily activity of each news-media organisation. An activity curve for the *New York Times* can be seen in Figure 3.3. A clear weekly trend, wherein tweet activity is decreased, but not zero, during the weekends is seen for most news-media organisations. Further, many news-media organisations have distinct spikes at key points during the year. These spikes indicate an organisation is covering a rapidly evolving breaking news story, or responding to major changes in discourse through the day. These are interesting and important features in the data; the full collection of figures containing the daily activity levels of all included organisations can be seen in Appendix B.

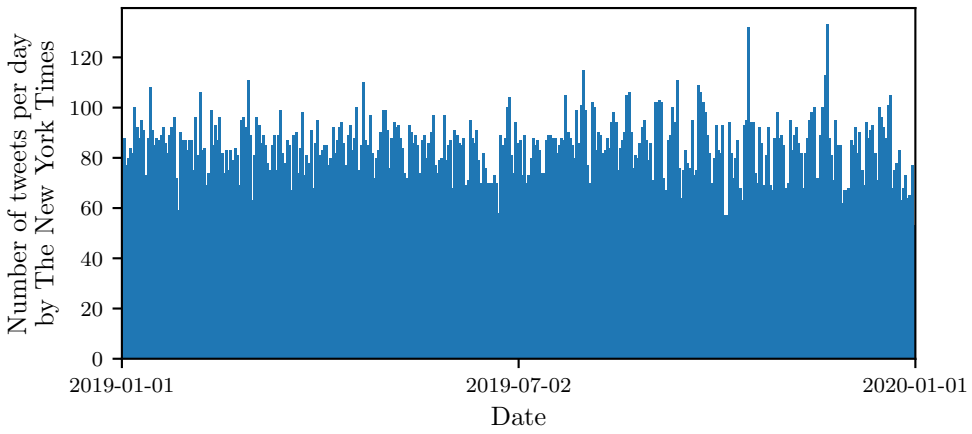


Figure 3.3: Twitter activity over 2019 for 'The New York Times'. Twitter handle is 'nytimes' with 44800317 followers and 31029 total tweets in 2019. This is only one news-media organisation and all other activity figures for other organisations are available in Appendix B.

Account analysis

After filtering we are left with 154 news-media organisations with complete data for the 2019 calendar year. These organisations average 52.97 tweets per day, with a total of 2,977,980 tweets. There are 73 organisations in the left half of the bias spectrum, 44 in the centre and 37 on the right (Table 3.3). This distribution, although shifted towards the left, still provides sufficient samples of bias to explore further.

The news-media Twitter accounts also provide metadata about each organisation. Two useful pieces of metadata are the geographic location, and the number of followers on Twitter.

Bias	Number of Organisations
Left	31
Lean Left	42
Center	44
Lean Right	16
Right	21

Table 3.3: The number of news-media organisations in each political bias classification in our data.

The Twitter account of each news-media organisation can elect to provide a free text 'location'. In some cases this option can be used for other purposes, such for self promotion (e.g. the *New York Daily* states its location as '`New York City / fb.com/nydailynews`') and many organisations elected to leave the field blank. Of the organisations with text, locations can be difficult to disambiguate and compare. We therefore classified locations manually in Table 3.4. Where multiple cities or locations are given, the largest possible inclusion was taken. For example '`New York and the World`' would become 'Worldwide' in our classification, as would '`NYC, London, Paris, Hong Kong`'. There is a notable U.S focus to the data, as is expected using a U.S. based bias rating tool.

Location	Counts
New York	34
Washington, D.C.	20
Cafilforna	11
Other US City	44
General US	8
Worldwide	6
United Kingdom	5
Qatar	1
Pakistan	1
Korea	1
Unspecified or Unclear	43

Table 3.4: The aggregated self-defined locations of news-media organisations according to their Twitter account metadata.

The number of followers a news-media organisation has on Twitter is an important metric, as is a proxy for the 'reach' of the account.

The most followed organisation in the dataset was the *New York Times* with 44,800,317 accounts following it at the time of collection on the 13th January 2020. The least-followed account was *CalMatters* with 15,069 followers. Interestingly, the follower counts were slightly higher for left biased organisations than for the right, in addition to being more numerous. This is shown via the followers distributions for each bias in Figure 3.4, which is indicative of the larger trend in social media of slightly left-leaning demographics [53].

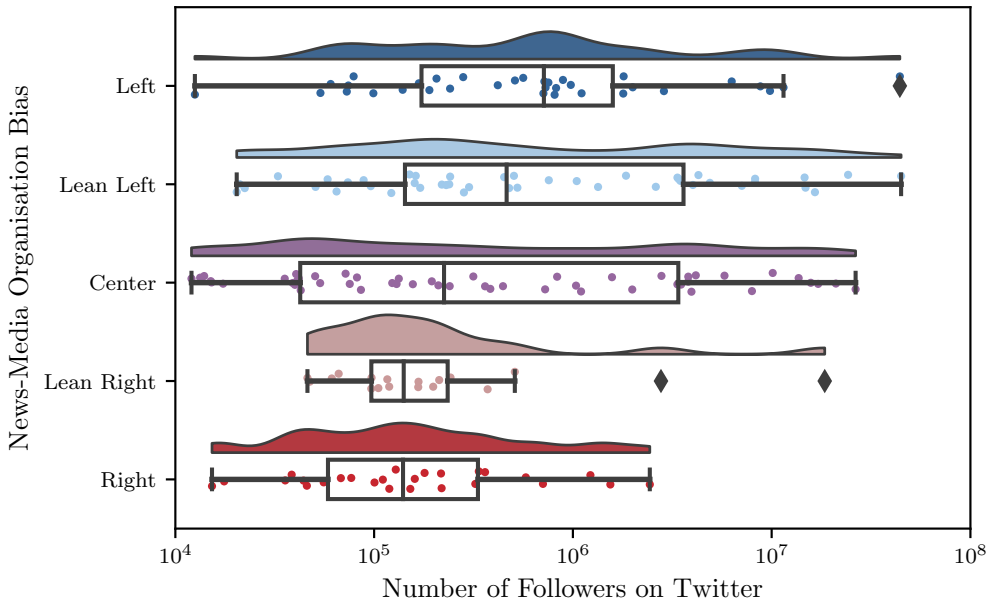


Figure 3.4: Number of followers on Twitter of news-media organisations included in the data. Grouping is according to Allsides bias.

3.1.1 Tokenization

We process the 2,977,980 tweets from filtered dataset into an array of words. As discussed in Subsection 2.3.1, the process of tokenization is applied to each string independently.

The TweetTokenizer[4] from the Natural Language Toolkit (nltk) in Python is used. This built-in tokenizer is specially designed for tweet-style strings and bundles several useful features.

For each tweet, four steps are applied:

[4] www.nltk.org/api/nltk.tokenize

1. Twitter account handles, which appear in the form '@account_name', are removed. Across all accounts, these handles make up 0.88% of the tokens. While some handles are contextual references, many are reference to piece authors. In the case of *latimes*, 69% of the references were to its own Twitter account. In general, these handles usually add uninformative differentiation between each organisations description of the news, and make matching sub-sequences of tokens harder.

2. Any sequence of a character that repeats more than three times is reduced to a maximum of 3. This has the effect of standardising text of similar meanings a common form, such as 'waaaaayyyy' and 'waaayyyyy' to 'waaayyy'. In doing so, this reducing makes matches between these tokens far more likely.

3. All URLs included in the text are removed. These URLs are almost always a link to an organisations own story associated with the tweet context. Many organisations have multiple domains or sister-sites making these hard to isolate without removing all URLs.

4. The string is converted to lower-case and split at each white-space, giving an array of individual words. This creates the sequence of tokens that can be most easily compared between organisations.

These steps are applied uniformly across the corpus to identify the flow of linguistic structure within news. In order to achieve this, we need to standardise the text to the most common possible format, such that text of similar phrasing and meaning will be matched between sources. These tokenization and text cleaning steps allow for these similarities to be expressed.

Vocabulary sizes

With the tweets of each Twitter account tokenized, we can begin to explore the vocabulary sizes. The vocabulary size is the number of unique tokens that exist in the collection of all tweets from a news-media organisation, as introduced in Section 2.3.1. Figure 3.5 shows the strong relationship between the amount of tweets produced and the number of unique words, with diminishing increases to vocabulary as new tweets are added.

The ratio of vocabulary size to number of tweets provides a first glimpse into the level of complexity in the language for each news-media organisation. Accounts that have a more specific focus/domain, such as poltical focused news (e.g. *The Hill*), increase their vocabulary at a slower rate as new tweets are added due to the consistently repeating domain specific words. In

contrast, an account such as the *The Guardian* produces a diverse array of content, and hence has a larger vocabulary size in contrast to its output.

We see the extreme variance in tweet activity reflected in the distribution of vocabulary size. The vocabulary sizes span from 2976 unique tokens (*ScienceDaily*) to 40824 (*The Guardian*).

This presents a key challenge for this work. We need to find information flows between news-media organisations which can have content corpora that differ in size by two orders of magnitude, and vocabulary sizes that can differ by up to one. This can exacerbate the problems already presented by natural language, and informs the need to normalise flows in later sections.

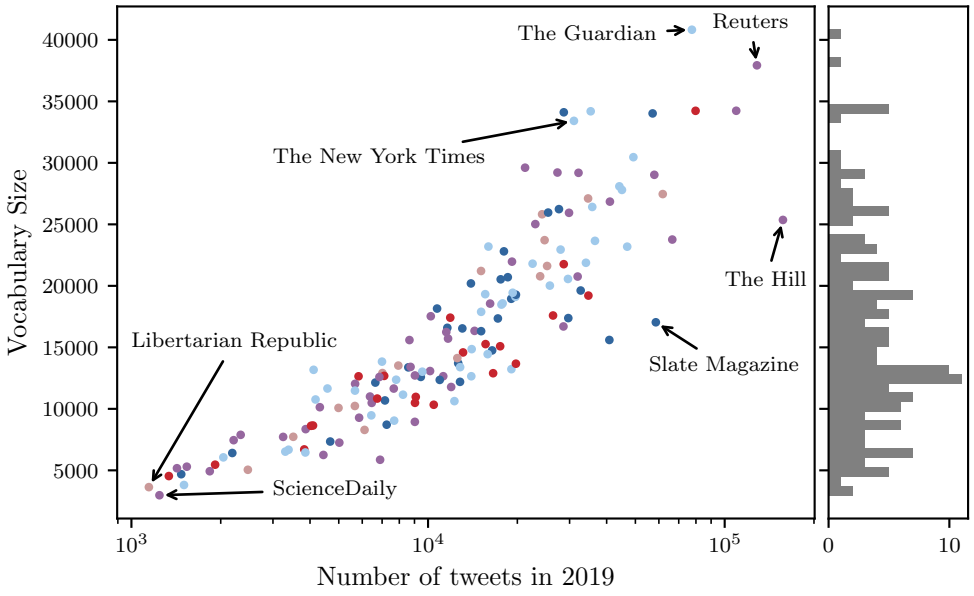


Figure 3.5: Vocabulary size from all tweets produced in 2019 for each news-media Twitter account and the total number of tweets produced.

3.1.2 Rank frequency distribution of vocabulary

In the remainder of this work we will often use this tokenized corpus of news-media tweets for our analysis. However, in some case we will want to generate synthetic text that has similar properties to our data here. As discussed in Subsection 2.3.2, we draw i.i.d. from a Zipf distribution to simulate such text. In order to match the properties of the data as best as possible, we fit a Zipf distribution to the data as show in Figure 3.6. To extract the fitted scaling parameter of a Zipf distribution, we can fit a linear line to the log

of each words frequency with the log of its frequency rank, in the corpus of all tweets. This fitting is meant to be approximate, as it is only used to inform the general range of Zipf law distributions used later, and the fitting of power-laws is often overdone [11].

As has been seen in other corpora [90], the rank-frequency distribution shows two different scaling regimes, where common words scale with $\alpha \approx 0.8$ and uncommon words scale with a much higher $\alpha \approx 2$, with an overall scaling parameter of $\alpha = 1.2$. Later in thesis this we will use a variety of values for α, usually in the range between 0.5 and 2.

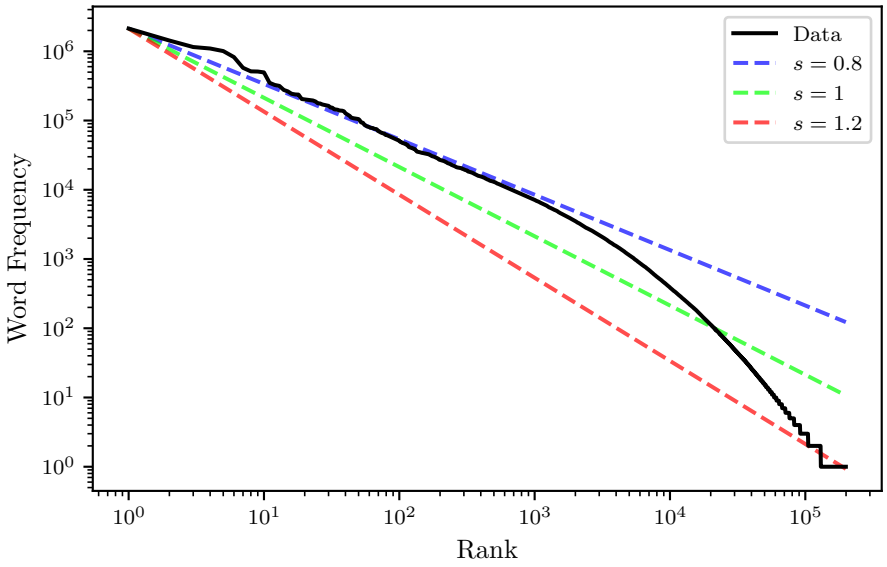


Figure 3.6: Word frequency of tokens in the corpus of all tweets produced by all news-media organisations compared the rank of the tokens by frequency. Zipf law distributions are also shown for varying scaling parameters, α.

Having now collected, cleaned, and performed a basic analysis of the data, we will now use it for two key purposes. Firstly, the development of the entropy estimation tools in Chapter 4 and information flow measures in Chapter 5 will use this dataset – and the synthetic generation of text motivated by it – to validate the techniques as reliable tools of information flow extraction. We then apply these tools to the data a final time to produce a network, the analysis of which is discussed in Chapter 6.

Chapter 4

Entropy rate estimation

> *"Semantic aspects of communication are irrelevant to the engineering problem."*
>
> Claude Shannon, *A Mathematical Theory of Communication*, 1948

Extracting information flows is a problem deeply rooted in information theory. To examine these flows requires tools to quantify and measure this information in the form of natural language. As discussed in Chapter 2, the words used to construct language have no qualitative meaning in the context of the numerical analysis. Thus the tools are comparative in nature. Indeed, information theory has been used extensively to compare properties of information in language [72, 19, 12].

In this chapter, we extend this philosophy in two key ways. We introduce a non-parametric entropy rate estimator and check its assumptions using real data. We then generalise this entropy rate to a cross entropy rate, developing a tool for analysing information flows. These new estimators are then made into a high speed open-source package which is applied to our news data.

4.1 Entropy rate estimation

Recall Definition 2.1.5 of the entropy rate of a stochastic process. While a useful theoretical tool, this can be very difficult to compute, requiring knowledge of the joint entropy for a infinite set of realisations.

To overcome this, we seek a way to estimate the entropy of the process from a known sequence of data. In 1998 Kontoyianni *et al.* [48] proved the convergence of a non-parametric entropy estimator for stationary processes.

Definition 4.1.1 (Kontoyianni Entropy Rate)**.** For a discrete valued stochastic process $\mathcal{X} = \{X_i\}$, with N realisations, the entropy rate is given by,

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{N \log N}{\sum_{i=0}^{N} \Lambda_i}, \tag{4.1}$$

where Λ_i is the length of the shortest subsequence starting at position i that does not appear as a contiguous subsequence in the previous i symbols X_0^i. This can also be obtained by adding 1 to the longest match-length,

$$\Lambda_i = 1 + \max\left\{\ell : X_i^{i+\ell} = X_j^{j+\ell}, 0 \le j \le N-i, 0 \le \ell \le N-i-j\right\}. \tag{4.2}$$

This idea of using matched sub-sequences of text draws from the original work by Lempel and Ziv [95] in compression algorithms. These algorithms attempt to compress a sequence down into the smallest possible representation, which at perfect efficiency would be the entropy, H. However these universal coding algorithms have no universal rate of convergence [76, 74] and in practice other approaches are often employed, tailored to the specific application at hand.

The idea of an entropy estimator based on match lengths was originally put forward by Grassberger [33] and proved consistent for independent and identically distributed (i.i.d.) processes and mixing Markov chains [75], stationary processes [47] and more generally to random fields [68].

Wyner and Ziv [92] showed that for every ergodic process the match length Λ_n grows like $\frac{\log n}{H}$ in probability. Extending from this notion Kontoyianni *et al.* showed the convergence of Equation 4.1 in stationary ergodic processes using the match-length Λ_i. This match-length in Equation 4.2 can be seen as the length of the next phrase to be encoded in the sliding-window Lempel–Ziv algorithm.

Conceptually, this match-length is simple. Figure 4.1 shows the calculation of two match-lengths at different time points of a line from Green Eggs and Ham by Doctor Seuss. At each index i, the elements immediately proceeding $(i, i+1, i+2, \dots)$ are compared to the history of elements before i. The matches of length k are found such that the elements from j to $j+k$ perfectly match the elements from i to $i+k$, for any $j < i$ where k is then maximised. This search only considers the length of the match, regardless of its location in the history.

Even before its formalisation by Kontoyianni *et al.*, similar estimators had appeared in the literature applied to experimental data to determine the entropy rates of processes [15, 16, 28, 46].

Moving forward we will assume any any discussion of the *entropy rate* of a single process is assumed to be the *Kontoyianni entropy rate* of that process, unless otherwise stated.

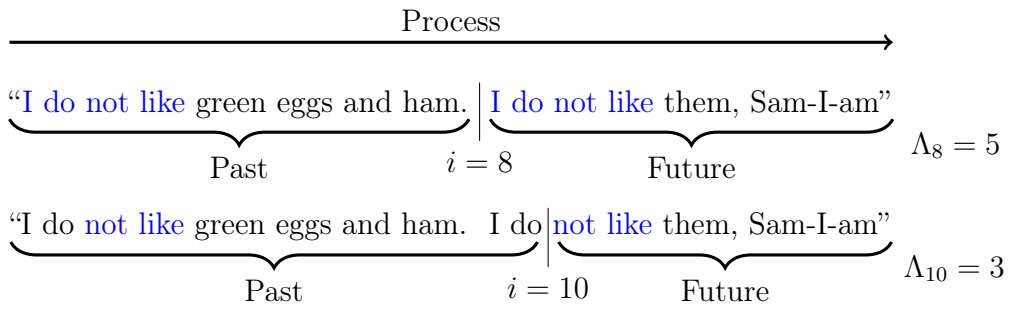


Figure 4.1: An example calculation of the match-length based Λ_i applied to a words in a line of text from Green Eggs and Ham by Doctor Seuss. Blue text is words which has been matched from past to the future. As i changes, the longest match length possible starting at index i will change.

4.2 Assumptions of entropy rate estimation

The proof of convergence of this entropy rate places some limits on the process of investigation. In particular, three assumptions are made for convergence: ergodicity, stationarity and the Doeblin Condition (DC).

The Doeblin Condition is a reasonably weak condition, but is fundamental in the proof of the convergence. Simply put, the DC requires that after an arbitrary r time steps, every state is possible again with positive probability [47]. More formally, the definition is as follows.

Definition 4.2.1 (Doeblin Condition (DC))**.** There exists an integer $r \geq 1$ and a real number $\beta \in (0, 1)$ such that, for all $x_0 \in \mathcal{A}$,

$$P\left\{X_0 = x_0 \mid X_{-\infty}^{-r}\right\} \geq \beta,$$

with probability one.

Fortunately, as Kontoyianni *et al.* themselves state, the DC is "certainly satisfied by natural languages" [48].

In contrast, the assumptions of ergodicity and stationarity are harder to confirm. A long-standing assumption of information theory is that natural language can be modelled by a stationary process [71, 72, 20]. The assumption, while flawed, has a long precedent and we use it again in this work.

Much of the literature including the work of Kontoyianni assume ergodicity of natural language, however some suggest that language should be modelled by a *strongly nonergodic* stationary process [21]. In brief, this contention is founded upon the idea that any given collection of text, such as a book, has a topic containing a small finite subset of words. Which suggests that its text cannot explore the full state space of language. While

well founded, our interest is not to look at the entropy rate of the English language as a whole, but rather to look at the entropy rate of individual text streams, which can all explore the state space of news under consideration. As such, the assumptions of ergodicity and stationarity appear justified in the context of the problem.

Convergence

With the assumptions of the proof addressed, the challenge of entropy rate estimation convergence needs to be examined. The entropy rate defined in Equation 4.1 is based upon an infinite set of data. In reality, we have finite data, and need to examine the convergence of a modified estimator.

Definition 4.2.2 (Kontoyianni Entropy Rate Estimator)**.** The Kontoyianni Entropy Rate in Definition 4.1.1 can be estimated on a finite stochastic process $\mathcal{X} = \{X_i\}$, with N realisations, by

$$\hat{h} = \frac{N \log N}{\sum_{i=0}^{n} \Lambda_i}, \tag{4.3}$$

where Λ_i is, as earlier, the length of the shortest subsequence starting at position i that does not appear as a contiguous subsequence in the previous i symbols X_0^i.

$$\Lambda_i = 1 + \max\left\{\ell : X_i^{i+\ell} = X_j^{j+\ell}, 0 \leq j \leq N - i, 0 \leq \ell \leq N - i - j\right\}. \tag{4.4}$$

To examine the convergence of this estimator, a model of language can be used to generate sequences of text, upon which we can estimate the entropy rate.

Figure 4.2 shows the convergence of the estimator for a set of i.i.d. realisations of a Zipf distribution distribution. As discussed in Subsection 2.3.2, the Zipf distribution is a common tool for generating simple text due to its similarity to the power-law distributions of vocabulary seen in real corpora. The Zipf distribution can be used with a number of scaling parameters, α, where larger scaling parameters tighten the distribution, reducing the observed vocabulary size of the sequence and hence the entropy.

Sequences are generated with 30,000 i.i.d. elements drawn from a Zipf distribution and the entropy rate of estimate of the process is calculated at each timestep between 1 and 30,000 using Equation 4.3 applied to only the elements before that timestep. Estimates of entropy start very low when few elements are available and rapidly rise as new elements add complexity. Within the first few hundred timesteps entropy estimates can vary between

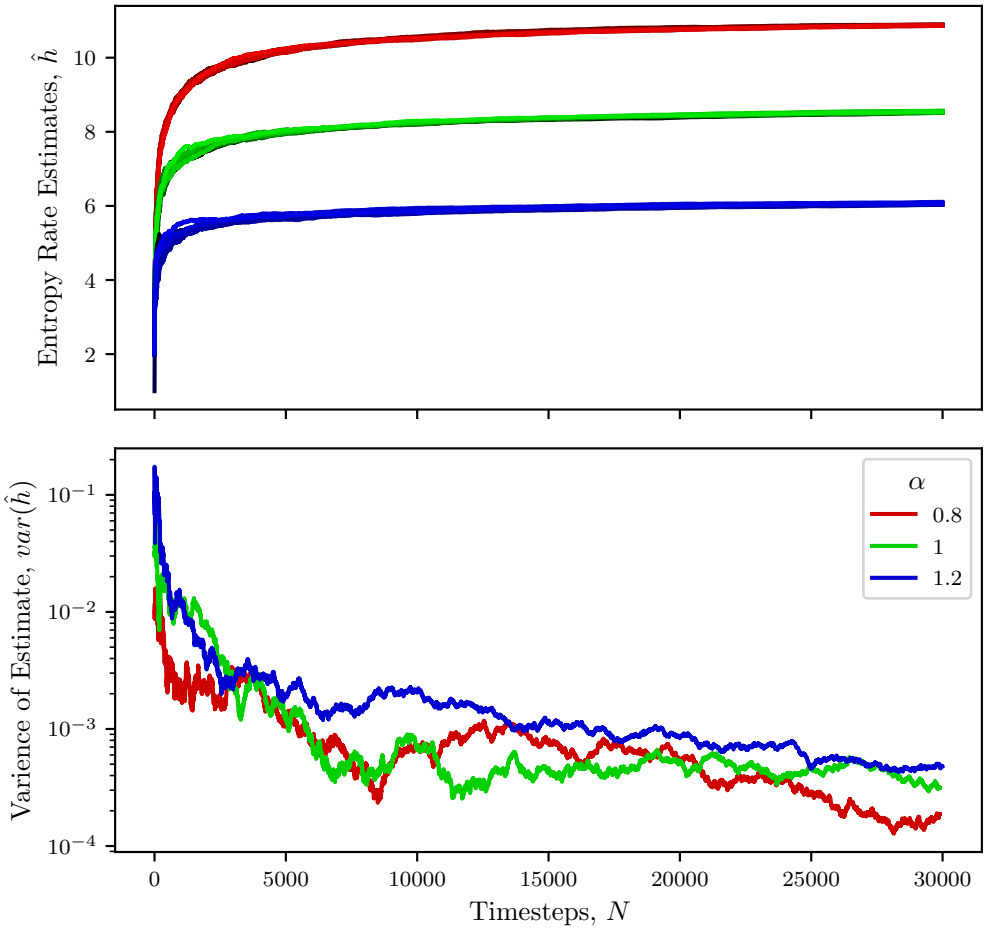


Figure 4.2: Convergence of the Kontoyianni entropy rate estimator on sequences of i.i.d. Zipf distribution realisations with varying Zipf distribution rates, α.

timesteps as new elements are added matching or not-matching previous elements. This is reflected in the high variance between entropies estimates for Zipf process with the same scaling parameter in these early stages. As the number of timesteps included reaches 5000 to 7500 the apparent bias from the asymptotic rate and variance of the estimates are significantly reduced and begin plateauing.

As more timesteps are added, the estimator continues to converge to the asymptotic entropy and the variance between estimates continues to reduce. High complexity sequences take longer to converge to this entropy, but achieve suitably small levels of bias and variance within 15,000 timesteps even for high entropy sequences. This is an important finding given the speed of calculating these estimates. The algorithm to calculate the match-lengths needed for estimating the entropy is $O(N^3)$ time complexity for the number of included timesteps N. As a result, calculations are extremely slow as the number of timesteps gets large. When performing simulations a parsimonious choice of simulation length is advantageous in allowing multiple simulations to be run. Hence, for Zipf distributions a simulation length of 15,000 is deemed sufficient for convergence to the entropy.

To confirm the validity of this approach, we can examine the known entropy rate of the Zipf processes. As proved in Subsection 2.1.3, the entropy rate of an i.i.d. process is simply the entropy of each element. In the case of a Zipf distribution the distribution has entropy[1],

$$\frac{s}{H_{n,\alpha}} \sum_{k=1}^{n} \frac{\ln(k)}{k^{\alpha}} + \ln\left(H_{n,\alpha}\right) \tag{4.5}$$

where $H_{n,\alpha}$ is the nth generalized harmonic number defined by,

$$H_{n,\alpha} = \sum_{k=1}^{n} \frac{1}{k^{\alpha}},$$

and n is the total state space size of the Zipf distribution. Many approaches to calculating this entropy use the asymptotic entropy as $n \to \infty$. This draws on the result that $\lim_{n\to\infty} H_{n,s} = \zeta(\alpha)$, where ζ is the Riemann zeta function. While this asymptotic approach works well for numbers well above 1, the Riemann zeta function diverges to infinity at $\alpha \to 1$. As such, the analytic entropy degenerates as $\alpha \to 1$ and is poorly defined for $\alpha < 1$.

[1]Proof: The probability of a word of rank k being selected from a pool of N elements using scaling parameters α is $(k^{\alpha} H_{N,\alpha})^{-1}$. Hence, the entropy of each individual element is $\sum_{k=1}^{N} (k^{\alpha} N_{N,\alpha})^{-1} \ln\left((k^{\alpha} N_{N,\alpha})^{-1}\right)$. Using $H_{N,\alpha} = \sum_{k=1}^{N} \frac{1}{k^{\alpha}}$, this can be rearranged to $\frac{\alpha}{H_{N,\alpha}} \sum_{k=1}^{N} \frac{\ln(k)}{k^{\alpha}} + \ln\left(H_{N,\alpha}\right)$.

In contrast, using a finite choice of n results in well defined processes and entropies for all $\alpha > 0$. A choice of $n = 199338$ is made to match the observed total vocabulary size seen in the news-media Twitter data as explored in Section 3.1.1. For high values of α, the two asymptotic and finite entropies are very close (e.g for $\alpha = 1.5$ the entropies are 3.18158 and 3.18368 respectively), and only differ significantly as α approaches 1. This finite entropy calculation allows the model to more accurately match the Zipf scaling parameters fitted to real text data, which are often closer to or below 1 [90].

Using simulations of the Zipf process for a variety of values of the scaling parameter α, we compare how the estimated entropy rate compares to the 'true' entropy rate as calculated above. In Figure 4.3 simulations are run for values of α in the range [0.01, 2] with increments of 0.01. High values of α in this range have a progressively lower entropy, as the skewness of the distribution becomes more extreme. This distribution results in a large number of realisations of low rank words creating repeated sequences which lower both the analytic entropy rate and the entropy rate estimate. Values above 2 reduce the entropy rate in vanishingly smaller increments.

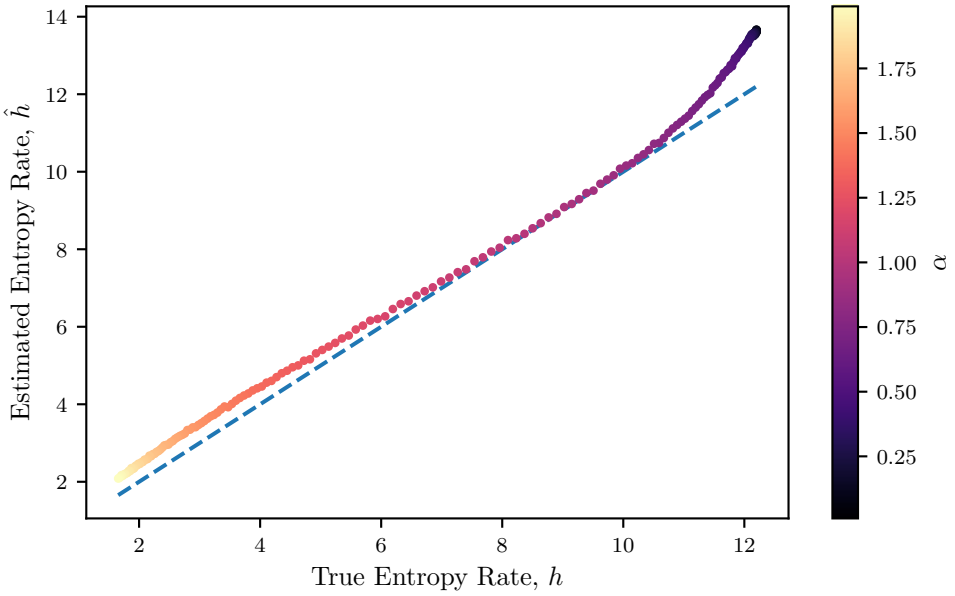


Figure 4.3: Estimated entropy rates and analytic entropy rates of sequences of 20,000 i.i.d. Zipf distribution random variables with scaling parameter α. Dashed line represents the true entropy rate equalling the entropy rate estimate. As values for α approach 0 the high variance of the distributions results in poor estimates due to the finite sample of the Zipf distribution.

For values of α between 2 and 0.5, the entropy rate estimate appears to be a rough upper bound on the true entropy rate of the process. This upper bound is only achieved with sufficient lengths of sequences such that the estimator can converge to this upper bound. Indeed, given sufficient length the variance of the estimates on sequences drawn from the same distribution is very low. This is in contrast to the bias of the estimate, which varies with the changing scaling parameter.

When α becomes lower than 0.5, the Zipf distribution becomes more evenly distributed with reduced skew. This results in a larger probability of low rank word occurrences, producing a process where many words appear very few times. As a result, the finite nature of the sequence results in a entropy rate estimate that grows faster than the true entropy, increasing the bias for these high entropy sequences.

In general, the convergence of the estimator is sufficient, with a slight caveat: while the estimator convergences to an estimate tightly with very little variance, the bias of the estimate is not constant and varies with the complexity of the sequences. While this finding is itself interesting and warrants future work, the estimator is both consistent and its estimates appear monotonic with the true entropy rate. As such, we will use this estimator to approximate the true entropy rate moving forward, with a cautious eye to the possible effects of this inconsistent bias.

To extend from this result, we apply the same approach replacing Zipf distribution generation with text from the news-source Twitter data. 5000 tweets are drawn uniformly from the collection of all tweets from all news-media outlets. These tweets are tokenized and concatenated into a single sequence of natural language text ranging from 85000 to 90000 tokens.

Unlike the case of Zipf, we cannot show a true entropy rate for this distribution, as it's unknown, but can demonstrate its convergence. Each sample has the entropy rate calculated using only the first N tokens, up to the length of 60,000 tokens. Figure 4.4 shows the clear convergence of the estimator on the real data, approaching an entropy rate of 7.53. As with the Zipf simulations, the variance of the estimated entropy rates of the samples reduces as more tokens are included in the estimate. The real data takes longer to converge, needing up to 50,000 tokens to achieve a variance of less that 10^{-3}.

Self entropy rate summary

Before moving away from the entropy rate of a single process, we visualise the how the Kontoyianni entropy rate estimator works for real data. Figure 4.5 shows a simplified version of the conceptual entropy rate calculation.

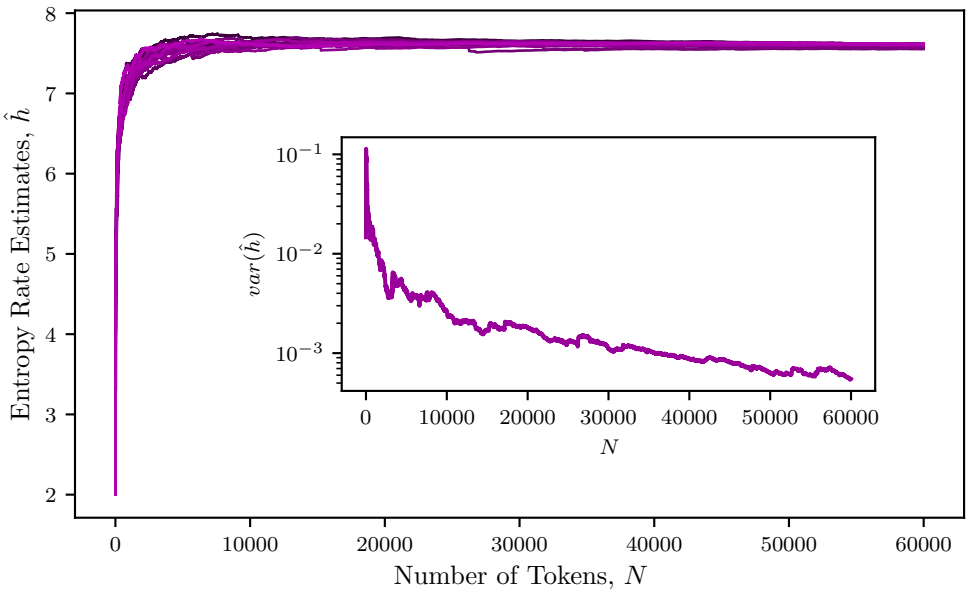


Figure 4.4: Convergence of the Kontoyianni entropy rate estimator on sequences words generated by drawing tweets uniformly without replacement from the pool of all tweets produced by all news-media organisations.

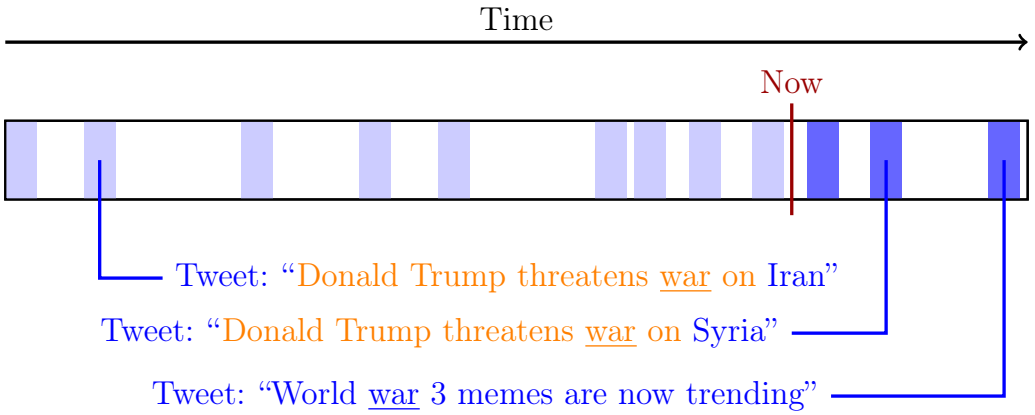


Figure 4.5: A conceptual diagram of entropy rate estimation using the Kontoyianni entropy rate estimation. Tweets shown as blue rectangles are positioned in time and contain textual content. Content proceeding the position *Now* will have snippets of text matched with text from the history of the process, denoted by orange and underlined text. These text matches are used to calculate Λ_i which is used in the entropy estimate.

For a given Twitter user, tweets appear sequentially, separated in time. At any given time, the content of the immediate future of that user's tweets is compared to the entire history of the tweets before that time. This process is then repeated for all possible times in the data. In essence, the calculation of the Λ_i's is a repeated examination of how much of the immediate complexity at timestep i can be described using the history of the sequence. In total, this estimator provides an average of how many bits are needed to describe the future of this process given its past at any point.

4.3 Cross entropy rate

To create a notion of information flow, we need to move beyond looking at an individual source in isolation. To do so, we need a tool of comparison between sources rooted in our tools from information theory. We find such a tool in a generalisation to a Kontoyianni cross entropy rate.

Similar to the extension of entropy,

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log p(x) = -\mathbb{E}[\log P(X)]$$

to cross entropy,

$$H(p, q) = \sum_{x \in \mathcal{X}} p(x) \log q(x) = -\mathbb{E}_p[\log q(X)],$$

in Definition 2.1.4, we can generalise our notion of Kontoyianni entropy rate from Definition 4.1.1 to a *cross* entropy rate which we will call the Kontoyianni cross entropy rate. This Kontoyianni cross entropy rate comes in two forms, a full cross entropy and a time-synced cross entropy.

Definition 4.3.1 (Kontoyianni Full Cross Entropy Rate)**.** The cross entropy rate of a target process $\mathcal{T}$ coded from a source process $\mathcal{S}$ can be estimated via,

$$H(\mathcal{T}||\mathcal{S}) = \frac{N_{\mathcal{T}} \log_2 N_{\mathcal{S}}}{\sum_{i=1}^{N_{\mathcal{T}}} \Lambda_i(\mathcal{T}|\mathcal{S})} \tag{4.6}$$

Where $N_{\mathcal{X}}$ is the length of process $\mathcal{X} \in \{\mathcal{T}, \mathcal{S}\}$ and $\Lambda_i(\mathcal{T}|\mathcal{S})$ is given by the shortest subsequence starting at position i in the target $\mathcal{T}$ that does not appear as a contiguous subsequence anywhere in the source $\mathcal{S}$.

$$\Lambda_i(\mathcal{T}|\mathcal{S}) = \max\left\{\ell : T_i^{i+\ell} = S_j^{j+\ell}, 0 \leq j \leq N_{\mathcal{S}}, 0 \leq \ell \leq \min(N_{\mathcal{S}} - j, N_{\mathcal{T}} - i)\right\}, \tag{4.7}$$

where T_a^b and S_a^b are continuous subsequences starting from index a to index b of the target, $\mathcal{T}$, and source, $\mathcal{S}$, processes respectively.

This approach to a cross entropy matches segments of text in the target to segments of text anywhere in the source in the same manner that the Kontoyianni entropy rate matched segments of text in the future of a process from a index, i, to the history before the index. In contrast to the entropy rate estimate, which asked how much information was needed on average to *describe the future of a source from its past*, this cross entropy rate estimate is asking how much information is needed on average to *describe the target given full knowledge of the source.*

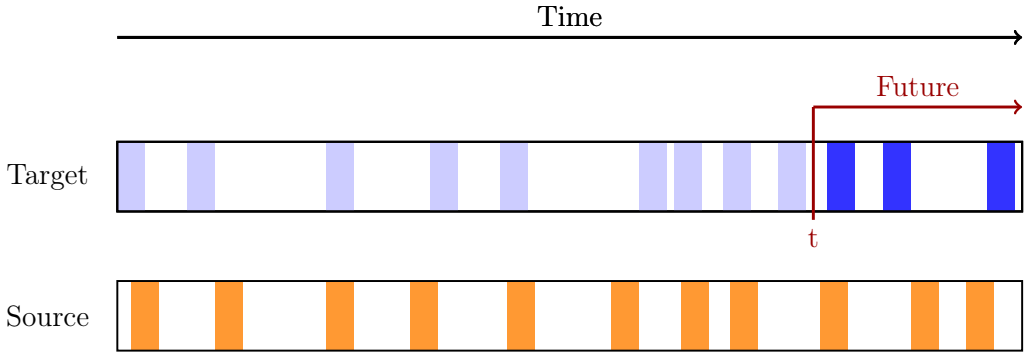


Figure 4.6: A conceptual diagram of **Kontoyianni full cross entropy rate** estimation. Tweets shown as rectangles are positioned in time for both a target and source, containing textual information. Content in the source is matched with content in the immediate future of the target for a given time point, t, to calculate match-lengths. This time point is shifted along the target timeline to average match-lengths and calculate the full cross entropy rate.

This full knowledge over all of the process gives the estimator the 'Full' in its title, but presents a impropriety. In the context of our problem, this estimator is cheating by viewing the future of news through the lens of the source process.

As Figure 4.6 illustrates, the text subsequence match in the target could be drawn from future time points in the source. Restated, the cross entropy rate will, in-part, be describing how much information is needed to encode the future of a piece of target news already knowing the future of the news from another source. While this may be an interesting insight in itself, and could be used in future work to format a notion of information divergence, it doesn't probe the underlying process of *information flow* with which this thesis focuses.

Rather than looking at the entire lifetime of the source during the matching calculations, we can reduce our search space to the text that occurred in

the *past* of the source. To achieve this we use an important piece of our data, the time that tweets occurred. For each word in the target process, T_i has an associated time with it, $t(T_i)$. When matching the future of $\mathcal{T}$, starting from an index i, we can reduce the source process, $\mathcal{S}$ to only the words that were themselves tweeted before time $t(T_i)$.

Put simply, we can alter the Kontoyianni full cross entropy rate to a time-synced cross entropy rate by replacing the full source process, $\mathcal{S}$, with a time reduced source process $\mathcal{S}_{\leq t(T_i)}$. This can be seen visually in Figure 4.7 and is formally defined as follows.

Definition 4.3.2 (Kontoyianni Time-synced Cross Entropy Rate)**.** The time-synced cross entropy rate of a target process $\mathcal{T}$ coded from a source process $\mathcal{S}$ can be estimated via,

$$H(\mathcal{T}||\mathcal{S}) = \frac{N_{\mathcal{T}} \log_2 N_{\mathcal{S}}}{\sum_{i=1}^{N_{\mathcal{T}}} \Lambda_i(\mathcal{T}|\mathcal{S}_{\leq t(T_i)})} \tag{4.8}$$

Where $\Lambda_i(\mathcal{T}|\mathcal{S}_{\leq t(T_i)})$ is given by the shortest subsequence starting at position i in target $\mathcal{T}$ that does not appear as a contiguous subsequence in the time reduced source $\mathcal{S}_{\leq t(T_i)}$ where,

$$\mathcal{S}_{\leq t(T_i)} = \{S_j | t(S_j) \leq t(T_i), \forall i\}. \tag{4.9}$$

Which gives,

$$\begin{aligned}\Lambda_i(\mathcal{T}|\mathcal{S}_{\leq t(T_i)}) = \max\{\ell : T_i^{i+\ell} = S_j^{j+\ell}, 0 \leq j \leq N_{\mathcal{S}},\\ 0 \leq \ell \leq \min(N_{\mathcal{S}} - j, N_{\mathcal{T}} - i)\},\end{aligned}$$

where T_a^b and S_a^b are continuous subsequences starting from index a to index b of the target, $\mathcal{T}$ process, and the time reduced source, $\mathcal{S}_{\leq t(T_i)}$, respectively.

This time-synced cross entropy rate is testing not just the differences in the language processes of the source and target, but also measuring what information in the target is present in the source's history. This is an important distinction, as it allows us to probe a very important aspect of our data, namely, the time in which news is created.

If a piece of information appears earlier in the source than in the target, it will be detected during the match length search, resulting in a lower entropy. This is to say, in the context of news, if the source breaks a story first, *less* information is required to describe the subsequent news output from the target.

Conversely, if a target produces a piece of information before the source, then that information will not appear in the history of the time-synced source

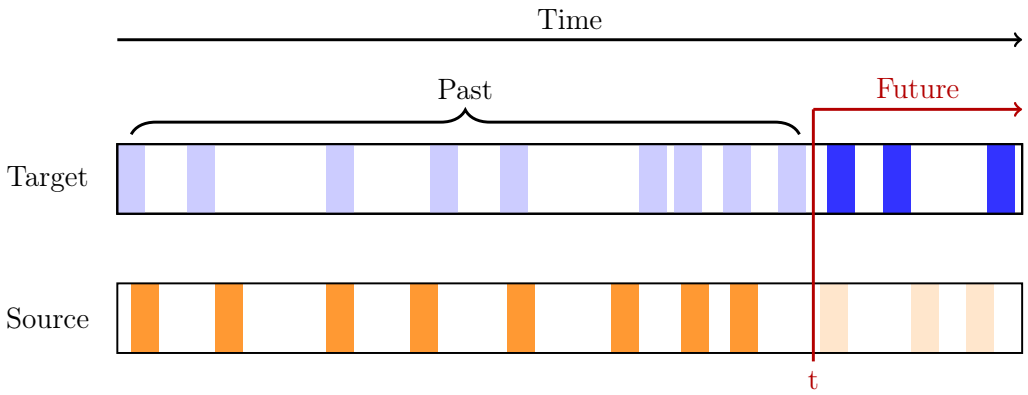


Figure 4.7: A conceptual diagram of **Kontoyianni time-synced cross entropy rate** estimation. Tweets shown as rectangles are positioned in time for both a target and source, containing textual information. Content in the source that occurs before time t is matched with content in the immediate future of the target from t to calculate match-lengths. This time point is shifted along the target timeline to average match-lengths and calculate the time-synced cross entropy rate.

during the match-length search. This will result in lower values of Λ_i for that piece of information, which raises the cross entropy rate.

From this, we can find that, on average, if a source produces information earlier than a target, the cross entropy rate, $\hat{h}(\mathcal{T}|\mathcal{S})$, will be lower than if the target produces information earlier than the source. This method of examining who produces information first can be extended into a notion of *information flow*, a discussion we will leave for Chapter 5.

4.3.1 Validating the assumptions of cross entropy estimation

To validate these new cross entropy rate estimators a similar process is performed as with the entropy rate. Fortunately, many of the assumptions transfer over directly.

In the case of ergodicity, stationarity and the Doeblin Condition, all three are properties of the process under investigation, which we argued above are well founded. We introduce a additional condition on the processes for the sake of cross entropy rates, namely that the processes have the same state space. This additional condition extends the earlier conditions to apply jointly between both the source and target processes. While not all news-media organisations will have the same realisations of words in their corpus,

the processes could be reasonably thought to have the same state space, as all processes are using English and discussing similar topics.

This then leads naturally to the next question of convergence. Following a similar process of uniform withdrawals from a distribution will result in functionally similar convergence to the entropy rate above. Indeed, this can be seen in Figure 4.8, where tweets are draw uniformly from the pool of all tweets and cross entropy rates are calculated on the resulting sequences. Here, the cross entropy rate is denoted $\hat{h}_\times$ to distinguish from the entropy rate estimates $\hat{h}$. This figure is, as expected, functionally similar to Figure 4.4 from the entropy rate section above. This cross entropy rate calculation is fundamentally no different from the original entropy rate calculation with these sequences, as a randomly selected history of another process is just a useful and a randomly selected history the original process when both draw from the same distribution.

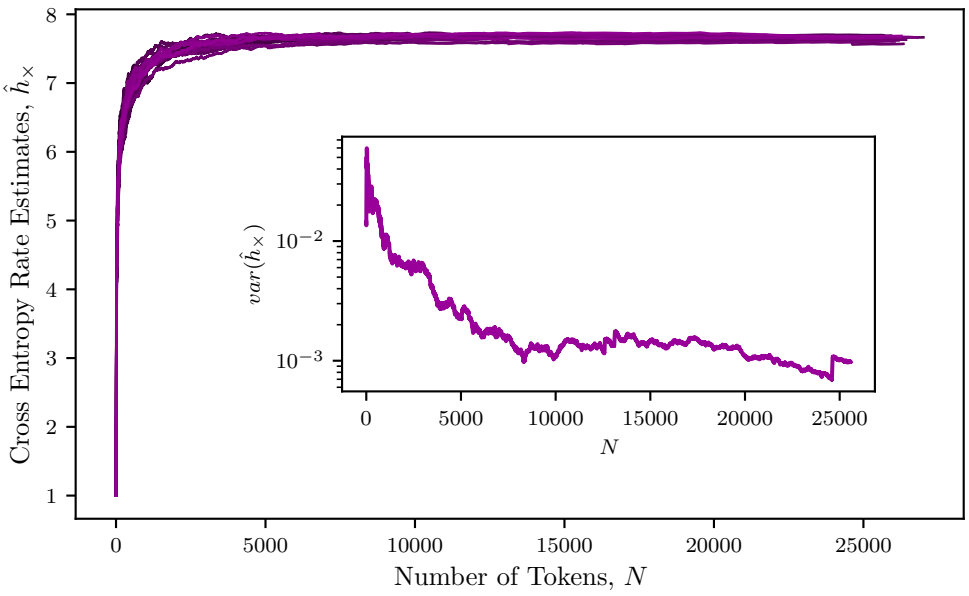


Figure 4.8: Convergence of the Kontoyianni time-synced cross entropy rate estimator on pairs of sequences independently generated by drawing tweets uniformly without replacement from the pool of all tweets produced by all news-media organisations.

A more nuanced investigation of the cross entropy convergence emerges when we utilise processes with *different* distributions. Figure 4.9 does exactly this. Using pairs of scaling parameters for different Zipf distributions, simulations of 30,000 long i.i.d. processes are made with separate target

and source processes. The cross entropies rates are then estimated across these pairs. When the scaling parameters are equal, $\alpha_{source} = \alpha_{target}$, we observe the same entropy and convergence pattern as we would for a single i.i.d. process with that distribution.

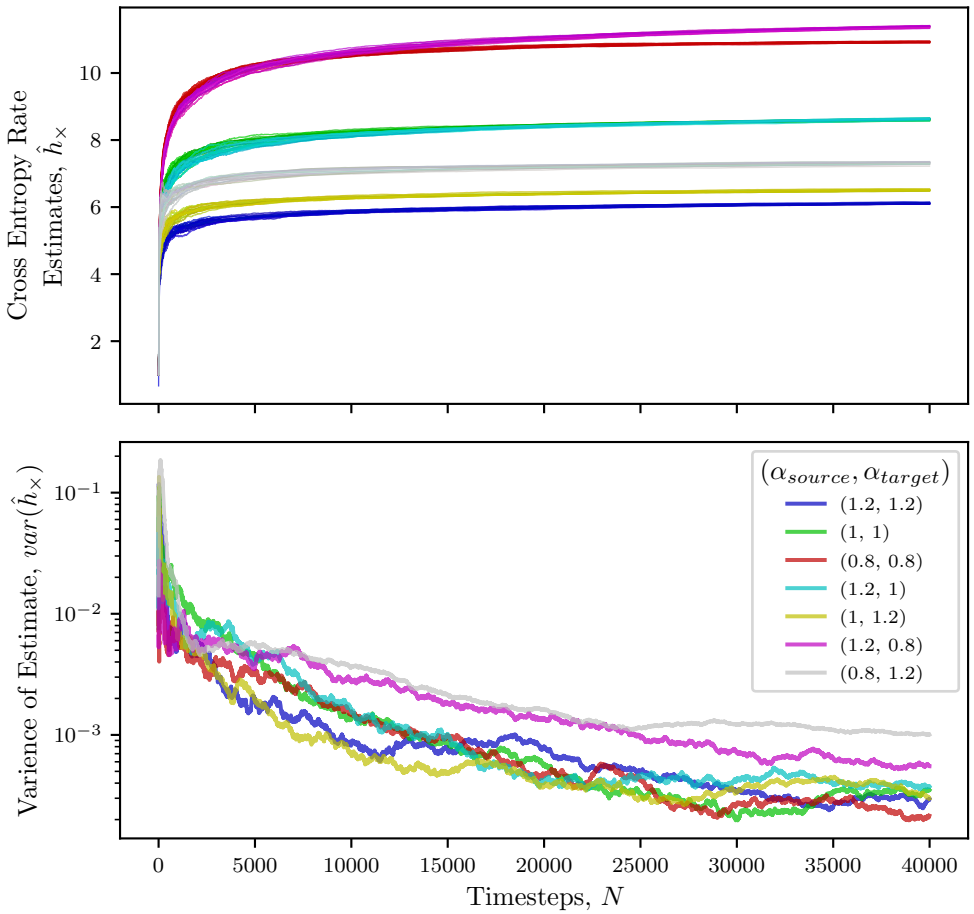


Figure 4.9: Convergence of the Kontoyianni time-synced cross entropy rate estimator on pairs sequences of i.i.d. Zipf distribution realisations with varying pairs of Zipf distribution rates, α_{source} and α_{target}.

When $\alpha_{source} \neq \alpha_{target}$ the entropy rates vary, and are not always similar to the entropy rate of the target or the source. Most cross entropy rates converge at a similar pace to the entropy rates above, although large deviations between distributions can reduce this rate of convergence. In particular when the source has significantly higher complexity than the target, as in the case of $(\alpha_{source}, \alpha_{target}) = (0.8, 1.2)$, the entropy takes much longer to converge

as it requires longer for common patterns to occur in the high complexity source for sequence matching.

4.3.2 Predictability

With the Kontoyianni cross entropy rate estimator converging, we also want to generalise the notion of maximal predictability. We introduced maximal predictability, $\pi^{max}(S, N)$, in Definition 2.1.6 as it is a robust way of determining an upper bound on how well the future of a process could be predicted from its past. The use of the state space size, N, allows for normalisation of the entropy rate when state spaces are large. So as to avoid confusion with the number of tokens mentioned above, the state space size will be represented as V in the section.

In order to generalise this notion of maximal predictability, we need to identify the two replacements for the inputs. The replacement for entropy rate is trivially the cross entropy rate, but the choice of state space size is more nuanced. Three choices are possible, the state space of the source, V_{source}, the target, V_{target}, and the union of the state spaces V_{union}.

The original motivation behind the maximal predictability is to normalised the entropy by *how complex the system we are trying to predict is*. This complexity is a property of the target. While a complex source does change how well the target can be predicted, it is the property of the target that needs to be controlled for. As such, the choice of V_{target} is taken and referred to as simply V in the following definition.

Definition 4.3.3 (Maximal Cross Predictability)**.** For two processes S and T with cross entropy $H(T||S)$ from the source to the target and state space size V of the target, the maximal cross predictability, $\pi^{max}(T||S)$ can be found numerically using,

$$H(T||S) = H(\pi^{max}(T||S)) + (1 - \pi^{max}(T||S)) \log(V - 1) \tag{4.10}$$

4.3.3 A note on package development

As stated earlier, a key challenge in estimating these Kontoyianni cross entropy rates is calculation speed. With time complexities of $O(N^3)$ on the number of input tokens, efficient code is necessary to allow for estimation using long sequences. To achieve speed and contribute to this field of work more broadly, a speed-focused open source package was developed to help researchers efficiently and easily calculate entropy rates such as those discussed above. The important snippets from code contributions of the pack-

age, named 'ProcessEntropy', are available in Appendix C and this section will outline tools used in speeding the code up.

Fundamental complexity limitations of the subsequence matching algorithm mean that speed improvements need to come from smart implementation. Two techniques used are interesting enough to discuss briefly here, hashing and just-in-time compilation.

In the context of analysing language like a process, each word is simply an element of the state space. As such, each word can be assigned a unique number to represent its position in the state space. In doing so it allows the computations of the Λ_i's to be performed using much faster integer operations. In practice developing such a lookup table for each word in the state space is slow and unnecessary. In the ProcessEntropy package, words are converted to 32 bit unsigned integers using Fowler–Noll–Vo hashing. Fowler–Noll–Vo is a non-cryptographic hash function which is fast to compute. This converts each words to the same integer every time, with almost no collisions.

The second major speed improvement is drawn from the use of just-in-time (JIT) compilation. Python is commonly used in scientific settings as an interpreted language. In most contexts the Python interpreter allows variables to exist as dynamic types. As such, the exact same function can often be applied to integers, strings or other objects interchangeably. As a result, a larger number of type checks and other control operations are required during runtime, slowing down computation. JIT compilers such as Numba[2] operate by observing variable types at the start of runtime and generating optimized machine code.

Figure 4.10 shows a comparison of the optimized methods compared to alternatives. The ProcessEntropy package consistently performs the fastest, using the speed improvements listed above and efficient implementation. In contrast, the unoptimized code (written in pure Python without fixed type compiling or parallelisation) performs over two orders of magnitude slower for all input sizes. For comparison, an alternative algorithm is shown using the highly optimized `.contains()` method included in `stringlib` library from Cython. This method uses a simplified version of the Boyer-Moore string-search algorithm, incorporating ideas from Horspool and Sunday [50]. Even with the alternative algorithm written in C, the ProcessEntropy code outperforms consistently. The package is available on the Python Package Index (PyPi) for interested researchers[3].

[2] https://numba.pydata.org/
[3] https://pypi.org/project/ProcessEntropy/

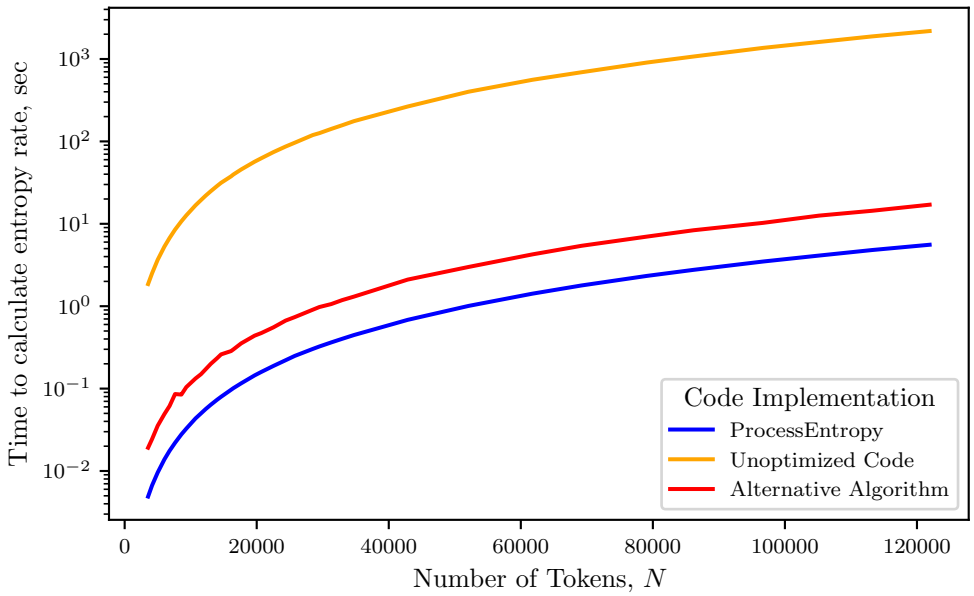


Figure 4.10: A speed comparison of implementations of the Kontoyianni entropy rate estimator. ProcessEntropy uses code from the package of the same name, unoptimized code is the same algorithm as ProcessEntropy without type and compile optimizations and Alternative Algorithm is an optimized alternative algorithm using the built-in `.contains()` method from `stringlib` library.

4.3.4 Running estimations

To close this chapter, the tools developed throughout can now be applied to the news dataset introduced in Chapter 3. This news data contains 154 year-long Twitter content streams, with the mean length of these streams at 337,284 tokens. Noting the earlier discussion on time complexity, each stream has its entropy rate calculated and each pair of news-media outlets have their time-synced cross entropy rate calculated in both directions. In total this is 23,716 entropy rate estimations on these very long sources. With the efficient ProcessEntropy package, these calculations take well over a week, and this analysis would not be computationally tractable on the available hardware using the other implementations of the code.

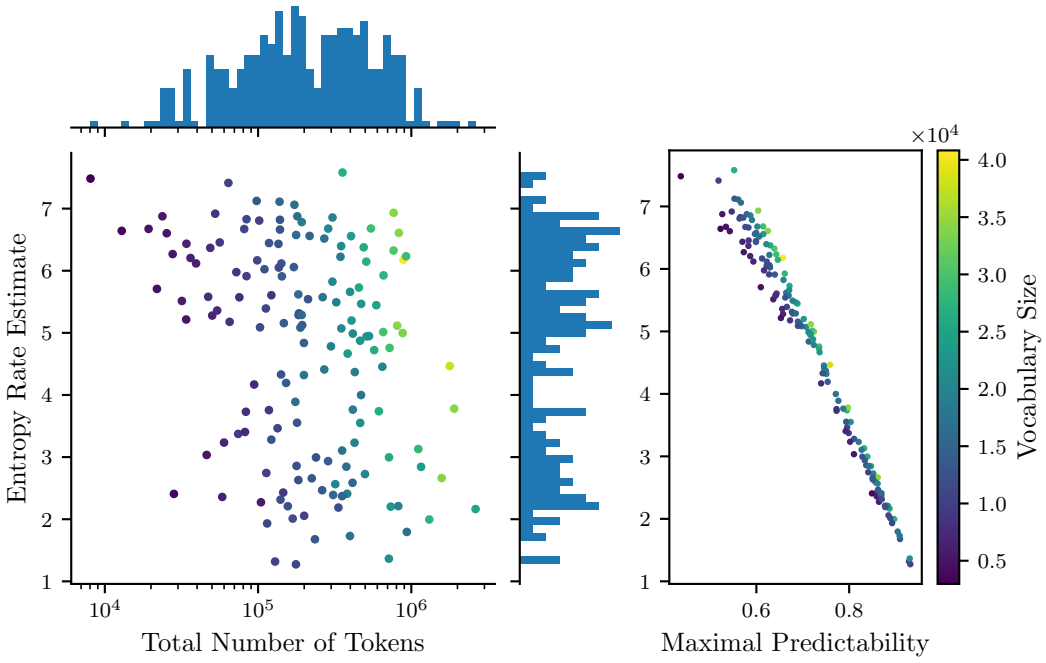


Figure 4.11: Entropy rate estimates for the Twitter timelines of 154 news-media organisations during the calendar year of 2019. (a) shows the limited relationship between the total number of tokens in the Twitter text history and the entropy rate estimates. (b) shows the tight relationship between the entropy rates estimate and its derived maximal predictability, with higher variances seen at high entropies.

Figure 4.11 shows these entropy rate estimates on the news-media Twitter histories. Importantly, the entropy rate estimate shows very limited corre-

lation with the total length of the content (number of tokens) or with the vocabulary size. This indicates that the estimate entropy rates have converged. Further, we see an expected strong correlation between the entropy rate estimate and the maximal predictability. Notably, the variance between the two increases for higher entropy rates, where larger vocabulary sizes can have an outsized effects on estimates.

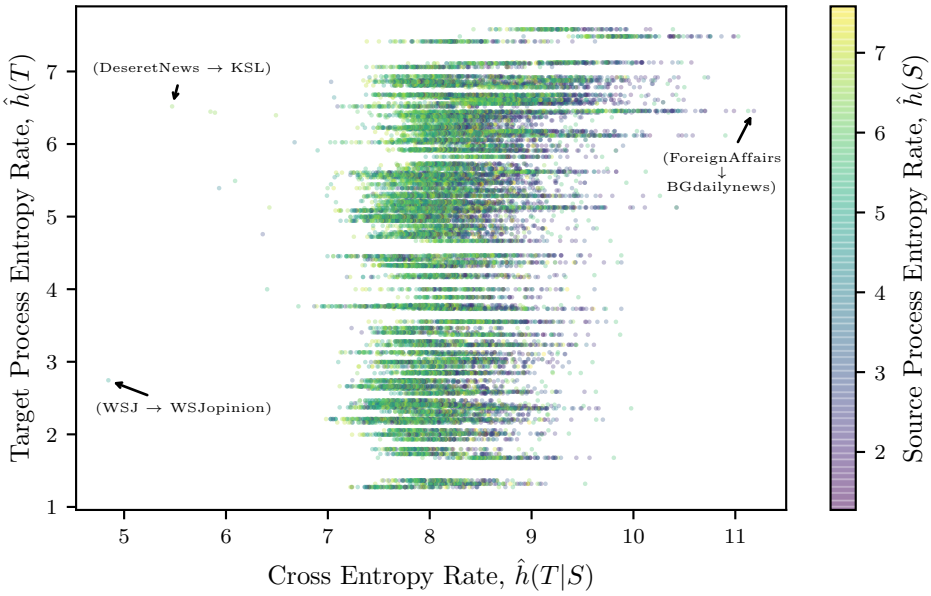


Figure 4.12: Time-synced cross entropy rate estimates on pairs of news-media organisations on Twitter using their full content for the 2019 calendar year. Cross entropy rates are compared to the entropy rate estimates of the sources, S, and targets, T, in isolation. Example outliers are shown as '(source → target)'.

Following this, the time-synced cross entropy rates are estimated and shown in Figure 4.12. Here we see similar results. The target entropy rate has limited correlation with the cross entropy rate ($R^2 = 0.116$), except for very high entropy targets, which receive higher cross entropy rates due to the high complexity nature of the target information. In contrast, source entropy rate has almost no correlation with the cross entropy rate ($R^2 = 0.0497$).

Several outliers exist at both the high and low ends of cross entropy. Low cross entropy outliers are all pairs of news-media organisations which are deeply related. Two examples of this are (*@WSJ* → *@WSJopinion*), where the information in the *Wall Street Journal's* Opinion news stream can be described extremely efficiently by the *Wall Street Journal's* main news stream,

and (*@DeseretNews* → *@KSL*), where both news-media organisations report news specifically about Salt Lake City, Utah, and *Deseret News* previously owned and remains closely linked to *KSL*.

At the high end of the cross entropy, pairs of organisations tend to have very little relationship. The highest cross entropy rate pair is (*@ForeignAffairs* → *@BGdailynews*) which suggests that very little information from the international relations focused *Foreign Affairs Magazine* is useful in describing 'Southcentral Kentucky's No. 1 Source for Information' *Bowling Green Daily News*.

In total, the cross entropy rates are much higher than the entropy rates, highlighted in Figure 4.13. This is an expected behaviour, as predicting the language and content behaviour of someone else is naturally harder than predicting your own future text. These distributional differences are carried over into maximal predictability, with generally lower maximal cross predictabilities.

Concluding remarks

Using both the Kontoyianni entropy rate estimate and the generalised Kontoyianni cross entropy rate estimate, we can construct a set of summary statistics about the language and behaviour patterns of the news sources under investigation. These estimators are well founded on both a theoretical backbone and experimental evidence, and can be applied efficiently using the new open-source package, ProcessEntropy. From here, these calculated estimates can be used to extract the measures of information flow sought in this thesis, a task for Chapter 5.

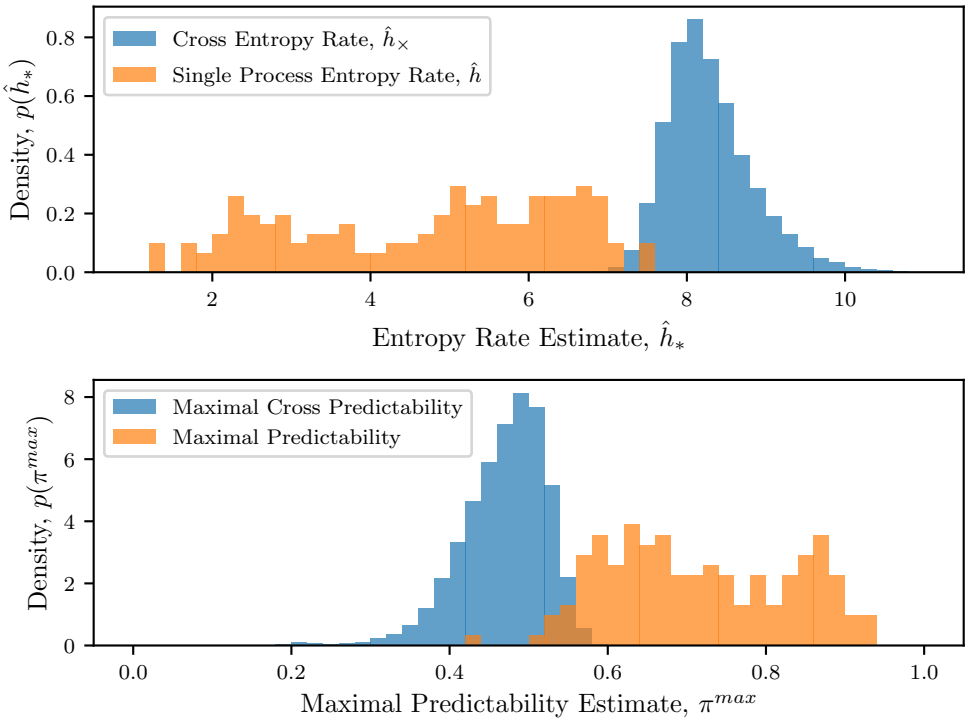


Figure 4.13: Comparison of cross and individual measures of information complexity. Cross entropy rates calculated of all pairs of 154 news-media organisation Twitter timelines is compared to the entropy rates of those text corpora in isolation. Maximal predictability of individual Twitter timelines is compared to the maximal cross predictability for these pairs of content.

Chapter 5

Creating robust information flow measures

The question underpinning this work is how information flows between news organisations. The cross entropy rate estimation introduced in Chapter 4 is an important tool to tackle this challenge, but is not sufficient in isolation. To understand information flow in the news-media ecosystem we need a tool that meets three criteria: (1) it must accurately identify how much information flows between outlets, (2) it must determine the direction of that flow, and (3) it must do so in the presence of information noise.

In this chapter we examine the efficiency of the cross entropy rate as a tool for measuring information flow, and introduce new measures derived from it. These measures are tested in a variety of simulated conditions using both synthetic and real language data to determine the best approach for quantifying information flow.

5.1 The quoter model

The 'quoter model' [5] is a simple model for capturing the dynamics of information flow on networks. This model places N individuals in a network connected by directed edges. These edges indicate that a quoting process is occurring from the source of the edge, j, to the target, i. This network of quoting across edges is designed to mimic the information generation process of users on social media, where users create content by either adding new information to the platform, or copying/quoting information already seen in their feed.

Each edge is assigned a quote probability, q_{ji}, and each node has a self generation probability, q_{ii}. These probabilities are used to decide how a

node behaves at each time step of a simulation and are normalised such that $\sum_{\forall i} q_{ij} = 1$. The model repeatedly performs a process of text generation for T steps. At each time point t, every node creates text through one of two processes.

With probability q_{ii}, the node *self generates* a new sequence of words with length $\lambda(t) \sim L(t)$. $L(t)$ can be any length distribution that is representative of natural text lengths. These words are timestamped with t, and can be generated by drawing from any distribution of words or sequences. This new text is concatenated into a single timestamped sequence for each node at every time point, starting from $t = 0$.

Alternatively, at each time point from $t = 1$ onwards the node can undergo a *quote process*. With probability q_{ji}, the target can quote from a source, j, by selecting a point in the source's sequence uniform randomly. The source's sequence includes only words generated before time t. Starting from the selected point in the source's history, a subsequence of length $\lambda(t) \sim L(t)$ is copied from the source's history into the present of the target, and is added to the target's timestamped sequence with time t.

Importantly, the quote process draws from the source's past indiscriminately, and could copy a sequence of text that the source had previous quoted from elsewhere. This sequence itself could originally generated by the target, and arrived in the source's history though a series of random quoting rounds.

In general, this results in a system of information being passed between the nodes in the network, dependent of the quoting probabilities over the edges.

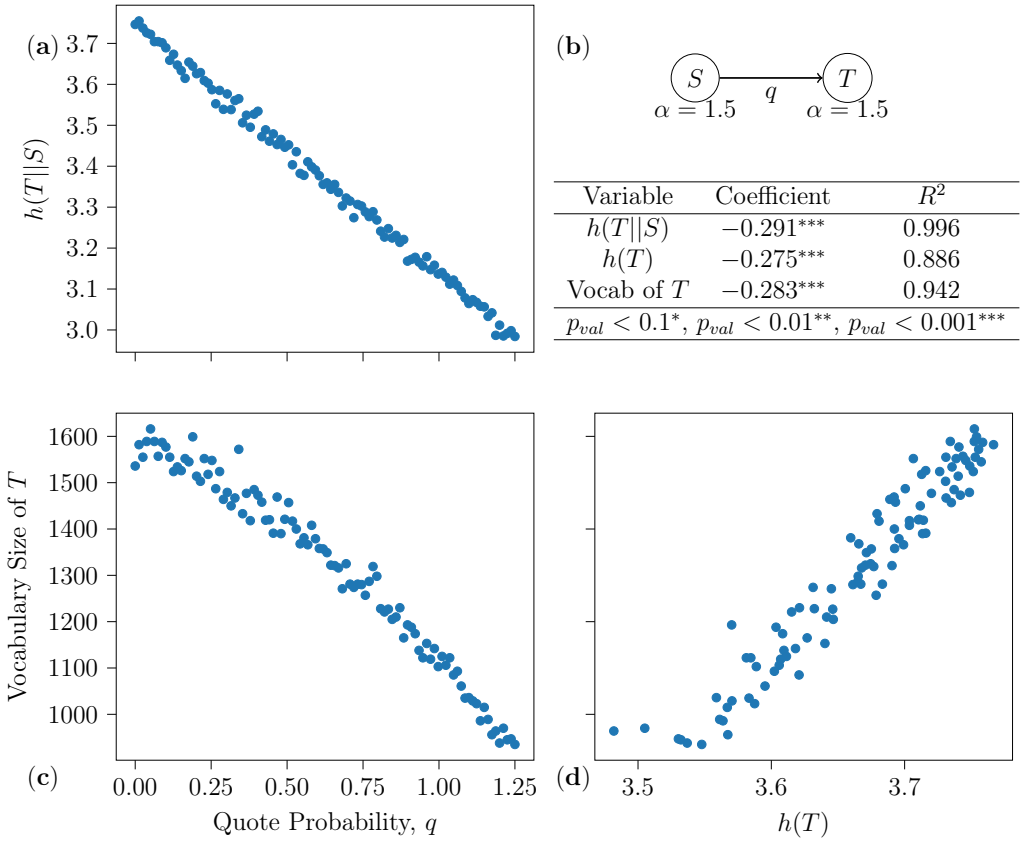
The *self generation* process can use any arbitrary method for generating a sequence given a length $\lambda(t)$. In the case of the original model, two methods are used; drawing uniformly from a fixed vocabulary and drawing from a rank ordered vocabulary according to a Zipf distribution. In line with the previous chapter we use a Zipf distribution to generate text here.

5.1.1 Single flow estimation

Previous work [5] used the Kontoyianni cross entropy rate to examine flow direction and quoting probability. This approach is reasonable, and deeply rooted in the quoter model construction. The quoter model produces sequences in the future of a quoter that exactly match the quoted source. It is these sequences which the match-length based cross entropy rate measure captures. However, this approach can break down when the producers of text have non-homogeneous text production methods.

We can see where this assumption of homogeneous generation is violated by performing an simple experiment. We take two text-producing nodes and

connect them by a single edge, as in Figure 5.1(b). The node S will always produce new text, without ever quoting. The node T will follow the simple quoter model rules, producing new text of length $\lambda \sim Poi(3)$ with probability $1-q$ or quoting a length λ from the history of the node S with probability q. This link is simulated for a range of values for q from 0 to 1. Both of these producers create their text by drawing from a Zipf distribution with scaling parameter $\alpha = 1.5$.



Variable	Coefficient	R^2
$h(T\|\|S)$	-0.291^{***}	0.996
$h(T)$	-0.275^{***}	0.886
Vocab of T	-0.283^{***}	0.942
$p_{val} < 0.1^{*}$, $p_{val} < 0.01^{**}$, $p_{val} < 0.001^{***}$		

Figure 5.1: Simulations with node S generating text from a Zipf distribution with scaling parameter $\alpha = 1.5$ and node T generating similarly with probability $1-q$ or quoting from S with probability q. While the cross entropy rate is tightly correlated with q (a), so two is the vocabulary size of T (c) and the self entropy rate of T (d). Simple linear models demonstrate that these comparison-free measures in (c) and (d) capture almost the same amount of explanatory information of q as the cross entropy.

We calculate the cross entropies between the two nodes in both directions, and other properties such as the entropy rate and vocabulary size of both S and T in isolation.

We perform separate linear regressions on the quoting probability for each of the properties, $h(T||S)$, $h(T)$ and the vocabulary. The properties and q are transformed to a standard normal, with the logarithm of the vocabulary first being taken. As expected, the cross entropy, $h(T||S)$ and q have a significant negative linear relationship ($p < 10^{-15}$), with 99.6% of the variance explained. This result seems positive, but needs to be addressed in context. The cross entropy does show the relationship between the two sources, but is also influenced by the sequence properties of T.

A fitted linear model between the self entropy rate $h(T)$ and q has a similar powerful negative linear relationship ($p < 10^{-13}$, $R^2 = 0.886$). This result is a direct consequence of an important caveat of this model, its sensitivity to the vocabulary sizes of quote distributions. Indeed, the linear relationship between the normalised logged vocabulary of T and the quote probability, q, shares the same predictive power ($p < 10^{-13}$, $R^2 = 0.942$). This suggests that much of the predictive power in the cross entropy and self entropy rate is provided by the differences in vocabulary size between source and target. Hence, we first need to examine the the distributions of vocabulary size under quoting regimes before we can move on.

Vocabulary sizes of quoted distributions

This tight relationship between vocabulary size and quoting probabilities can be explained by sampling from a distribution with replacement.

Theorem 5.1.1. Suppose we have a set of tokens S with a fixed finite vocabulary size V_S. If we take a sample with replacement of size N from S, this sample T has a random vocabulary size V_T, according to the distribution,

$$P(V_T = v) = \frac{S_2(N, v)V_S!}{V_S^N(V_S - v)!},$$

where $S_2(\dot{,}\dot{)}$ is the Stirling number of the second kind [32].

Proof. The vocabulary size is given by the number of unique elements in the sample. For a sample with vocabulary size v, each element must come from the set of size V_S. Hence, the number of possible vocabulary sets of size V_T is $\binom{V_S}{v}$.

Given a sample vocabulary set (which has size v), we must choose how the elements that we draw are distributed amongst that vocabulary set. In essence, we have a set of v bins that we can fill with our N elements in our sample. Rephrased, we are finding the number of ways to partition N elements into v labelled sets, which is given by, $v!S_2(N, v)$.

Together, this gives that the total number of possible samples with vocabulary size v is,

$$\binom{V_S}{v} v! S_2(N, v) = \frac{v! S_2(N, v) V_S!}{(V_S - v)! v!} = \frac{S_2(N, v) V_S!}{(V_S - v)!}.$$

Finally, we must divide by the total number of possible samples, V_S^N, which gives,

$$P(V_T = v) = \frac{S_2(N, v) V_S!}{V_S^N (V_S - v)!}.$$

■

From this we find that the expected value for V_T is,

$$E[V_T] = V_S \left(1 - \left(1 - \frac{1}{V_S}\right)^N\right) < V_S.$$

This highlights an important point, that when taking a finite sample with replacement, the expected vocabulary size of the sample will be *lower* than the source.

This is a useful upper bound on the expectation of our quoter vocabularies. We expect the quoter model vocabularies sizes to be smaller again, due to the finite size of the text data during quoting. At early stages of the simulations, very little text exists in the source, and hence V_S will be very small to start with. This means that under very high quote probabilities, V_S must grow before V_T can grow.

We can see this result in Figure 5.2. Two nodes are again simulated with a single flow edge between then as per Figure 5.1. When there is no quoting between the nodes ($q = 0$), meaning they both draw words identically and independently using the self generation process, the nodes have almost identical vocabulary distributions. The nodes are simulated 10,000 times for 1000 time steps of generation, giving a mean vocabulary size of 415 for both. However, when $q = 1$ the target node is exclusively quoting from the source and does not produce text of its own. As expected from the above result, we see a significantly left-shifted vocabulary distribution, with the target node having a mean vocabulary size of 240. Figure 5.2 reveals that this shift in vocabulary size change under increase q is smooth, by running 10,000 simulations with $q \sim U(0, 1)$.

Variable vocabularies

We next validate the ability of a measure to detect information flow beyond quoting. To achieve this, we introduce an extra dimension of variability to the model.

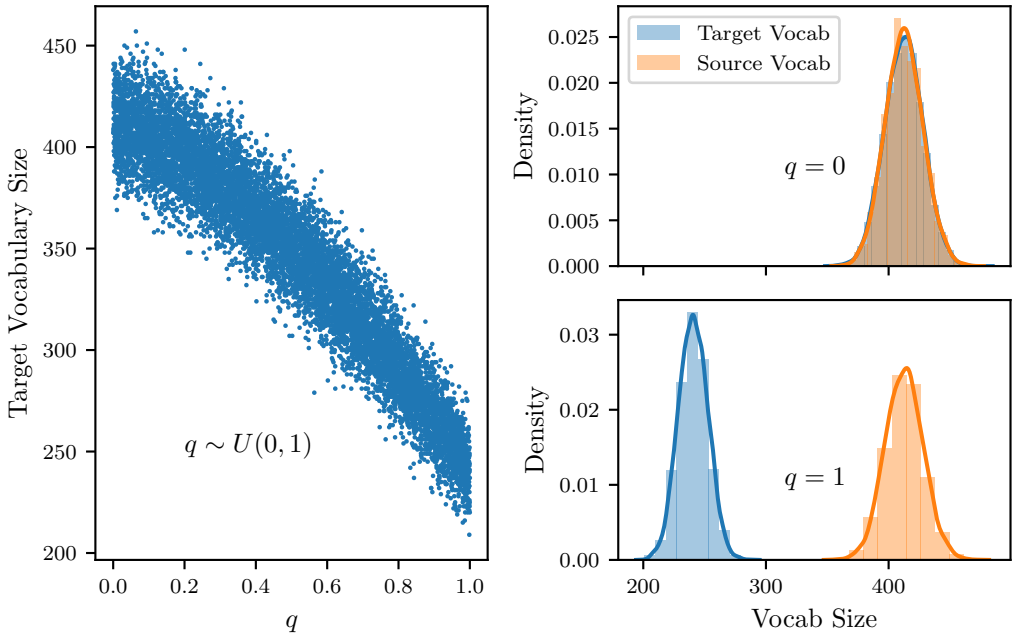


Figure 5.2: A source produces text from a Zipf distribution of words. A target produces text similarly with probability $1 - q$ or quotes from the source with probability q. The vocabulary size of the source and target are calculated by taking the number of unique elements after 1000 time steps. The vocabulary size of the target decreases as q increases.

We again have two text producers S and T, shown in Figure 5.3, where T quotes from S with probability q. The source S still creates text via a Zipf law distribution with scaling parameter $\alpha_S = 1.5$, but the target T now generates its own text at $\alpha_T \sim U(1.25, 1.75)$. Having a larger scaling parameter will result in a smaller vocabulary distribution and *vice versa*. This creates an interesting challenge for our measure. It needs to both account for the quoting along the edge, as well as natural variations in the text generation of T.

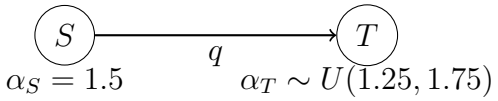


Figure 5.3: An diagram of a new quoting simulation regime. The target, T, now has a variable Zipf distribution scaling parameter for self generation, adding variability to the vocabulary size of its own self generated text.

The single flow experiment is run with these new generation protocols to vastly different results. Figure 5.4, the simulations with very low quoting probabilities exhibit high variance in the vocabulary size, which is directly attributable to the choice of α_T. As the quoting probability approaches 1, the amount of self generation of text by T reduces resulting in a vocabulary of slightly under 66% of the S vocabulary size, as expected.

Again a linear model is fitted on q using $h(T||S)$, with both variables being normalised. There is still a significant negative relationship ($p < 4 \times 10^{-6}$), but now with much less variance explained ($R^2 = 0.356$) compared the the previous experiment ($R^2 = 0.996$). In a similar fashion, both the self entropy rate and logged vocabulary size of T have far lower explanatory power ($R^2 = 0.088$ and $R^2 = 0.163$ respectively).

Outside of simplified model conditions where text producers are identical, cross entropy rates are not themselves sufficient to identify the level of information flow. This is important as the real data discussed in Chapter 3 has already been shown to have a wide range of vocabulary sizes. If only cross entropy rates were used on this data, the results would largely capture commonalities in vocabulary between pairs of organisations, and information flow measures would not be comparable.

Performing a multiple linear regression on q using a combination of variables provides a picture of how these variables relate. Table 5.1 shows models which combine $h(T||S)$ with the opposite direction cross entropy, $h(S||T)$, the self entropy rate of T, $h(T)$, and the self entropy rate of S, $h(S)$; with all variables being normalised. These models regressed on q show a significant relationship with greater predictive power, hinting that more complex

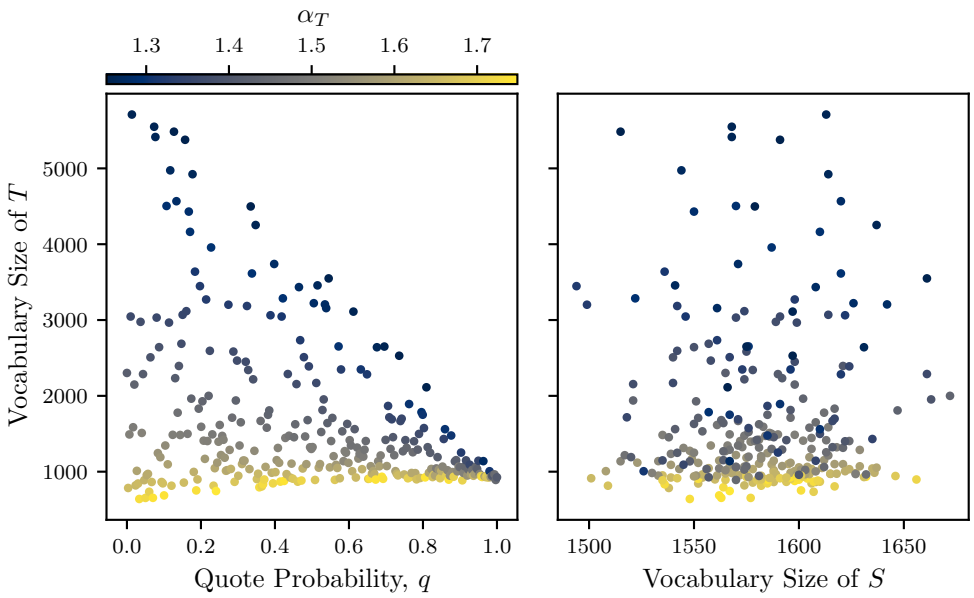


Figure 5.4: The target node T self generates with probability $1-q$ from a Zipf distribution according to a variable scaling parameter, $\alpha_T \sim U(1.25, 1.75)$. With probability q, T quotes from the source, S, which always self generates with $\alpha_S = 1.5$. The added variability in the self generation T decouples the tight previously correlation between the vocabulary size and the self generation probability.

relationships between the metrics may be required to measures information across these edges. Indeed, in an environment where sources and targets have varying text generation parameters, such as in our real data, the use of other entropic estimates may be useful in controlling for these exogenous properties of the language.

Variable	Model 1	Model 2	Model 3	Model 4
$h(S)$				-0.026^{**}
$h(T\|\|S)$	-0.444^{***}	-0.814^{***}	-0.641^{***}	-0.742^{***}
$h(T)$		0.679^{***}		0.485^{***}
$h(S\|\|T)$	0.343^{***}			0.132^{***}
Vocab(T)			0.490^{***}	
R^2	0.876	0.966	0.626	0.979
adj. R^2	0.871	0.965	0.610	0.977
$p_{val} < 0.1^{*}$, $p_{val} < 0.01^{**}$, $p_{val} < 0.001^{***}$				

Table 5.1: Four ordinary linear regressions are fit on q using the entropy rates of S & T, the cross entropy rates between S & T in both directions, and the vocabulary size of T, Vocab(T). These variables are calculated from 1000 simulations of T quoting S with probability $q \sim U(0, 1)$, or T self generating using $\alpha_T \sim U(1.25, 1.75)$ with probability $1-q$. S always self generates using $\alpha_S = 1.5$. These models suggest that a combination of variables can help predict q when T and S have different generation distributions.

5.2 Novel measures of information flow

To better capture the true flow of information in a network, a more robust measure may be needed. Here we introduce several possible measures and test them in larger simulated networks.

Our first measure is the most naive. We simply take the difference between the cross entropy rates in both directions,

$$a_{\hat{h}} = \hat{h}(T||S) - \hat{h}(S||T). \tag{5.1}$$

Remembering that the measure needs to identify both the direction and magnitude of the flow across the edge, $a_{\hat{h}} > 0$ indicates a flow from S to T, and the reverse from T to S if $a_{\hat{h}} < 0$. Further, a good metric should be able to distinguish between edges with and without flow ($q = 0$). This first measure can be seen as an 'absolute' measure.

In contrast, the second and third measures,

$$b_{\hat{h}} := \frac{\hat{h}(T||S)}{\hat{h}(T)} - \frac{\hat{h}(S||T)}{\hat{h}(S)}, \tag{5.2}$$

and

$$c_{\hat{h}} := \frac{\hat{h}(T||S)}{\hat{h}(S)} - \frac{\hat{h}(S||T)}{\hat{h}(T)}, \tag{5.3}$$

normalise the cross entropy rates by the self entropy rates of the target, in $b_{\hat{h}}$, and the source, in $c_{\hat{h}}$. In theory, these self entropy rates may help create a fair comparison between the cross entropy rates in each direction.

For example, if a source was to have an extremely large vocabulary, or otherwise have complex language leading to a high information content, the cross entropy rate would be naturally inflated regardless of the quoting probability. Normalisation by entropy rate may hence improve detection.

The final set of measures seek to solve the problem of entropy normalisation, as well as the challenges of increased complexity of quote dynamics (such as the presence of quote cycles and chains of quoting) posed by larger networks. Measures,

$$d_{\hat{h}} := \frac{\hat{h}(T||S)}{\sum_X \hat{h}(X||T)} - \frac{\hat{h}(S||T)}{\sum_X \hat{h}(X||S)}, \tag{5.4}$$

and

$$e_{\hat{h}} := \frac{\hat{h}(T||S)}{\sum_X \hat{h}(T||X)} - \frac{\hat{h}(S||T)}{\sum_X \hat{h}(T||X)}, \tag{5.5}$$

seek to normalise the cross entropy rates using local neighbourhood network information, by dividing by the average cross entropy into the source ($d_{\hat{h}}$) or target ($e_{\hat{h}}$).

This seeks to solve a deeper challenge, namely that in densely connected networks, there can be feedback loops and chains of information flow. Normalising using local network information may provide additional insight into the flow on single edges within the larger network.

So far we have discussed measures using only cross entropy rates and self entropy rates. For each metric we also consider the same measures with entropy rates $\hat{h}(T||S)$ replaced with predictabilities $\hat{\pi}(T||S)$. These predictabilities introduce a level of normalisation by the vocabulary of the target distribution. Greater flow likelihoods are represented by *higher* predictabilities and *lower* cross entropy rates; as such, the equations using $\hat{\pi}(T||S)$ are reversed in sign.

Table 5.2 shows the collection of all of the measures discussed here.

Entropy Based	Predictability Based
$a_{\hat{h}} := \hat{h}(T\|\|S) - \hat{h}(S\|\|T)$	$a_{\hat{\pi}} := \hat{\pi}(S\|\|T) - \hat{\pi}(T\|\|S)$
$b_{\hat{h}} := \frac{\hat{h}(T\|\|S)}{\hat{h}(T)} - \frac{\hat{h}(S\|\|T)}{\hat{h}(S)}$	$b_{\hat{\pi}} := \frac{\hat{\pi}(S\|\|T)}{\hat{\pi}(S)} - \frac{\hat{\pi}(T\|\|S)}{\hat{\pi}(T)}$
$c_{\hat{h}} := \frac{\hat{h}(T\|\|S)}{\hat{h}(S)} - \frac{\hat{h}(S\|\|T)}{\hat{h}(T)}$	$c_{\hat{\pi}} := \frac{\hat{\pi}(S\|\|T)}{\hat{\pi}(T)} - \frac{\hat{\pi}(T\|\|S)}{\hat{\pi}(S)}$
$d_{\hat{h}} := \frac{\hat{h}(T\|\|S)}{\sum_X \hat{h}(X\|\|T)} - \frac{\hat{h}(S\|\|T)}{\sum_X \hat{h}(X\|\|S)}$	$d_{\hat{\pi}} := \frac{\hat{\pi}(S\|\|T)}{\sum_X \hat{\pi}(X\|\|S)} - \frac{\hat{\pi}(T\|\|S)}{\sum_X \hat{\pi}(X\|\|T)}$
$e_{\hat{h}} := \frac{\hat{h}(T\|\|S)}{\sum_X \hat{h}(T\|\|X)} - \frac{\hat{h}(S\|\|T)}{\sum_X \hat{h}(T\|\|X)}$	$e_{\hat{\pi}} := \frac{\hat{\pi}(S\|\|T)}{\sum_X \hat{\pi}(T\|\|X)} - \frac{\hat{\pi}(T\|\|S)}{\sum_X \hat{\pi}(T\|\|X)}$

Table 5.2: A glossary of the measures introduced to detect information flow.

5.2.1 Network simulations

To evaluate the quality of the information flow measures, we use quoter model simulations on larger networks to examine how the measures perform under various network types.

In Figure 5.5a, cliques are generated for sizes, $N = 2$ to $N = 50$. In the clique, each possible pair of nodes, (j, i), is connected with a direction (bi-directional links are not allowed) and a preliminary edge quote probability, $q'_{ji} \sim U(0, 1)$. Each node is assigned a self generation probability, $q_{ii} = 0.5$. The incoming quote probabilities are normalised across the local incoming edges,

$$q_{ji} = q'_{ji} / \left(\sum_{k \neq i} q'_{ki} + q_{ii} \right),$$

while the self generation probability is held constant.

As above, the network undergoes a procedure of text generation wherein a node i generates a new sequence of length $\lambda \sim Poi(3)$ from a Zipf distribution with scaling parameter $\alpha = 1.5$ or quotes a λ long subsequence from neighbour j with probability q_{ji}.

This generation procedure is performed for 7000 time steps to produce a quoter model network with sequences averaging $7000 \times \mathbb{E}[\lambda] = 21,000$ tokens. As we saw in Chapter 4, this is a sufficient number of tokens to allow for cross entropy convergence while balancing speed of computation. Cross entropy rates are calculated between every pair of edges and self entropy rates are calculated for every node. We then compute the information flow measures in Table 5.2. For each network size, M, simulations were repeated

such that there were at least 500 edges being estimated (e.g. $M = 2$ requires 500 simulations while $M = 3$ requires 167), with a minimum number of 4 simulations (e.g. $M = 49$ has 1176 edges to estimate, but is repeated 4 times for a total of 4704 data points). For each M, the Pearson correlation coefficient is calculated between the true quote probability and the estimated information flow using each measure.

Figure 5.5a shows that estimating quote probabilities becomes harder for larger network sizes. Measures e and c, which both normalise by the complexity of the target, outperform all others, with the local neighbourhood network information adding a slight improvement for e. These measure perform well using both cross entropy and cross predictability, with almost identical results which obscure the overlapping lines in the figure. Measures a and d perform equally well, with the predictability measures performing worse than the cross entropy measures. The tight relationship is surprising given that a has no normalisation and d normalises by the cross entropy / cross predictability into each source. Normalising by the source information proves counter productive, with b have the lowest correlations between the true edge flows and estimated flows.

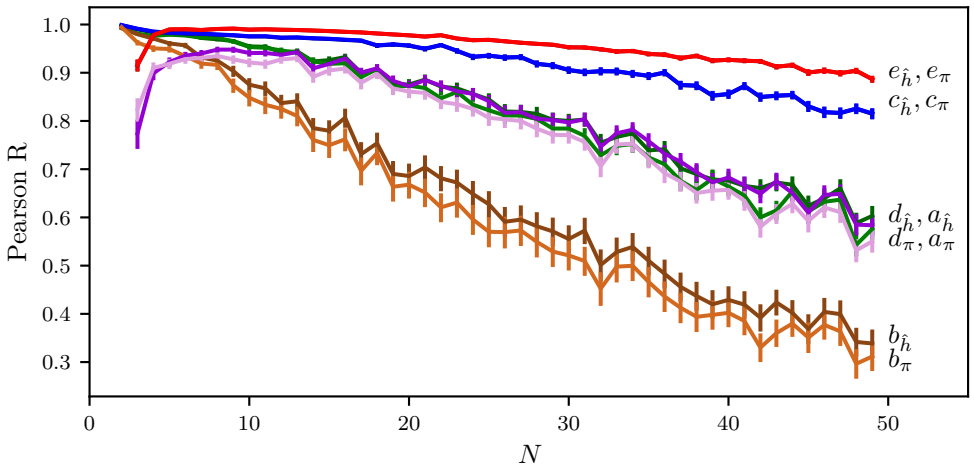
These trends are consistent as N increases, with one caveat. For small N (≤ 5) the local neighbourhood information based measures perform *worse* than their non-network counterparts.

A similar set of simulations is performed using a fixed size network ($N = 20$) and directed edges generated randomly with probability p. Each edge in this Erdős–Rényi graph is assigned a quote probability and normalised exactly as above. All nodes self generate with a fixed probability $q_{ii} = 0.5$. The general ranking of the measures remains consistent for all p in Figure 5.5b. Measure e consistently outperforms all other measures followed closely by measure c, this is likely as both measures e and c are normalising by the target complexity in both directions, with the network information giving a slight advantage to e. The rankings of measures a and d have similar Pearson correlation coefficients and remain close throughout.

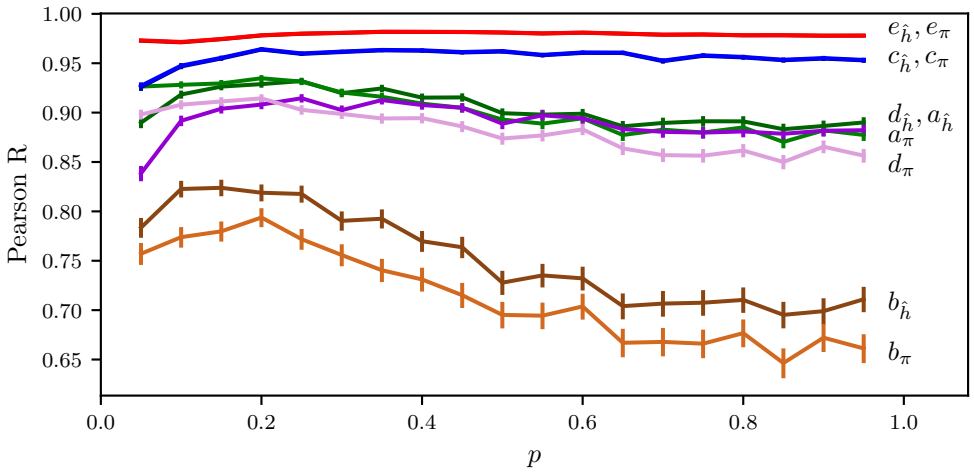
It is useful to remember that the Erdős–Rényi graph with $p = 1$ is identical to the graph generation at point $N = 20$ above. This approach will help us with the next question, disentangling the effect of edge quote probability and network complexity.

5.2.2 Disentangling complexity from quote probabilities

With increasing network sizes comes both increasing conversational complexity and lower individual quote probabilities. This conversational complexity simply comes from having more nodes quoting each other. Whenever a quot-



(a) Changing network size of random-orientation directed cliques



(b) Changing edge probability of Erdős–Rényi networks. Density increases.

Figure 5.5: Networks are generated with a simulated quoter model and 95% confidence intervals are shown for the Pearson correlation between the true quote probability and the estimated information flow for each flow measure. (a) shows simulations on directed networks with every pair of nodes having a directed single edge between them. Network size N is varied, with larger networks resulting in smaller correlations. (b) takes an Erdős–Rényi network with 20 nodes and varying edge probability p. Flow correlations are consistent across p for measures e and c despite increasing edge density while b measures see a slight decline in performance.

ing cycle appears (A quotes B; B quotes C; C quotes T) we could expect for difficulty in measuring the flow across individual edges, as the signal between the three would become mixed. As network size increases in a clique graph, more of these cycles appear.

In addition to this, the increasing network size lowers the quote probabilities across each individual edge. As a directed clique graph with a single edge between each pair of node, each node will have an average of $(N-1)/2$ neighbours to quote from. Since the self generation probability is fixed at $q_{ii} = 0.5$, the remaining neighbours must share the leftover probability between them. This results in lower quote probabilities across the edges, which provide less signal making them harder to detect amongst the noise.

Figure 5.6 shows these quote probability differences by comparing measures $a_{\hat{h}}$ and $e_{\hat{h}}$ against the edge quote probabilities (q_{ij}) from networks of size $N = 8$ and $N = 30$. The quote probability has a much larger range of [0,0.5] for the smaller $N = 8$, with a smaller range [0, 0.14] for $N = 30$.

The variance in the measures for low quote probabilities is very similar at both network sizes, however, the high quote probabilities of the small network add leverage to the Pearson correlation. In essence, we can view the underlying variance in flow estimates as a property of the language complexity. Sub-sequences of text can repeat themselves independently in different generators when both draw from the same or similar distributions. This noise is a property of the self generation process, not the quoting process. Given the fixed self generation probability, this aspect of variance is constant regardless of the quote probabilities. Indeed, when a no-intercept linear regression is performed for each N, the residuals exhibit homoscedasticity across the range of true quote probabilities and the standard deviation of the residuals is similar across all N (95% CI of $\sigma_{residuals}$: [0.186, 0.228]).

Having high individual edge quote probabilities allows for the signal of those flows to be picked up through the noise, and leads to a stronger tail of high flow and high quote probability points. This naturally increases the Pearson correlation coefficient.

A common solution to high leverage points is to use a Spearman correlation coefficient, which uses ranks rather than values. However in this problem the Spearman correlation provides almost identical values to the Pearson. The higher value points are not outliers, but rather the tail end of a well-spread distribution across the [0,0.5] range, and low rank words still experience the same signal-noise trade-off as when comparing Pearson correlations for different distributions of edge quote probabilities.

To further verify this effect we can vary the previous fixed self generation probability. As before, the edge quote probabilities normalise such that all generation and quote probabilities sum to one. Increasing q_{ii} thus has the

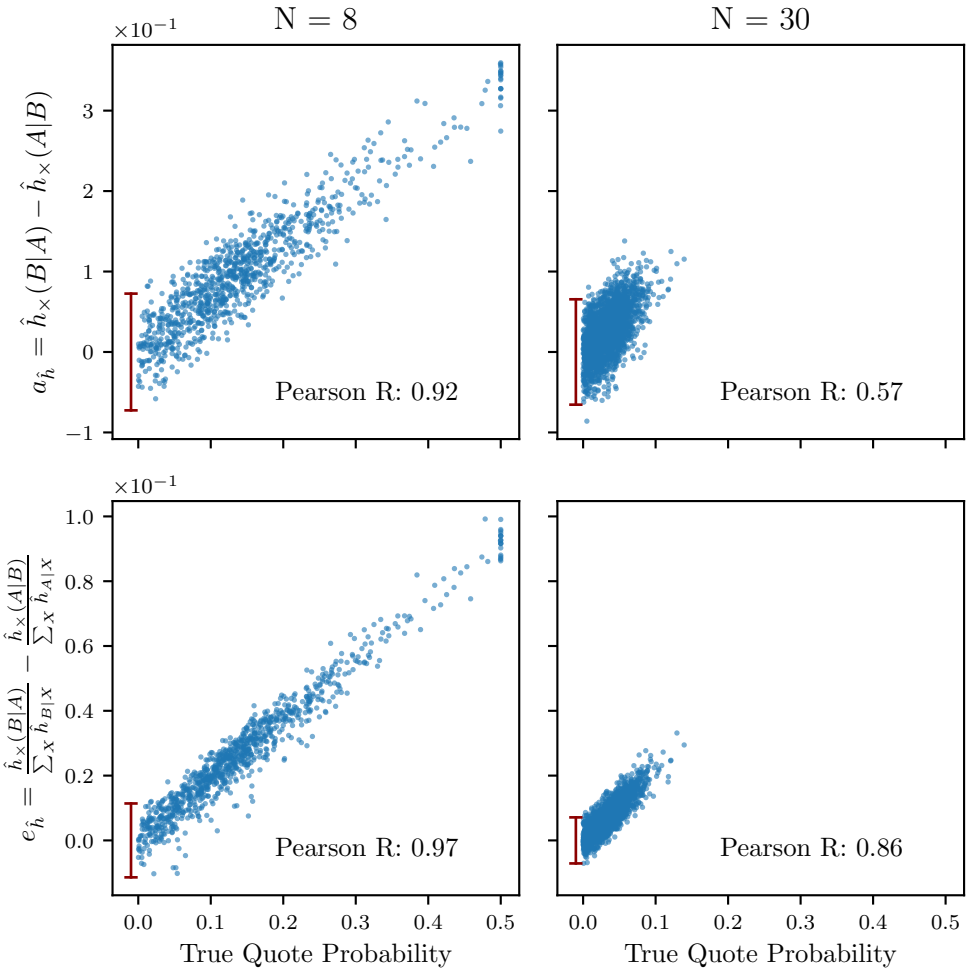


Figure 5.6: Examples relationships between the true quote probability on a link between two notes and an information flow metric. Simple difference metrics such as $a_{\hat{h}}$ perform well on smaller networks with few nodes (N=3), but larger networks (N=30) need local neighbourhood information to perform well. For small networks, the tail high end quote probabilities pull up the Pearson correlation as they have stronger signal to overcome the constant variance from natural language generation. The dark red bar shows the width of the 99% confidence interval for the residuals of a no-intercept linear regression.

effect of shrinking these edge quote probabilities q_{ji}. If the above logic holds, then we would expect the increasing q_{ii} values to decrease the Pearson correlation between q_{ji} and the estimated flow. Indeed, in Figure 5.7 this pattern is observed with a clear decline in performance for all measures as the self generation probability increases.

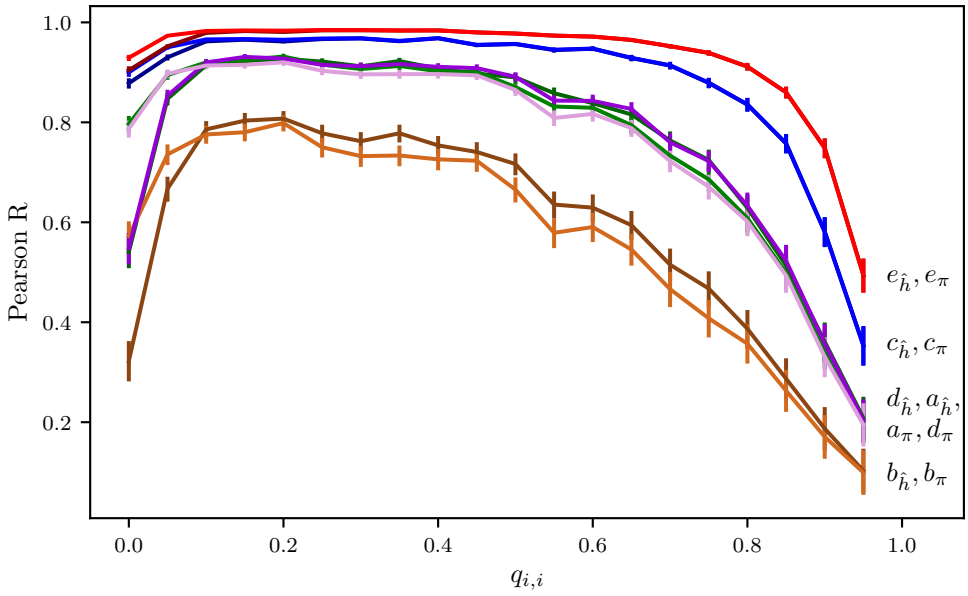


Figure 5.7: Networks with 20 nodes are generated with each edge being assigned a direction and an edge quote probability q'_{ji}. The edge quote probabilities are added to the fixed self generation probability q_{ii} and normalised. Increasing q_{ii} decreases the edge quote probabilities on average which in turn decreases the Pearson correlation between the edge quote probabilities and the measured information flow. When no self generation is present ($q_{ii} = 0$) only the generation seeds from $t = 0$ are propagated, reducing measure performance.

Notably, there is also a decline in correlation at $q_{ii} = 0$. In such a scenario the model would have all nodes quoting from one another based solely on the initial self generation step that seeds the process. This is surprising that even in an environment where limited source material exists, the measures still perform reasonably robustly.

5.2.3 The effect of rewiring on measure performance

In Subsection 5.2.1, we observed that increasing network size reduced the performance of information flow measures. Subsection 5.2.2 showed that this is, in part, due to decreasing individual edge quote probabilities. However, increasing network size also increases the number of edges into and out-off each node. This leads to more cycles within the graph (even if they are of lower weights), making it difficult to determine sources of information.

To isolate the effect of the network complexity on measure performance, we introduce a new simulated network. The Watts–Strogatz random graph model [88] is used with a rewriting parameter β. In the model a graph is generated as a lattice, where each node in a network of size 20 is connected to its 8 nearest neighbours (without rewiring this is similar to an ER(20, 0.4) graph). These edges are assigned a random direction and weight, similar to the network models above. With probability β, the endpoint of each edge will rewire to attach to a random node. As β increases, the network becomes less structured and more chaotic, with the clustering coefficient decreasing from its initial value of 0.64. In Figure 5.8 these networks are generated for values of β and the correlation coefficients are calculated for each measure. No trend is observed between β and the correlation, indicating that the structural complexity of the network has no effect on measure performance when the distribution of individual quote probabilities are fixed.

This leads to the conclusion: the ability of the measures to correctly identify information flow is mostly dependent on the strength of the underlying quote probabilities to overcome the inherent noise present in text. While this conclusion is clear from the above results, they have used a simulated language model using Zipf distributions rather than real natural language. In order to confirm these results real data must be incorporated.

5.3 Incorporating properties of real data

A key limitation of the quoter model is the lack of foundation in real text data. Word generation processes such as those based on Zipf law are indeed derived from the analysis real data, but fall short of generating text that appears natural in most senses. The words generated using this method are independent, and lack the coherent structure of English grammar.

To rectify this, we update the *self generation* process to use real text data. Using the corpus of Twitter data for news sources, each node in a simulated quoter graph is assigned a single news source. At each time step of the simulation a node can self generate by drawing a single random tweet

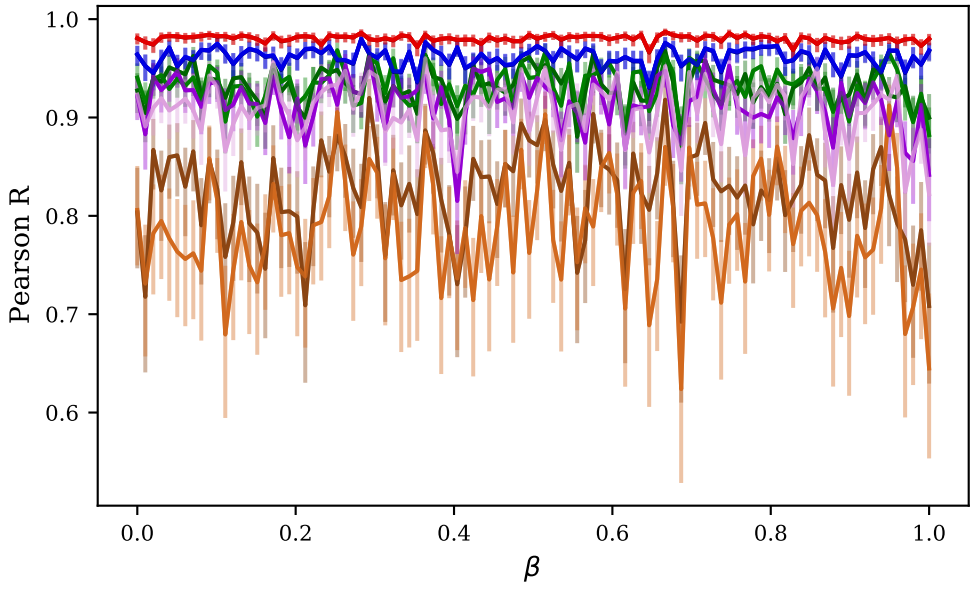


Figure 5.8: A Watts–Strogatz random graph is generated with $N = 20$ nodes and each edge assigned a direction and quote probability. These edges rewire with probability β. Information flow measures perform at consistent levels for all values of β, indicating that changing network structure has no impact on measure performance. High variance in correlation is seen for measure b (brown) with medium variance in d (purple) and a (green). This variance is exemplified by the large confidence intervals on the Pearson correlation coefficient which stands in contrast to the tight confidence intervals and low variance of measures e (red) and c (blue).

from the entire history of the chosen outlets tweets and appending it to the node's timestamped sequence. The quoting process is identical, with each quote now drawing on sequences made from real tweets. Each simulation is run for 8000 time steps to maintain a consistent average sequence length.

Since the average number of tweets for each news outlets is 19,337, drawing uniformly randomly with replacement would result in 33.9% of tweets being drawn multiple times. To quickly prove this, assume we can draw a single tweet from m tweets with probability $\frac{1}{m}$. The distribution of the number of times we draw that tweet over N draws is $Binomial(N, \frac{1}{m})$, giving the cumulative probability that we draw it greater than one time being $1 - \binom{N}{1}\frac{1}{m}(1-\frac{1}{m})^{N-1} - \binom{N}{0}(1-\frac{1}{m})^{N} = 1 - Np(1-\frac{1}{m})^{N-1} - (1-\frac{1}{m})^{N}$. With $m = 19337$ and $N = 8000$ this gives 0.338783. To rectify this, tweets are drawn without replacement.

Using this new self generation process the same experiments are run as above. Figure 5.9 combines the results from Figure 5.5, Figure 5.7 and Figure 5.8, updated to use real data. The core results outlined above remain exactly the same when using real data in place of simulated text in the model.

5.4 Go with the flow: applying the information flow measure

Selecting the best measure

Put together, the results in this chapter suggest two key findings; the cross entropy is not a sufficient measure of information flow without normalising by the target entropy properties; and the information contained in the local neighbourhood structure is more informative than the entropy rate of a target alone.

Throughout every comparison done here the ranking of the measures fall into four categories.

- Measures $e_{\hat{h}}$ and $e_{\hat{\pi}}$ consistently outperform all other measures for networks of size greater than 5. These measures work by normalising the cross entropy or cross predictability using the local neighbourhood cross entropies or cross predictabilities going into the *target*. While both types of the measure perform equally well in the large networks, the measure under-performs for networks with less than 5 nodes, as expected given the limited neighbourhood information.

- Measures $c_{\hat{h}}$ and $c_{\hat{\pi}}$ perform slightly worse than the e measures in large networks, but still hold their own. These measures work by normalising

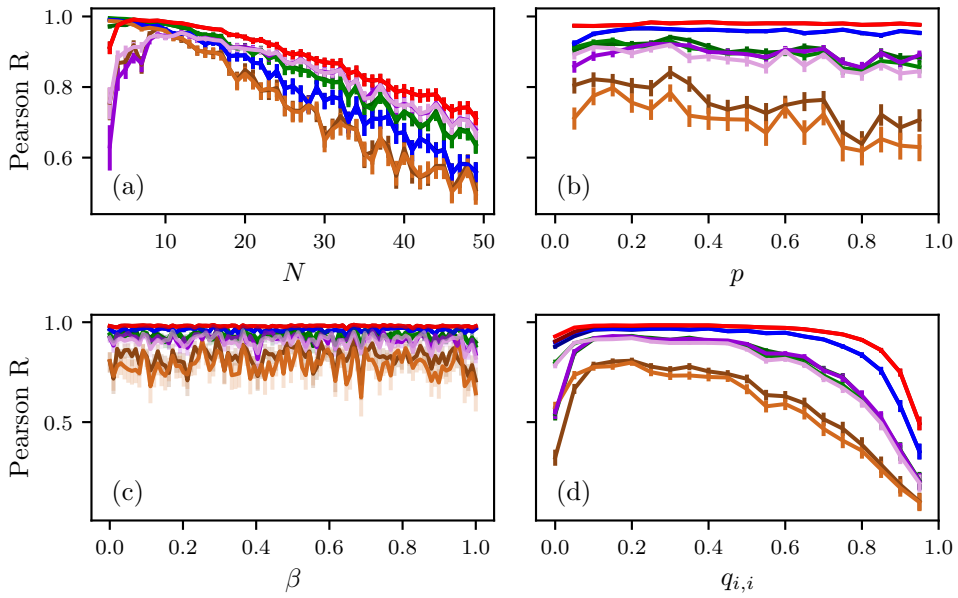


Figure 5.9: Networks are generated with N nodes where each pair of nodes has an directed edge that exists with probability p and is assigned a quote probability q'_{ji}. These quote probabilities are normalised such that their sum added to the fixed self generation probability, q_{ii} equals 1. This self generation process draws from the Twitter history of a news outlet which is assign at network creation. (b), (c) and (d) use $N = 20$ while (a) varies N. (a), (b) and (c) use $q_{ii} = 0.5$ while (d) varies q_{ii}. (a) and (d) use $p = 1$ while (b) varies p and (c) uses a more sophisticated Watt-Strogatz model which starts as a lattice with $p = 0.4$ and requires edge endpoints with probability β. All simulations using real data for text generation follow the same results as their counterpart experiments using Zipf distributions for text generation.

by the entropy or predictability of the *target* without any knowledge of the network structure. Indeed, the lack of required network information allows this measure to perform the best for very small networks.

- Measures $a_{\hat{h}}$, $a_{\hat{\pi}}$, $d_{\hat{h}}$ and $d_{\hat{\pi}}$ perform consistently poorly when quote probabilities become low. The similarly between these measures is somewhat surprising given that a has no normalisation while d normalises by the local neighbourhood of the *source*; indicating that the source neighbourhood provides almost no value to the calculation of flow.

- Indeed, measures $b_{\hat{h}}$ and $b_{\hat{\pi}}$ which normalise by the entropy rate or predictability of the source without network information perform even worse than the simple difference between un-normalised cross entropies.

From these results we find that $e_{\hat{h}}$ and $e_{\hat{\pi}}$ are clear the best measures. While either measure would be suitable in synthetic conditions, we choose to use $e_{\hat{\pi}}$ as we apply the measure to real data. In the simulations throughout this chapter we have seen the impact of vocabulary size on the measurement of information flow. As the real variety of text produces variety in vocabulary by an order of magnitude, we use the predictability version to help further counteract the impact of the vocabulary.

Hence, we move forward using the measure,

$$e_{\hat{\pi}} := \frac{\hat{\pi}(S||T)}{\sum_X \hat{\pi}(T||X)} - \frac{\hat{\pi}(T||S)}{\sum_X \hat{\pi}(T||X)} \tag{5.6}$$

as the most robust tool to estimate information flows in networks of natural language text.

Results of applying the best measure

Using measure $e_{\hat{\pi}}$ we can perform estimates of information flow between each pair of news organisations. In Subsection 4.3.4 we performed the estimation of the entropy rates and cross entropy rates between all pairs of organisations using the entire corpus of tweets by each organisation for the 2019 calender year. Once these estimates are calculated, applying the information flow measure is computationally simple final step to extracting the estimates of flow. Figure 5.10 shows the distribution of these estimates for all pairs of organisations. As $e_{\hat{\pi}}$ is symmetrical around 0 when you swap the source and target, we choose the positive flow as the direction of the edge and assign its weight to be the flow estimate.

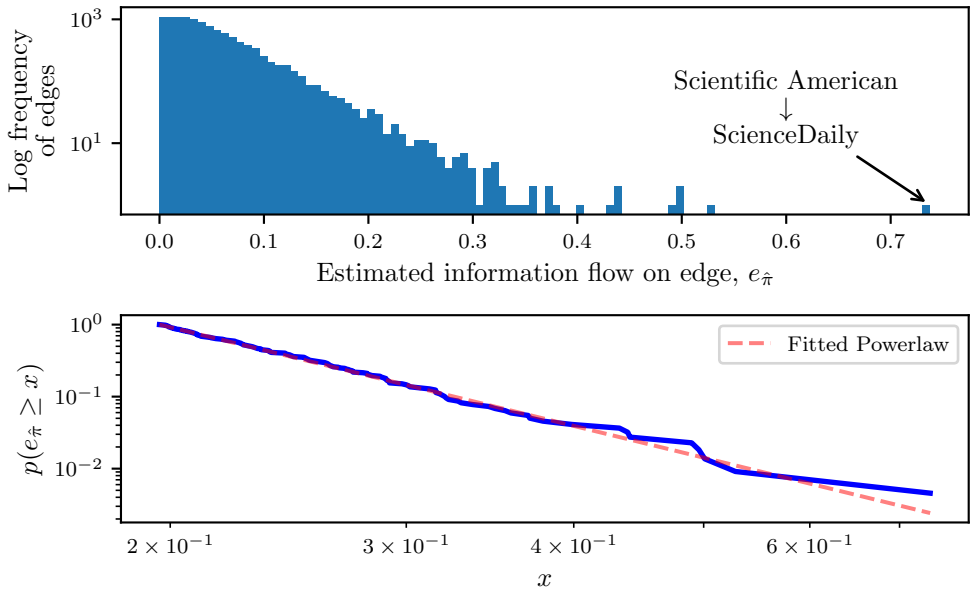


Figure 5.10: [[TS: This is your CCDF Matt. I'm not sure how much information this actually adds. Should we keep it?]] Information flow is estimated using measure $e_{\hat{\pi}}$ between each pair of news-media organisations using their entire tweet corpus. Each edge is assigned a direction and the positive weights of those edges are shown. Most edges show very little information flows with only 1.6% of edges having a flow estimate above 0.2. A complementary cumulative distribution function is shown with a fitted power law with exponent 5.6.

It is important to note that these flow estimates are not directly measuring a proportion of tweets that are copied, but rather a relative measure of how much informational content of the source is present in the target, compared to other pairs of organisations.

Two key features of Figure 5.10 stand out: the heavy right skew of the distribution and the few outliers. Of the edges weight, 98.4% are less that 0.2 leaving only 199 edges with weight greater than 0.2. This is important to note as Subsection 5.2.2 highlighted that flow estimates are harder to disentangle from the noise of natural language when the information flow is low. Much of the flow estimate edges may be detecting a weak flow purely due to the commonality of their natural language.

Noting the challenge created by information noise, these flow edges are useful for two purposes: comparing the magnitude of flows between two different pairs, and identifying the direction of flow when the flow estimate is large enough. As an example of the first, we can find that, as an information source, the *New York Times* has relatively little net information flow to the *Washington Post* (0.0053) but relatively high net information flow to the *Foreign Affairs Magazine* (0.1480).

As an example of the second purpose, we examine the highest flow edge in the network, from the *Scientific American* to *ScienceDaily*. This edge with weight (0.7378) is likely due to a confounding of factors. Firstly, *Scientific American* is a popular and well resourced organisation that produces a large volume of content. The smaller and less popular *ScienceDaily* is unlikely to get the 'scoop' on many scientific stories meaning most content it posts will exist in the history of *Scientific American*. Finally, *ScienceDaily* has the least tweets, tokens and second least vocabulary of any organisation. This makes the normalisation of the information flows difficult due to its vocabulary being at the extreme of the distribution.

In constructing these weighted edges into a network, it would be ideal to filter out low flow edges to make the graph more sparse for analysis. However, choosing such cut-offs can prove dubious at best. In the next chapter we explore several methods of using this network to rank the net information influence of news-media organisations and examine the inconsistency in their results.

Chapter 6

Influence detection & ranking stability

> Everything should be made as simple as possible, but not simpler.
>
> Albert Einstein, *commonly paraphrased quote from his Herbert Spencer Lecture*, Oxford, June 10, 1933.

Complex networks are powerful tools to encapsulate and probe deeper aspects of systems; however, this can often come at the cost of human interpretability. Answering questions is a important task for such networks, but simplifying networks to answer questions can be fraught with challenges.

In this chapter we will examine the use of the network created in Chapter 5 to explore the process of answering a specific question: which news-media organisations are the most important information sources? In essence, this question is one of ranking the news-media organisations according to their importance to the information flow network.

While ranking is often assumed to be a simple task, we will show in this chapter that different approaches to ranking a influence network can result in vastly different answers. Such approaches are built upon assumptions of importance that are often inconsistent with one another. Even standalone these approaches can exhibit unexpected sensitivities to changes in the network.

This chapter will not produce a definitive ranking of importance, but rather will use the question to explore the difficulty in constructing such an answer from a complex network. Doing so reveals the interconnected nature of the information flow ecosystem and its resistance to simplification.

6.1 Spotify: a motivating example

Recent work by South *et al.* [80] has looked into the stability of eigenvector centrality in the network of musical artist collaborations using data from Spotify. This work collected the discography, genre, popularity and other metadata for over 1.25 million artists using the online music streaming platform. From the discography, a network of artists was created with edges indicating that two artists appear on an piece of music together.

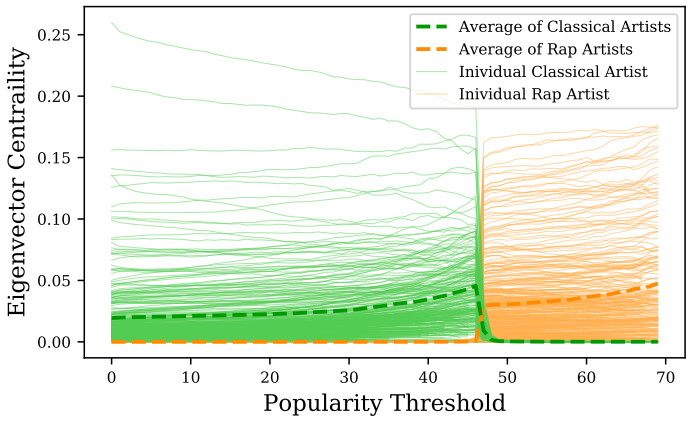
Such a collaboration network leads to a natural question; who are the most important musical artists? Eigenvector centrality (discussed further below) is applied to this large undirected network to create a ranking of how important an artist is the full network. In this calculation the popular classical artists are ranked highest in centrality. The most popular artist being Mozart followed closely by Bach, Beethoven and Schubert. This result, interesting in its own right, is not what troubles us.

Popularity in Spotify is a value between 0 and 100 based on relative music streams, which is a strongly skewed distribution resulting in a large number of low popularity artists. An experiment was run where artists with a popularity less than some threshold number were removed from the network. Such a hypothetical situation is not uncommon in practice, where challenges in data collection or limits of computation on large graphs can naturally lead to decisions to discard data below certain thresholds of relevancy.

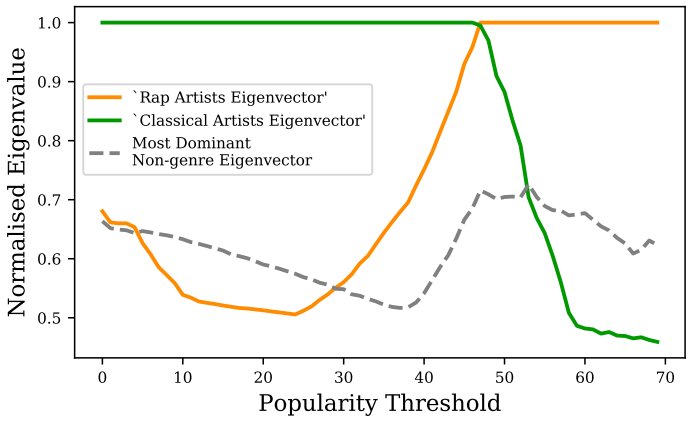
When centralities are re-calculated on networks with artists below a popularity threshold removed, the results change. For thresholds between 1 and 46, small perturbations in the rankings occur but general trends persist. However, at a threshold of 47 the centrality rankings change *en masse* to a central core of popular rap artists (Rick Ross, Lil Wayne, T.I and others). These new rankings remain consistent as thresholding increases.

This critical transition in centrality is shown in Figure 6.1a. The transition is not caused by the removal of any key artists in the graph but rather by a swapping of dominant and secondary eigenvectors as shown in Figure 6.1b. This is further validated using a novel network model by the authors and has been observed in the context of the HITS centrality [59].

The purpose of this example is to highlight a key issue. Centralities and rankings can be susceptible to critical changes even from small perturbations in the data. This certainly does not make for a good ranking, but how does one measure a ‘good’ ranking?



(a) Changing Centralities



(b) Changing Eigenvalues

Figure 6.1: Change in centrality of all classical artists and all rap artists in the Spotify artist collaboration graph as a popularity threshold is applied. Artists of other genres have negligible centrality. In (a), the critical transition in centrality between the two groups can be seen at a threshold of 46. In (b) the changes in the most dominant eigenvalues of the adjacency matrix are shown as popularity thresholding is applied to the network. Eigenvalues are normalised to the largest eigenvalue and are labelled according the group of nodes with high centrality in the corresponding eigenvectors. A swap between the dominant eigenvectors can be seen, corresponding to the critical transition in centrality.

6.2 Rank stability

There are many criteria which one can judge a ranking system: a notion of 'truthfulness'; a desire for resistance to being 'gamed'; or a desire for consistency of the ranking when small changes occur. Though not an exhaustive list, these are fair criteria, but are rife with issues. In this chapter we use these building blocks to construct some more robust metrics.

Every ranking method is truthful to its own definition, but generalising this to a Platonic ideal is impossible. Instead we could seek a unanimity between rankings. As we will see below, there are usually a large number of possible ranking approaches to any given problem. A good set of ranking methods should produce similar rankings when drawing upon similar definitions and assumptions.

Creating resistance to undue manipulation is a difficult problem, not only because defining 'undue' is problematic. Any system of ranking must be manipulated by the results of comparisons between elements. The issues arise when a single element, player, or node can alter the rankings dramatically though only its behaviour. In our context of ranking news information influence we will use a simple definition. We want to minimise the influence that a single addition of removal of a node from the network has on the rankings.

This definition sits close to our third criterion, consistency of ranking. When a small change in the graph occurs, such as the removal of a single node, we would want the ranking change to be correspondingly small. This desire for rank stability stands in contrast to the motivating example above, where the centrality ranking underwent a total rearrangement during a critical transition. This is something a strong ranking should avoid. Further to this, consistency of ranking should be maintained across rankings, especially when these ranking methods are similar in nature.

6.2.1 Ranking methods

Before proceeding further, we first define some methods of ranking the news-media outlets in our network. A vast array of ranking methods exist, but not all are suited to any one problem. In our problem, we want to rank influence in a weighted directed network. Here we will use four methods for comparison: two network centralities, a method of sport team ranking, and a network topology approach. We will also outline why many alternative metrics are not useful in this context. This list of methods is not exhaustive, but provides a substantive example of several approaches which will allow for an analysis of the viability of ranking influence in an information flow network.

Network centralities

Network centrality metrics are a common tool for measuring different notions of importance within a network. However, these notions of importance are often context specific. For example, betweenness centrality is a powerful metric that measures the number of shortest paths that pass through a node. This centrality is very useful in a context such a packet routing and load balancing, where a comparatively high value in the centrality (and thus a low rank) would indicate a high burden of traffic and the importance of reliability of that node. In the context of our question, this centrality doesn't illuminate who *contributes* the information, but rather what nodes have the most *pass-through* of information. In a similar vein, centralities such as degree and closeness also provide answers to questions we are not asking here.

A centrality measure that has characteristics appropriate to our context is *eigenvector centrality*. Eigenvector centrality, sometimes referred to as *eigencentrality*, is defined recursively in terms of the centrality of a node's neighbourhood. This stems from the notion that a node is important if it is connected to other important nodes. In our context of information flow we can restate this as: a node is an important contributor of information if it contributes to other important information contributors. In essence, if a node contributes to a few unimportant nodes, it itself is unimportant; but those nodes that contribute to influential sources, are themselves influential.

This importance rating is encapsulated in a vector $v^{(\text{eig})}$ which is defined through the recursive equation

$$\mathbf{v}_i^{(\text{eig})} = \frac{1}{\lambda} \sum_{k=1}^{N} A_{k,i} \mathbf{v}_k^{(\text{eig})}, \tag{6.1}$$

with a constant $\lambda \neq 0$ and where $A_{k,i}$ is the element of the weighted adjacent matrix of the graph, A, with N total nodes. This can then be expressed in matrix form as,

$$\lambda v^{(\text{eig})} = A v^{(\text{eig})}, \tag{6.2}$$

which can be solved as the dominant left-hand eigenvector of the adjacency matrix A.

Importantly, both here and below, the direction of 'flow' in the graph (the direction of each edge) is reversed before taking this calculation. Eigenvector centrality works such that the node *being pointed to* is assigned the importance. Since we are interested in what originates information, we must point *towards* the information source.

Eigenvector centrality has been shown to be more robust to conditions of imperfect data [18] and network manipulation [60] than other centrality

measures, but can undergo critical transitions when the eigengap is very small [80].

Similar to eigenvector centrality, *PageRank* [10, 63] is a based on random walks. PageRank extends eigenvector centrality by normalising the adjacency matrix such that the elements in the matrix represent transition probabilities between nodes on a random walk. PageRank also introduces a damping factor d, which allows walks on disconnection graphs to become ergodic and dampens extremes in centrality.

Mathematically PageRank centrality is expressed as,

$$v_i^{(\text{PageRank})} = \frac{1-d}{N} + d\sum_{k=1}^{N} \frac{A_{k,i} v_k^{(\text{PageRank})}}{\sum_i A_{k,i}}, \tag{6.3}$$

Traditional formulations of PageRank use unweighted directed graphs, such as the hyper-link web graph, but it naturally extends to weighted graphs.

PageRank and Eigenvector centrality make for an interesting comparison. They are very similar metrics and hence we should expect their rankings to be similar.

There are other centrality measures that may be relevant here, such as the HITS algorithm for finding hubs and authorities, however we choose only these two a popular examples of network centralities as they illustrate the interesting phenomena.

Game result rankings

Moving beyond graph theory, there is a rich literature of ranking in the sporting world [49]. A core feature of this world is the notion of a match, where two opposing sides face off, resulting in some score difference or binary outcome. In most of these situations, matches can be seen as an edge between two competing nodes, with the outcome determining the direction. As such several sports ranking algorithms can be directly applied to a network setting.

For our purposes, each edge represents a "information competition" between two news-outlet to see who produces the most novel information into the ecosystem. Since we are viewing all these 'matches' at the end of the season (the year of data collection), we rule out ranking systems that evolve over time according to the relative rankings at the time of play (such as the Elo system [25]). We choose a simple but relatively institutive ranking system as a means of representing these game based ranking methods.

Massey's rating method [51, 49] draws on a simple idealised equation,

$$r_i - r_j = y_k, \tag{6.4}$$

which states when team i with rating r_i and team j with rating r_j face off, the the margin of victory, y_k should be equal to their difference in ratings.

In our case, there is a single match between each pair of nodes to produce a vector $\mathbf{y}$ of length $n(n-1)$ and a corresponding unknown rating vector $\mathbf{r}$ of length n. To complete this equation we create a $m \times n$ matrix X which has two indicator variables in each row to denote two nodes played in each match.

This system is best solved as a least squares problem, $X^T X\mathbf{r} = X^T\mathbf{y}$, however $X^T X$ is not invertible as the columns are linearly dependent. In Massey's method, the final row of the matrix $X^T X$ is replaced with ones, and the corresponded $X^T\mathbf{y}$ element is replace by a zero. This additional constraint ensures the ratings sum to one and forces the least squares problem to have a unique solution.

Topological sorting

Finally, we present a method of using a directed acyclic graph to produce the rankings. Consider again our question of examining which outlets produce the most novel information. In an idealised sense, we would hypothesise a set of super-producer nodes which produce a large amount of information which then trickles down the information flow graph in which each node absorbs information and re-synthesises it with new discussions, which is henceforth passed down to other nodes. This mental picture as described has a mathematical representation, a directed acyclic graph (DAG). Formally defined, a DAG is a graph that has no cycles, meaning that no sequence of edge hops will ever loop back on itself.

Our densely connected information flow network is not itself a DAG as many cycles of flow exists. To create a DAG, a greedy approach can be taken where the lowest flow edges are removed sequentially until a DAG is achieved. This greedy approach removes 91.1% of the edges, and although more optimal DAG creation approaches exist, this method is used for simplicity.

Using this DAG a topological sorting can be created. A topological sorting is one in which node x will be higher ranked than y if and only if no directed edge or series of connected directed edges exist such that one could move from y to x. Put more simply, if an edge in the DAG goes from an outlet a to outlet b, then a must be higher ranked than b in the sorting.

This naive topological sorting has one major flaw in that it is not necessary unique[1]. Consider the diamond problem as shown in Figure 6.2, where a

[1]Naive topological sorts *can* be unique if the DAG itself is a Hamiltonian path. This would require the constructed DAG to be a single long chain of flows, which our data most definitely does not produce.

single node a connects down to two nodes b and c which themselves individually connect down to node d. In such a DAG, b and c are interchangeable in the ranking (i.e. orders (a, b, c, d) and (a, c, b, d) are both correct rankings).

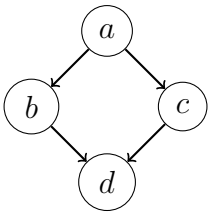


Figure 6.2: A demonstration of the diamond problem which can give non-unique solutions to naive topological sorts on directed acyclic graphs.

To circumvent this challenge, a simple heuristic is imposed on the sorting. In cases where two nodes are interchangeable, the node with the highest outgoing edge weight sum is placed first. This draws from our context of the problem and place nodes with a greater net influence higher in such circumstances.

6.2.2 Kendall rank correlation coefficient

In order to quantify a notion of rank stability, we can run sensitivity tests on a ranking system. We do this using a simple ranking experiment. We take our network and rank it using a chosen system. We then remove a single node from the network and rerun the ranking.

We compare these rankings using the Kendall rank correlation coefficient, referred to as Kendall's τ throughout. Kendall's τ measures the ordinal association between the two rankings with

$$\tau = \frac{(\text{ number of concordant pairs }) - (\text{ number of discordant pairs })}{\binom{n}{2}}, \tag{6.5}$$

where a pair of elements (x, y) are said to be concordant if the relative ordering of those two elements in both rankings is the same. The pair are discordant if x ranks *higher* than y in one ranking and *lower* in another. If two rankings are perfectly matched τ is 1 and τ is -1 if a ranking is compared with its reverse.

We run the experiment, removing each node once before putting it back, to test the rank stability of the measures introduced above. When ranking is rerun on the network with a single node removed, there can be a significant change in the ordering. While many node removals result in very little

ranking change in Figure 6.3, some nodes have a large effect with the new ranking appearing close to a random reshuffle of the ordering (τ close to 0). When we compare two rankings with n and $n-1$ nodes from a removal, we ignore the removed node from the original ranking order.

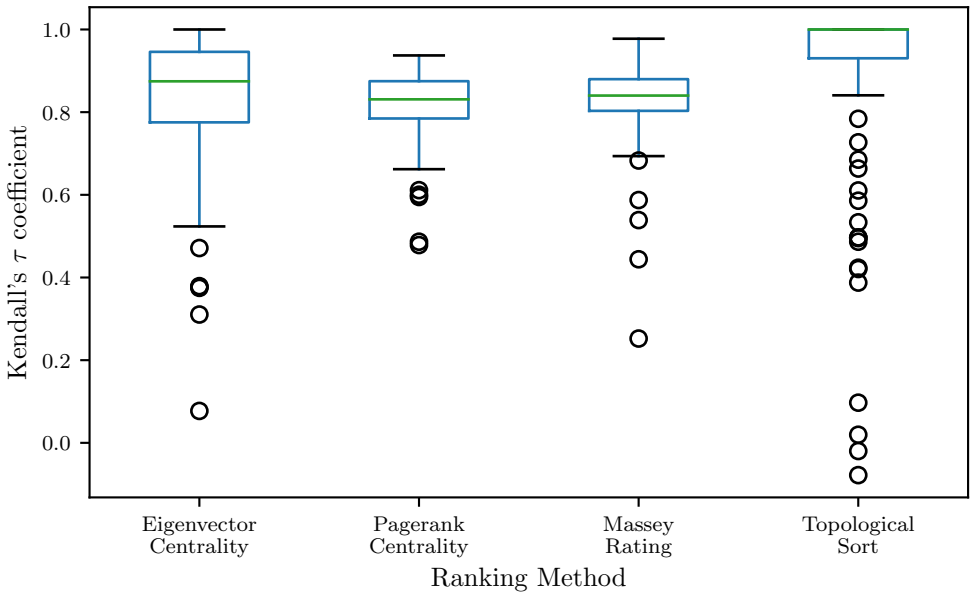


Figure 6.3: Network ranking measures undergo sensitivity testing by ranking the network with a single node removed and comparing to the original ranking. This is repeated for each possible node removal in the network for all ranking measures. All measures show an average positive correlation between new and original rankings, with a high degree of variance in the sensitivity. PageRank has the best worst-case and worse average-case while topological sorting has the best average-case and worst worst-case.

The eigenvector, PageRank and Massey methods all have similar characteristics of sensitivity, while the topological sorting has a slightly more skewed distribution. The directed acyclic graph means that any node removal of a leaf (a node without any children), will have no change in the ranking aside from its removal, creating a strong cluster of results at $\tau = 0$. In contrast, removal of nodes in the central paths of the DAG will likely result in a new DAG structure being generated, strongly altering the rankings.

Importantly, all of these measures have a comparable level of sensitivity, without any stand-outs. Does this mean any of the measures are sufficient as rankings? To answer this we turn to comparing the methods directly.

We use Kendall's τ to compare the ranking of the information flow network produced by each measure. Notably in Figure 6.4, all rankings have a comparative τ of less than 0.11 between each other. This score indicates that the coherence between these measures is limited. The heuristic topological sorting deviates the most from the other measures rankings, likely due to this method not using all of the non-negative adjacency matrix of the full graph.

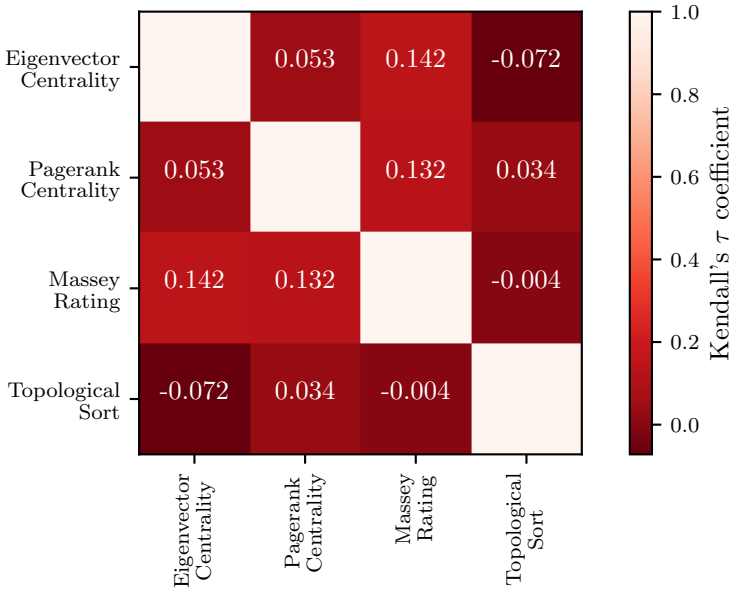


Figure 6.4: A comparison of the ordinal association between the rankings of each measure when applied to the information flow graph. All four measures show a limited correlation between each other, with the topological approach differing most from the non-negative matrix approaches.

To reinforce this, Table 6.1 shows the top ten most influential news-media organisations according to each of the ranking methods. While many organisations are common to each ranking's top 10 (e.g. *USA Today*, *The Washington Post*) , the detailed order differs significantly among each of the rankings. This is representative of the differences through the range of rankings for each method. Full lists of rankings are available in Appendix D.

Regression of rank onto organisation metadata

An analysis was performed to examine the impact of news-organisation size and bias on ranking. For each ranking approach, a series of linear regressions were performed fitting the rank of an organisation to its follower count,

Rank	Eigencentraility	Pagerank	Massey	Topological Sort
1	usatoday	usatoday	rightsidenews	business
2	huffpost	nytimes	usatoday	realdailywire
3	washingtonpost	bostonglobe	YahooNews	nytopinion
4	bostonglobe	washingtonpost	huffpost	time
5	YahooNews	huffpost	voxdotcom	newyorker
6	voxdotcom	npr	washingtonpost	NYMag
7	npr	voxdotcom	DeseretNews	RollingStone
8	nytimes	DeseretNews	bostonglobe	theatlantic
9	DeseretNews	latimes	townhallcom	motherjones
10	rightsidenews	WestJournalism	npr	rawstory

Table 6.1: Top 10 most influential news-media organisations according to each measure, listed as their Twitter account handles. The rankings can differ significantly across methods although many of the top 10 are shared between methods.

number of tweets and political bias as introduced in Chapter 3. Regressions were performed for all possible combinations of explanatory variables. Political bias was treated as a categorical variable and all models included an intercept.

For eigencentraility, pagerank and Massey rankings, none of the 21 models had sufficient significance to reject the null model at the 5% level. The topological ranking had significant models, but among them only the follower count was a significant variable. In these models a higher follower count was positively correlated with a better ranking; while statistically significant, this only explained 5.2% of the variance.

6.3 Discussion

Ranking methodologies are inherently fraught with assumptions and bias. As is the case here, ranking influence in a flow graph is underpinned by a constructed notion of influence. These assumptions are contextual and can be well justified; however communicating these assumptions can be difficult when a viewer seeks a simple definitive list. This exacerbates the challenge of communication, where assumptions and bias must be so clear as to be obvious to a viewer. In some cases this is possible, such as where the ranking metric is a simple statistic (i.e. students average grades in a course, the market capitalisation of a company) or where the ranking principles are well

established (i.e. the use of Elo in chess ranking). In situations where the source material is more complex – such as the case of a flow graph – norms and assumptions around ranking is not well established, and communication of these can prove difficult.

Even with clearly established assumptions, often no method of ranking is clearly superior. It is often dangerous to choose a single method *ad-hoc*, as each method comes with its own trade-offs. Consider the case of the Elo ranking system [25]; this ranking system is only a useful comparative tool within its own rating pool. If two individuals are ranked equally first or have the same high Elo across two sports, say chess and table tennis, it's difficult to compare how exceptional these two individuals are to each other without a knowledge of the ratings of many other players. Elo himself noted the difficulty of rating players in the 1962 Chess Life magazine, comparing it to "the measurement of the position of a cork bobbing up and down on the surface of agitated water with a yard stick tied to a rope and which is swaying in the wind" [24]. Indeed, most popular rating or ranking systems have their critics. The desire to fairly rate academic influence has brought criticism to naive citation counts in favour of alternative metrics such as the h-index [41], which has itself spawned its own academic niche through its many derivatives and alternatives [3].

Even in cases where methods are well respected and assumptions are well defined, these methods can have major flaws. Take for example the case of PageRank centrality discussed above. This centrality is considered robust and well established in the field of web page ranking, and underpins the original Google search engine [10]. Despite this, PageRank centrality can have dangerous sensitivity even in the context of a web-graph [59]. We saw ourselves in Figure 6.3 that small changes in a complex network can have a large effect on the rankings. Many rating methods are exposed to sensitivity concerns under the right network conditions, and these risks can be hard to both account for and to communicate to a viewer.

An alternative approach may be to join these ranking methods into a meta-ranking. These often either average various rankings or ratings, or use a voting process between methods to combine results into a final ranking. While this approach may appear a tool for 'smoothing' the assumptions and bias of various methods, it can again prove treacherous as choices of methods to include and how to combine these methods can strongly alter the final rankings. The breadth of combinations of approaches cannot be understated, even when data is somewhat limited [8]. While meta-ranking approaches likely do assist in creating more generally robust measures, they do not answer our fundamental problem with these rankings.

The search for a definitive ranking is a flawed venture. The purpose of

information stored in a complex network is not to be over-simplified into facile representations. News is a dynamic interconnected system, which is already simplified into net information flow edges in a network. Reducing this further can lead to deeply flawed or misleading narratives, often without it being clear what error has been made.

As discussed in Section 5.4, the network can be used to answer some specific questions, such as in comparing magnitude difference of edges or discussing a the flow between a single pair of organisations. These question can be extended to look at local neighbourhood structure. For example, if a news-media outlet has only outgoing information flow edges, than we can definitively say that they are a completely-net-contributor of information into the ecosystem. This is true for a single node in our data, *USA Today*, although these out-flowing edges are mostly low weight and therefore susceptible to error from noise.

Further, some clear network results can be extracted at the extremes of results. For example, while it may be hard to compare the total influence of the *New York Times* to the total influence of the *Washington Post*, we can concluded that both are more influential than *Arkansas Democrat-Gazette* based on the rankings from all methods.

While this chapter does not provide a convenient definitive ranking, it highlights the caution needed in investigating the network, as any result is grounded in the assumption made during the analysis. This difficulty in creating a clean ranking reflects in inherent complexity in the network of information flows and highlights the fact that the news ecosystem is itself complex.

Chapter 7

Conclusion

7.1 Summary and contribution to literature

The purpose of this thesis has been to outline and implement a methodology for estimating textual information flows between news-media organisations using data from social media and to explore these flows in a dataset from 2019 on Twitter. Importantly, these methods are context-free and capture all aspects of grammar and word choice present in text while maintaining the strict forward-in-time flow of information.

To achieve this goal, Chapter 3 outlined our collection of 2,977,980 tweets corresponding to a years worth of content from 154 news-media organisations on Twitter. These accounts represented a wide range of the political spectrum and news-topic interests (e.g. sports, science, politics, foreign policy), with each account having over 10,000 follows. While US-centric, these account provided a large corpus of data with which to develop and apply our information flow analysis.

Using this data and synthetic text generation models based on it, Chapter 4 introduced, formalised and verified the convergence of a non-parametric entropy rate estimator based on match lengths. This entropy rate estimator was then generalised into a cross-entropy rate estimator which carefully incorporated time-dependency into its cross-source match length calculations.

Chapter 5 examined the use of these cross-entropy rate estimates as a measure of information flow, finding that simple entropies on their own are insufficient to capture the flow. Several new approaches to information flow estimation were introduced and their performance compared using simulations of quoting text on random networks of natural language text generators. These measures produce strong correlations with the true flow, but small quote probabilities are hard to detect amongst the inherent noise in

natural language. This measure is then applied to the collected news Twitter data to produce a directed network of news-media organisations with edge weights representing information flow between them.

This information flow network allows for investigation into the direction and magnitude of net information flow between two organisations and facilitates the comparison of flow magnitudes between different pairs of news-media organisations.

Chapter 6 discovers that net influence of organisations in the network cannot be easily ranked as ranking methods are subject to high levels of sensitivity and different ranking approaches do not produce concordant results. The discussion of this limitation draws on two key ideas: that each ranking method is built from assumptions which are not always shared by other ranking approaches; and that attempting to create a robust simplification of a complex network is inherently fraught without over-reliance on those assumptions. In this sense, the investigation into ranking reveals the level of complexity in both the network of information flows and the news ecosystem from which it is built.

From these results this thesis contributes to the original literature in three main ways:

- the analysis of limits in the use of simple cross entropy estimators for robustly identifying information flow;
- the introduction and validation of new measures of information-theoretic information flow that perform well in natural language systems;
- and the application of information flow measures to a large corpus of textual news data from Twitter and the analysis of the resulting graph.

7.2 Future research

While this research contributes to the literature in several ways, it also provides a platform for future research. Four clear areas with room for improvement are the extension of this measure to shorter time periods, the use of alternative flow measures, the further analysis of the news-media information flow network and the application of these techniques to other datasets.

A shortcoming of the approach in this thesis is the averaging of information flow across the entire time period. While this helps counteract random noise from language and ensures convergence of estimators, if estimation could be performed using text data over smaller sub-periods of time, a new

lens of temporal information flow could be explored. Rather than examining net information flow over a year, a finer grained temporal analysis could explore the impact of major stories throughout the year on the information flow ecosystem. In addition to this, added metadata about the individual stories being produced – such as information about journalistic awards or breaking-news stories – could unlock interesting research question about the relationship between journalistic quality and impact.

An alternative to the refinement of these non-parametric information flow measures is their replacement with other formulations of information. This work used match-lengths on sequences of tokens, but recent advances in transformer based natural language models has allowed for contextual embeddings of language. These embeddings and other contextual information tools could provide an interesting alternative approach to measuring information flow in systems of textual data.

The analysis of the constructed news-media information flow network was limited within this thesis in favour of focusing on the information estimation process. This network still has many interesting features yet to be explored and a more detailed analysis of the dynamics of flow may prove interesting as future research.

Finally, we hope the tools developed in this thesis are useful in future research studying information flows in any and every ecosystem where individuals or organisations produce and exchange textual information.

Appendix A

News media Twitter accounts

A.1 Included news-media organisations

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
The New York Times	nytimes	Lean Left	31029	44800317
CNN	cnn	Left	57155	44153462
BBC News (World)	BBCWorld	Center	11679	26446876
The Economist	theeconomist	Lean Left	35771	24239639
Reuters	reuters	Center	128448	21042215
The Wall Street Journal	WSJ	Center	32151	17139922
TIME	time	Lean Left	29631	16512854
Forbes	Forbes	Center	31940	15736852
ABC News	ABC	Lean Left	44149	14804638
The Washington Post	washingtonpost	Lean Left	45043	14644580
The Associated Press	ap	Center	10214	13671750
HuffPost	huffpost	Left	27622	11472382
TechCrunch	techcrunch	Center	14348	10124437
Mashable	mashable	Left	40854	9809420
The New Yorker	newyorker	Left	13948	8774527
The Daily Show	thedailyshow	Lean Left	3386	8256511
The Guardian	guardian	Lean Left	77532	8243012
NPR	npr	Center	21236	7959364

Continued on next page

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
CBS News	CBSNews	Lean Left	34060	7078950
Rolling Stone	RollingStone	Left	18571	6295950
Bloomberg	business	Center	109367	5790419
VANITY FAIR	vanityfair	Lean Left	15585	4887894
TODAY	TODAYshow	Lean Left	17678	4282063
Lifehacker	lifehacker	Center	9026	4153410
POLITICO	politico	Lean Left	25742	4016243
USA TODAY	usatoday	Center	27303	3949159
Financial Times	FT	Center	41030	3806483
Scientific American	sciam	Center	2335	3805593
The Hill	thehill	Center	157172	3501890
Los Angeles Times	latimes	Lean Left	35341	3473994
Newsweek	newsweek	Lean Left	46922	3407905
Teen Vogue	TeenVogue	Lean Left	12270	3358001
CNBC	cnbc	Center	66578	3355408
MSNBC	msnbc	Left	32699	2870702
Business Insider	businessinsider	Center	57915	2790009
The Telegraph	Telegraph	Lean Right	24235	2774439
The Verge	verge	Lean Left	19096	2613577
Daily Mail Online	MailOnline	Right	28671	2437611
VICE	vice	Left	17195	2000034
CSPAN	cspan	Center	5022	1989017
The Atlantic	theatlantic	Lean Left	15100	1850164
New York Magazine	NYMag	Left	19072	1800255
Slate	slate	Left	58598	1792093
Al Jazeera News	AJENews	Center	11252	1573047
New York Post	nypost	Right	79771	1543609
BuzzFeed News	BuzzFeedNews	Lean Left	19705	1338555
Breitbart News	BreitbartNews	Right	17501	1225203
Yahoo News	YahooNews	Left	19781	1106092
Chicago Tribune	chicagotribune	Center	22994	1099101
AJC	ajc	Lean Left	27978	1045546
PBS NewsHour	NewsHour	Center	16189	1036993

Continued on next page

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
Salon	salon	Left	11614	976078
Vox	voxdotcom	Left	17574	891290
ProPublica	propublica	Center	6458	833038
Mother Jones	motherjones	Left	16447	809121
The Boston Globe	bostonglobe	Lean Left	49224	756522
The Intercept	theintercept	Left	6642	753001
New York Daily News	nydailynews	Left	28682	728512
Foreign Affairs	ForeignAffairs	Center	11964	724994
New York Times Opinion	nytopinion	Left	25402	718663
Democracy Now!	democracynow	Left	9453	710714
TheBlaze	theblaze	Right	13121	706991
Daily Caller	DailyCaller	Right	34716	579741
The Root	TheRoot	Lean Left	6994	525879
Upworthy	upworthy	Left	1472	511090
One America News	OANN	Lean Right	6108	510392
Chicago Sun-Times	Suntimes	Lean Left	15976	505161
SFGate	sfgate	Lean Left	17838	477782
Miami Herald	MiamiHerald	Lean Left	19290	450869
The Jerusalem Post	Jerusalem_Post	Center	29868	445004
Esquire	esquire	Left	12817	418881
Quartz	qz	Center	28604	384220
The Washington Times	WashTimes	Lean Right	34635	372421
Roll Call	rollcall	Center	10157	361580
The Daily Wire	realdailywire	Right	26385	361505
National Review	NRO	Right	16572	336300
Axios	axios	Center	19184	315533
Austin Statesman	statesman	Lean Left	9562	300429
Daily Kos	dailykos	Left	12636	280636
reason	reason	Lean Right	7939	242262
Mercury News	mercnews	Lean Left	36535	241228
Jacobin	jacobinmag	Left	8579	241092
ScienceDaily	sciencedaily	Center	1243	240013
Las Vegas Sun	LasVegasSun	Lean Left	8233	238797

Continued on next page

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
grist	grist	Lean Left	6438	230344
Sacramento Bee	sacbee_news	Lean Left	22511	219066
RedState	redstate	Right	19778	218545
The Federalist	FDRLST	Right	5830	217053
O.C. Register	ocregister	Lean Right	23872	212517
SF Weekly	sfweekly	Center	4438	210524
Raw Story	rawstory	Left	29702	205992
Washington Examiner	dcexaminer	Lean Right	61833	198229
OBSERVER	observer	Center	5856	195067
San Francisco Chronicle	sfchronicle	Left	18021	189685
KSL	KSLcom	Right	11881	179424
Truthout	truthout	Lean Left	7806	170572
The New Republic	newrepublic	Left	10725	167916
Investors.com	IBDinvestors	Lean Right	2469	167242
Pittsburgh Post-Gazette	PittsburghPG	Lean Right	24711	166684
Commercial Appeal	memphisnews	Lean Left	15885	162002
Mediaite	Mediaite	Lean Left	14030	159980
Townhall.com	townhallcom	Right	10452	152082
U.S. News	usnews	Lean Left	12827	150674
AlterNet	alternet	Left	7156	139194
National Journal	nationaljournal	Center	1424	133097
The Week	theweek	Center	9031	129703
CBN News	CBNNews	Right	15627	128487
Intl. Business Times	IBTimes	Center	11519	123358
CNSNews.com	cnsnews	Right	9035	119236
Times-Dispatch	RTDNEWS	Lean Right	7018	116510
Free Beacon	FreeBeacon	Right	9091	110634
Boston Herald	bostonherald	Lean Right	12550	105046
Newsmax	newsmax	Right	3825	100437
Current Affairs	curaffairs	Left	2188	99136
Deseret News	DeseretNews	Lean Right	15087	97060
WSJ Editorial Page	WSJopinion	Lean Right	4990	96734
Bustle	bustle	Lean Left	1504	96170

Continued on next page

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
Courier Journal	courierjournal	Lean Left	13988	88551
Portland Press Herald	PressHerald	Center	6372	85834
Defense One	DefenseOne	Center	6892	81992
The Nation	The_Nation	Left	13048	78860
Christian Science Monitor	csmonitor	Center	5676	75406
AR Democrat-Gazette	ArkansasOnline	Left	10951	74177
INDY Week	indyweek	Lean Left	4175	73802
PoliticusUSA	PoliticusUSA	Left	7253	72856
The Daily Signal	Dailysignal	Right	6740	68024
PJ Media	PJMedia_com	Lean Right	5666	66324
Delco Times	delcotimes	Lean Left	3302	64714
Daily Press	Daily_Press	Lean Right	3512	61225
YES! Magazine	yesmagazine	Left	4679	60455
SpokesmanReview	SpokesmanReview	Lean Left	4113	58352
Tallahassee Democrat	TDOnline	Left	15098	53700
The Korea Herald	TheKoreaHerald	Center	8730	53517
The Michigan Daily	michigandaily	Lean Left	2041	50748
Honolulu Civil Beat	CivilBeat	Center	1837	50503
Libertarian Republic	TheLibRepublic	Lean Right	1145	47475
The American Conservative	amconmag	Lean Right	25197	46189
The American Spectator	amspectator	Right	1915	45877
The Red & Black	redandblack	Center	7663	42735
KQED News	kqednews	Center	8655	41720
Indiana Daily Student	idsnews	Center	3867	41621
WGBH	wgbh	Center	2211	39971
The Western Journal	WestJournalism	Right	7114	38478
Commentary Magazine	Commentary	Right	1337	35704
The Daily Progress	DailyProgress	Center	6856	35506
VTDigger	vtdigger	Lean Left	5671	32820
BG Daily News	bgdailynews	Lean Left	7686	22341
Longmont Times-Call	TimesCall	Lean Left	3858	21077

Continued on next page

Table A.1: All included news-media organisations in clean dataset with full names, Twitter handles, the number of Twitter account followers, the numbers of tweets for 2019, and the political bias assigned to the organisation by AllSides.

Name	Twitter Account	Assigned Bias	Number of Tweets	Followers
The Daily Northwestern	thedailynu	Lean Left	4580	20357
Right Side News	rightsidenews	Right	4033	17600
WFAE	wfae	Center	4317	17385
The College Fix	CollegeFix	Right	4094	15289
The Chronicle	DukeChronicle	Center	1537	15207
CalMatters	calmatters	Center	3247	15069

A.2 Excluded news-media organisations

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
InfoWars	None	Right	No Twitter account
AllSides	None	Mixed	No Twitter account
Aquinas College Saint	None	Left	No Twitter account
Conservative HQ	None	Right	No Twitter account
Right Wing News	None	Right	No Twitter account
Canyon County Zephyr	None	Left	No Twitter account
Boston Herald Editorial	None	Lean Right	No Twitter account
The Republican	None	Center	No Twitter account
Progressive Voices of Iowa	None	Left	No Twitter account
PXW News	None	Center	No Twitter account
Sky-Hi Daily News	None	Lean Left	No Twitter account
Test Source	None	Center	No Twitter account
The Reliable Bias	None	Center	No Twitter account
Center for Public Integrity	Publici	Lean Left	Account suspended
Inacow	inacowcom	Right	Single person or group, not an organisation
MichelleMalkin.com	michellemalkin	Right	Single person or group, not an organisation
Fact Checker Blog	GlennKesslerWP	Center	Single person or group, not an organisation
Drudge Report	DRUDGE_REPORT	Lean Right	Single person or group, not an organisation
The Gateway Pundit	gatewaypundit	Right	Single person or group, not an organisation

Continued on next page

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
Smerconish	smerconish	Center	Single person or group, not an organisation
Wake Up to Politics	wakeup2politics	Center	Single person or group, not an organisation
Intellectual Conservative	rach_ic	Lean Right	Single person or group, not an organisation
Mismatch.org	AllSidesNow	Mixed	Fact checking, not news
FactCheck.org	factcheckdotorg	Center	Fact checking, not news
PolitiFact	politifact	Lean Left	Fact checking, not news
Truth or Fiction	erumors	Center	Fact checking, not news
Media Matters	mmfa	Left	Fact checking, not news
MIT News	mit	Center	Not a news site
Harvard Business School	HarvardHBS	Lean Left	Not a news site
Rasmussen Reports	Rasmussen_Poll	Center	Not a news site
Boing Boing	boingboing	Left	Not a news site
Care 2	Care2	Left	Not a news site
Journalist's Resource	JournoResource	Center	Not a news site
ProCon.org	procon_org	Mixed	Not a news site
Socialist Alternative	SocialistAlt	Left	Not a news site
Jubilee Media	jubileemedia	Center	Not a news site
How Do We Fix It?	fixitshow	Center	Not a news site
FiveThirtyEight	fivethirtyeight	Center	Not a news site
Judicial Watch	JudicialWatch	Lean Right	Not a news site
HotAir	hotairblog	Lean Right	Not a news site
Live Action News	LiveActionNews	Lean Right	Not a news site

Continued on next page

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
Quillette	Quillette	Lean Right	Not a news site
City Journal	cityjournal	Right	Not a news site
Univision	univision	Lean Left	Not in English
Daily Beast	dailybeast	Left	No media presence
NBCNews.com	nbcnews	Lean Left	Superseded by sister/parent organisation
Media Research Center	theMRC	Right	Superseded by sister/parent organisation
NPR Editorial	npr	Lean Left	Duplicate organisation
Saturday Evening Post	SatEvePost	Center	Duplicate organisation
The Courier-Journal	courierjournal	Lean Left	Duplicate organisation
Watchdog.org	Watchdogorg	Lean Right	Hacked Twitter account
Washington Monthly	washmonthly	Lean Left	Inactive Account
Blue Virginia	bluevirginia	Left	Less than 10,000 followers
The Fiscal Times	TheFiscalTimes	Lean Right	Less than 10,000 followers
The Daily Cardinal	dailycardinal	Center	Less than 10,000 followers
Leesburg Today	leesburgtoday	Lean Right	Less than 10,000 followers
Socialist Project	socialism21	Left	Less than 10,000 followers
Record-Journal	Record_Journal	Center	Less than 10,000 followers
The Daily Targum	daily_targum	Lean Left	Less than 10,000 followers

Continued on next page

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
Countercurrents.org	Countercurrents	Lean Left	Less than 10,000 followers
The Oracle	USFOracle	Center	Less than 10,000 followers
Whatfinger News	WhatfingerNews	Right	Less than 10,000 followers
#ListenFirst Project	ListenFirstProj	Mixed	Less than 10,000 followers
The State Journal	statejournal	Lean Left	Less than 10,000 followers
Bearing Drift	bearingdrift	Right	Less than 10,000 followers
CalWatchdog	CalWatchdog	Center	Less than 10,000 followers
heralddemocrat	heralddemocrat	Left	Less than 10,000 followers
Wisconsin Gazette	wigazette	Lean Left	Less than 10,000 followers
Advocate Messenger	amnewsonline	Lean Left	Less than 10,000 followers
Falls Church News-Press	fcnp	Left	Less than 10,000 followers
The Volante	thevolante	Center	Less than 10,000 followers
NMPolitics.net	nmpoliticsnet	Center	Less than 10,000 followers
Trail Gazette	EPTrailGazette	Center	Less than 10,000 followers
The Saturday Evening Post	SatEvePost	Center	Less than 10,000 followers
The Flip Side	knowtheflipside	Mixed	Less than 10,000 followers
CU Independent	The_CUI	Center	Less than 10,000 followers

Continued on next page

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
Living Rm Convos	LivingRoomConvo	Mixed	Less than 10,000 followers
The Justice	thejustice	Lean Left	Less than 10,000 followers
Barnstable Patriot	BarnPat	Center	Less than 10,000 followers
The Cadiz Record	TheCadizRecord	Lean Left	Less than 10,000 followers
Centre View	CentreView	Lean Left	Less than 10,000 followers
CNN WebNews	CNNWebNews	Lean Left	Less than 10,000 followers
Counterpointing	countertweeter	Mixed	Less than 10,000 followers
The Independent FLC	flcindependent	Center	Less than 10,000 followers
HamptonRoadsMessengr	H_R_Messenger	Center	Less than 10,000 followers
Suspend Belief	SBeliefPodcast	Mixed	Less than 10,000 followers
CookPoliticalReport	CookPolitical	Center	Less than 10,000 lifetime Tweets
FrontPage Magazine	fpmag	Right	Less than 10,000 lifetime Tweets
Fox News	foxnews	Lean Right	Inactive since 2018
Fox News Opinion	FoxNewsOpinion	Right	Inactive since 2018
Fox News Latino	foxnewslatino	Right	Inactive since 2016
The Weekly Standard	weeklystandard	Right	Less that 1,000 tweets in 2019
RealClearPolitics	RealClearNews	Center	Less that 1,000 tweets in 2019
IJR	TheIJR	Lean Right	Less that 1,000 tweets in 2019

Continued on next page

Table A.2: A list of news-media organisations that are listed by AllSides but are not included in the cleaned data. The reason for removal from the dataset in listed.

Name	Twitter Account	Assigned Bias	Reason for Removal
WND News	worldnetdaily	Right	Less that 1,000 tweets in 2019
PRI	pri	Center	Less that 1,000 tweets in 2019
EurekAlert!	eurekalert	Center	Less that 1,000 tweets in 2019
FAIR	FAIRmediawatch	Center	Less that 1,000 tweets in 2019
Crowdpac	Crowdpac	Center	Less that 1,000 tweets in 2019
Inside Philanthropy	InsidePhilanthr	Center	Less that 1,000 tweets in 2019
Diplomatic Courier	diplocourier	Center	Less that 1,000 tweets in 2019
Peacock Panache	PeacockPanache	Left	Less that 1,000 tweets in 2019
Independent Voter	IVN	Center	Less that 1,000 tweets in 2019
American Thinker	americanthinker	Right	Large period of inactivity in 2019
Pacific Standard	PacificStand	Lean Left	Large period of inactivity in 2019
Philly.com	phillydotcom	Lean Left	Large period of inactivity in 2019
Splinter	splinter_news	Left	Large period of inactivity in 2019
ThinkProgress	thinkprogress	Left	Large period of inactivity in 2019

A.3 News-media organisation locations

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
The New York Times	44800317	New York City
CNN	44153462	NaN
BBC News (World)	26446876	London, UK
The Economist	24239639	London
Reuters	21042215	Around the world
Fox News	18496616	U.S.A.
The Wall Street Journal	17139922	New York, NY
TIME	16512854	NaN
Forbes	15736852	New York, NY
ABC News	14804638	New York City / Worldwide
The Washington Post	14644580	NaN
The Associated Press	13671750	Global
HuffPost	11472382	NaN
TechCrunch	10124437	San Francisco, CA
Mashable	9809420	NaN
The New Yorker	8774527	New York, NY
The Daily Show	8256511	NaN
The Guardian	8243012	London
NPR	7959364	NaN
CBS News	7078950	New York, NY
Rolling Stone	6295950	New York, New York
Bloomberg	5790419	New York and the World
VANITY FAIR	4887894	New York, NY
TODAY	4282063	Studio 1A
Lifehacker	4153410	NaN
POLITICO	4016243	Washington, D.C.
USA TODAY	3949159	McLean, Va.
Financial Times	3806483	London
Scientific American	3805593	New York City

Continued on next page

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
The Hill	3501890	Washington, DC
Los Angeles Times	3473994	NaN
Newsweek	3407905	New York, NY
Teen Vogue	3358001	NaN
CNBC	3355408	NaN
MSNBC	2870702	NaN
Business Insider	2790009	New York, NY
The Telegraph	2774439	London, UK
The Verge	2613577	New York
Daily Mail Online	2437611	NaN
VICE	2000034	NaN
CSPAN	1989017	Washington, D.C.
The Atlantic	1850164	Washington, D.C.
New York Magazine	1800255	New York, NY
Slate	1792093	NaN
Al Jazeera News	1573047	Doha, Qatar
New York Post	1543609	New York, NY
BuzzFeed News	1338555	NaN
Breitbart News	1225203	NaN
Yahoo News	1106092	New York City
Chicago Tribune	1099101	Chicago, IL
AJC	1045546	Atlanta, GA
PBS NewsHour	1036993	Arlington, VA | New York, NY
Salon	976078	NaN
Vox	891290	NaN
ProPublica	833038	New York, NY
ThinkProgress	823165	Washington, D.C.
Mother Jones	809121	NaN
The Boston Globe	756522	Boston, MA
The Intercept	753001	NaN
New York Daily News	728512	New York City
Foreign Affairs	724994	New York, NY
New York Times Opinion	718663	NYC, London, Paris, Hong Kong

Continued on next page

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
Democracy Now!	710714	New York
TheBlaze	706991	Dallas, TX
Daily Caller	579741	Washington, DC
Splinter	562456	NaN
The Root	525879	New York, NY
Upworthy	511090	The Internet
One America News	510392	NaN
Chicago Sun-Times	505161	Chicago, IL
SFGate	477782	San Francisco
Miami Herald	450869	Miami, FL
The Jerusalem Post	445004	Israel
Esquire	418881	New York, NY
Quartz	384220	The World
The Washington Times	372421	Washington, D.C.
Roll Call	361580	Washington, D.C.
The Daily Wire	361505	California, USA
National Review	336300	New York
The Weekly Standard	322998	Washington
Axios	315533	NaN
Austin Statesman	300429	Austin, Texas
Philly.com	282993	Philadelphia, PA
Daily Kos	280636	NaN
reason	242262	Washington, DC and Los Angeles
Mercury News	241228	Silicon Valley, CA
Jacobin	241092	New York City
ScienceDaily	240013	Rockville, MD
Las Vegas Sun	238797	Las Vegas, NV
grist	230344	Seattle, WA
The Sacramento Bee	219066	Sacramento, CA
RedState	218545	Washington, D.C.
The Federalist	217053	United States of America
O.C. Register	212517	Orange County, CA
SF Weekly	210524	San Francisco

Continued on next page

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
Raw Story	205992	Washington, DC
Washington Examiner	198229	NaN
OBSERVER	195067	New York City
San Francisco Chronicle	189685	San Francisco, CA
KSL	179424	Salt Lake City, Utah
Truthout	170572	United States
The New Republic	167916	New York, NY
Investors.com	167242	Los Angeles, CA
Pittsburgh Post-Gazette	166684	Pittsburgh, Pa.
Commercial Appeal	162002	Memphis, TN
Fox News Opinion	160176	New York, NY
Mediaite	159980	New York, NY
RealClearPolitics	156611	NaN
Townhall.com	152082	Washington D.C.
U.S. News	150674	Washington, DC
AlterNet	139194	NaN
National Journal	133097	Washington, D.C.
The Week	129703	New York, NY
CBN News	128487	D.C.-Nashville-Jerusalem-VA
Intl. Business Times	123358	New York, NY
Pacific Standard	122028	Santa Barbara, California
CNSNews.com	119236	DC Metro Area
IJR	119083	NaN
Times-Dispatch	116510	NaN
Free Beacon	110634	United States
Boston Herald	105046	Boston, MA
Newsmax	100437	United States
Current Affairs	99136	New Orleans, LA
Deseret News	97060	Salt Lake City, UT
WSJ Editorial Page	96734	New York
Bustle	96170	NaN
Courier Journal	88551	Louisville, Kentucky
Portland Press Herald	85834	Portland, Maine

Continued on next page

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
Defense One	81992	Washington, D.C.
The Nation	78860	Pakistan
WND News	76723	NaN
The Christian Science Monitor	75406	Boston, MA
AR Democrat-Gazette	74177	Little Rock, AR
INDY Week	73802	NaN
PoliticusUSA	72856	USA
PRI	71848	NaN
The Daily Signal	68024	NaN
PJ Media	66324	NaN
Delco Times	64714	Primos, PA
Daily Press	61225	Newport News, Virginia
YES! Magazine	60455	Seattle, WA
SpokesmanReview	58352	Spokane, WA
Fox News Latino	55749	NaN
Tallahassee Democrat	53700	Tallahassee, Florida
The Korea Herald	53517	Seoul, Korea
The Michigan Daily	50748	Ann Arbor, Michigan
Honolulu Civil Beat	50503	Honolulu, HI
Libertarian Republic	47475	NaN
The American Conservative	46189	NaN
The American Spectator	45877	Washington, DC
American Thinker	44149	Worldwide
The Red & Black	42735	Athens, GA
KQED News	41720	San Francisco, California
Indiana Daily Student	41621	Bloomington, Indiana
EurekAlert!	40855	Washington, D.C.
FAIR	40528	New York, NY
WGBH	39971	Boston
Crowdpac	38950	Northern Virginia
The Western Journal	38478	Phoenix, AZ

Continued on next page

Table A.3: The declared location and number of followers for each news-media organisation in the cleaned dataset. The declared location is the location as listed on the organisations Twitter account.

Organisation Name	Number of Followers	Declared Location
Commentary Magazine	35704	New York, NY
The Daily Progress	35506	Charlottesville, Virginia
VTDigger	32820	Montpelier, VT
BG Daily News	22341	Bowling Green, Ky.
Longmont Times-Call	21077	Longmont, Colorado
The Daily Northwestern	20357	Evanston, Ill.
Right Side News	17600	United States
WFAE	17385	Charlotte, NC
The College Fix	15289	Washington, D.C.
The Chronicle	15207	Durham, NC
CalMatters	15069	Sacramento, CA
Inside Philanthropy	13962	Los Angeles, CA
Diplomatic Courier	13324	Global
Peacock Panache	12539	NaN
Independent Voter	12054	NaN

Appendix B

News-Media organisation Twitter activity

This appendix includes the tweet activity over the 2019 calendar year for a select group of representative news-media organisations. In each figure, a red line at 5 tweets per day is shown as a reference for comparison between accounts. The full collection of 154 news-media organisations can be accessed online via figshare [79].

The news-media organisations with the largest number of tweets

These large organisations produce a wealth of content that is viewed by a large number of consumers. While these organisations slow down entropy estimation, they easily converge.

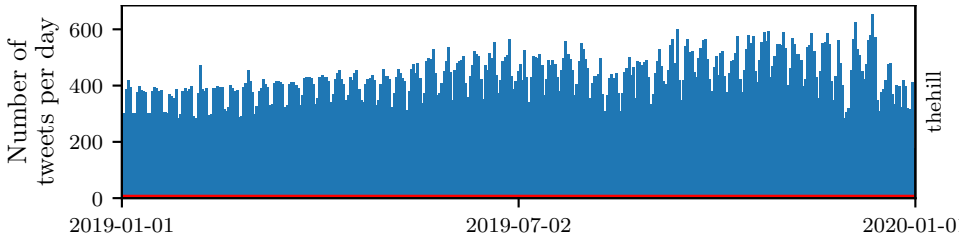


Figure B.1: Twitter activity over 2019 for 'The Hill' with 3501890 followers and 157172 total tweets. Red line is 5 tweets per day.

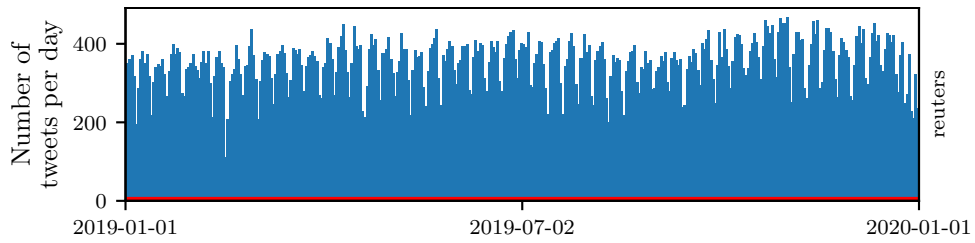


Figure B.2: Twitter activity over 2019 for 'Reuters' with 21042215 followers and 128448 total tweets. Red line is 5 tweets per day.

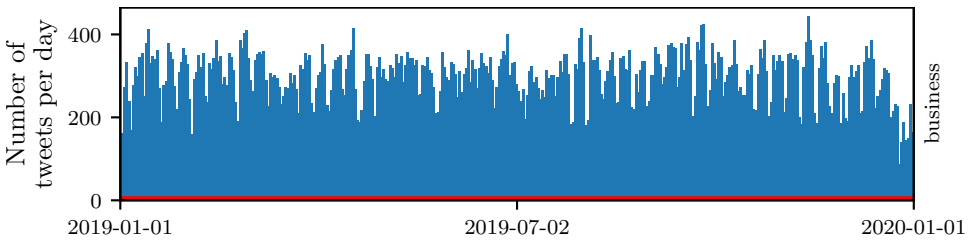


Figure B.3: Twitter activity over 2019 for 'Bloomberg' with 5790419 followers and 109367 total tweets. Red line is 5 tweets per day.

The news-media organisations with the smallest number of tweets

These news organisations are the smallest tweet counts that have passed through our filtering processes. Each organisation here exemplifies abnormalities that are exhibited by many of these low activity organisations. Figure B.4 shows both the low overall activity (which makes entropy rate estimation difficult) and the presence of large spikes in activity, Figure B.5 shows a sustained low activity over the year, with clear weekend dips in activity, and Figure B.6 show a initial period of low activity paired with a later period of higher activity starting in August.

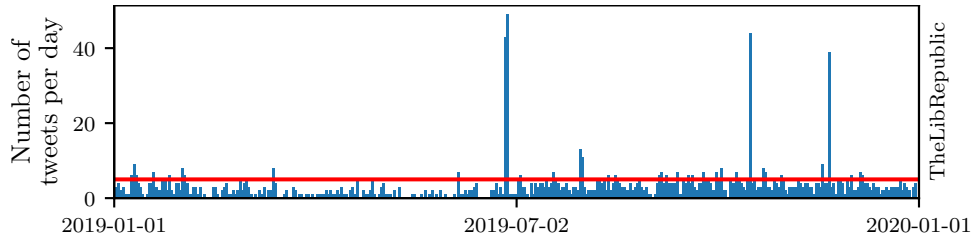


Figure B.4: Twitter activity over 2019 for 'Libertarian Republic' with 47475 followers and 1145 total tweets. Red line is 5 tweets per day.

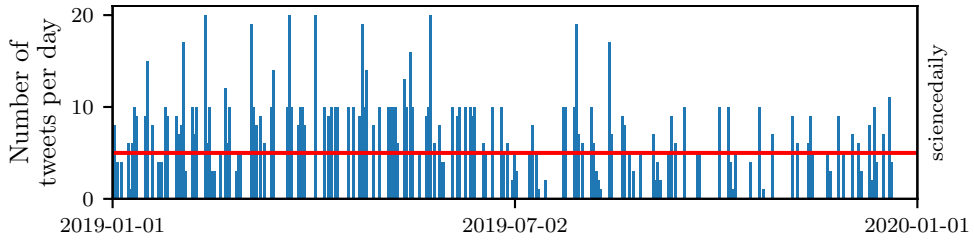


Figure B.5: Twitter activity over 2019 for 'ScienceDaily' with 240013 followers and 1243 total tweets. Red line is 5 tweets per day.

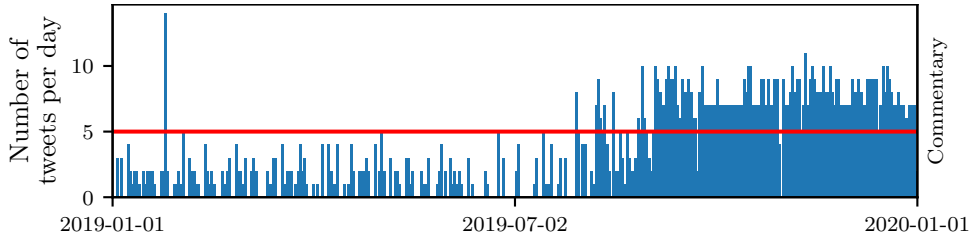


Figure B.6: Twitter activity over 2019 for 'Commentary Magazine' with 35704 followers and 1337 total tweets. Red line is 5 tweets per day.

Appendix C

ProcessEntropy: Open source high-speed entropy calculation package.

This source code is up-to-date and available on Github[1] and the Python Package Index (PyPi)[2].

Listing C.1: Kontoyianni cross entropy rate estimation code from the package 'ProcessEntropy' available at Github.

```
1  import numba
2  from numba import jit, prange
3  import numpy as np
4  import math
5  import nltk
6
7  from ProcessEntropy.Preprocessing import *
8
9
10 @jit(nopython=True, fastmath=True)
11 def find_lambda_jit(target, source):
12     """
13     Finds the longest subsequence of the target array,
14     starting from index 0, that is contained in the source array.
15     Returns the length of that subsequence + 1.
16
17     i.e. returns the length of the shortest subsequence starting at 0
18     that has not previously appeared.
19
20     Args:
21         target: NumPy array, preferable of type int.
22         source: NumPy array, preferable of type int.
23
24     Returns:
25         Integer of the length.
```

[1] https://github.com/tobinsouth/ProcessEntropy

[2] https://pypi.org/project/ProcessEntropy/

```
26
27      """
28
29      source_size = source.shape[0]-1
30      target_size = target.shape[0]-1
31      t_max = 0
32      c_max = 0
33
34      for si in range(0, source_size+1):
35          if source[si] == target[0]:
36              c_max = 1
37              for ei in range(1,min(target_size, source_size - si+1)):
38                  if(source[si+ei] != target[ei]):
39                      break
40                  else:
41                      c_max = c_max+1
42
43              if c_max > t_max:
44                  t_max = c_max
45
46      return t_max+1
47
48
49
50
51  @jit(parallel=True)
52  def get_all_lambdas(target, source, relative_pos, lambdas):
53      """
54      Finds all the the longest subsequences of the target,
55      that are contained in the sequence of the source,
56      with the source cut-off at the location set in relative_pos.
57
58      See function find_lambda_jit for description of
59          Lambda_i(target|source)
60
61      Args:
62          target: Array of ints, usually corresponding to hashed words.
63
64          source: Array of ints, usually corresponding to hashed words.
65
66          relative_pos: list of integers with the same length as target ↘
                denoting the
67              relative time ordering of target vs. source. These integers ↘
                    tell us the
68              position relative_pos[x] = i in source such that all symbols↘
                     in source[:i]
69              occurred before the x-th word in target.
70
71          lambdas: A pre-made array of length(target), usually filled with↘
                 zeros.
72              Used for efficiency reasons.
73
74      Return:
75          A list of ints, denoting the value for Lambda for each index in ↘
                the target.
76
77      """
78      i = 0
79      while relative_pos[i] == 0: # Preassign first values to avoid check
80          lambdas[i] = 1
81          i+=1
82
```

```
83      # Calculate lambdas
84      for i in prange(i, len(target)):
85          lambdas[i] = find_lambda_jit(target[i:], source[:relative_pos[i
                ]])
86
87      return lambdas
88
89
90  def timeseries_cross_entropy(time_tweets_target, time_tweets_source,
        please_sanitize = True, get_lambdas = False):
91      """
92      Finds the cross entropy H_cross(target|source) where the processes
            are embedded in time.
93
94      i.e. How many bits we would need to encode the data in target
95      using information in the source that is before the current time.
96
97
98      This is described mathematically in [1] as,
99
100     T = target
101     S = source
102
103     $$
104     \hat{h}_{ \times}(T | S)=\frac{N_{T} \log _{2} N_{S}}{\sum_{i=1}^{N_
            {T}} \Lambda_{i}(T | S)}
105     $$
106
107
108     Args:
109         time_tweets_target: A list of tuples with (time, tweet_content).
110             This is the stream of new information that we can testing
                    the ability to encode.
111             If please_sanitize = True (default) then tweet_content can
                    be a string.
112
113         time_tweets_source: A list of tuples with (time, tweet_content).
114             This is the stream of previous information from which we try
                    to encode the target.
115
116         please_sanitize: Option to have the tweet string converted to
                numpy int arrays for speed.
117             If False, please sanitize tweet into a list of tokenized
                    words, ideally converting these to ints,
118             via a hash.
119
120         get_lambdas: Boolean choice to return the list of all calculated
                Lambda values for each point
121             in target. Usually used for debugging.
122
123     Return:
124         The cross entropy as a float
125
126
127     [1] I. Kontoyiannis, P.H. Algoet, Yu.M. Suhov, and A.J. Wyner.
            Nonparametric entropy
128     estimation for stationary processes and random fields, with
            applications to English text.
129     IEEE Transactions on Information Theory, May 1998.
130
131     Credit to Bagrow and Mitchell for code ideas that I've stolen for
            this function.
```

```
132
133     """
134
135     # Decorate tweets (so we can distinguish the users), before sorting ↘
            in time:
136
137     if please_sanitize: # Option to have the tweet string converted to ↘
            numpy int arrays for speed.
138         # tweet_to_hash_array fucntion can be found in package.
139         decorated_target = [ (time,"target",tweet_to_hash_array(tweet)) ↘
                for time,tweet in time_tweets_target ]
140         decorated_source = [ (time,"source",tweet_to_hash_array(tweet)) ↘
                for time,tweet in time_tweets_source ]
141     else:
142         decorated_target = [ (time,"target",tweet) for time,tweet in ↘
                time_tweets_target ]
143         decorated_source = [ (time,"source",tweet) for time,tweet in ↘
                time_tweets_source ]
144
145     # Join time series:
146     time_tweets = decorated_target + decorated_source
147
148     # Sort in place by time:
149     time_tweets.sort()
150
151     # Loop over combined tweets and build word vectors and target->↘
            source relative_pos:
152     target, source, relative_pos = [], [], []
153     for time,user,tweet in time_tweets:
154         words = tweet
155         if user == "target":
156             target.extend(words)
157             relative_pos.extend( [len(source)]*len(words) )
158         else:
159             source.extend(words)
160
161
162     target = np.array(target, dtype = np.uint32)
163     source = np.array(source, dtype = np.uint32)
164     relative_pos = np.array(relative_pos, dtype = np.uint32)
165     lambdas = np.zeros(len(target), dtype = np.uint32) # Premake for ↘
            efficiency
166
167     lambdas = get_all_lambdas(target ,source, relative_pos, lambdas)
168
169     if get_lambdas:
170         return lambdas
171     return  len(target)*math.log(len(source),2) / np.sum(lambdas)
172
173
174 @jit(parallel=True)
175 def conditional_entropy(target, source,  get_lambdas = False):
176     """
177     Finds the simple conditional entropy as a process.
178
179     Entropy of target process conditional on full knowledge of states of↘
             source process.
180
181     Args:
182     target: A 1-D numpy array of integers.
183
```

```
184         This is the stream of new information that we can testing the ↘
                ability to encode.
185
186     source: A 1-D numpy array of integers.
187             This is the stream of previous information from which we try↘
                    to encode the target.
188
189        get_lambdas: Boolean choice to return the list of all calculated ↘
               Lambda values for each point
190             in target. Usually used for debugging.
191
192     Return:
193         The conditional entropy as a float
194     """
195     lambdas = np.zeros((1,len(target)))
196     for i in prange(0, len(target)):
197         lambdas[i] = find_lambda_jit(target[i:], source)
198
199     if get_lambdas:
200         return lambdas
201     else:
202         return  len(target)*math.log(len(source),2) / np.sum(lambdas)
```

Listing C.2: Kontoyianni entropy rate estimation code from the package 'ProcessEntropy' available at Github.

```
1  import numba
2  from numba import jit, prange
3  import numpy as np
4  import math
5  import nltk
6
7  from ProcessEntropy.Preprocessing import *
8
9  @jit(nopython = True)
10 def get_all_self_lambdas(source, lambdas):
11     """
12         Internal function.
13
14         Finds the Lambda value for each index in the source.
15
16         Lambda value denotes the longest subsequence of the source,
17         starting from the index, that in contained contigiously in the ↘
               source,
18         before the index.
19
20     Args:
21         source: Arry of ints, usually corresponding to hashed words.
22
23         lambdas: A premade array of length(target), usually filled with ↘
               zeros.
24             Used for efficiency reasons.
25
26     Return:
27         A list of ints, denoting the value for Lambda for each index in ↘
               the target.
28
29     """
30
31     N = len(source)
32
```

```
33      for i in prange(1, N):
34
35          # The target process is everything ahead of i.
36          t_max = 0
37          c_max = 0
38
39          for j in range(0, i): # Look back at the past
40              if source[j] == source[i]: # Check if matches future's next ↘
                    element
41                  c_max = 1
42                  for k in range(1,min(N-i, i-j)): # Look through more of ↘
                        future
43                      if source[j+k] != source[i+k]:
44                          break
45                      else:
46                          c_max = c_max+1
47
48                  if c_max > t_max:
49                      t_max = c_max
50
51          lambdas[i] = t_max+1
52
53      return lambdas
54
55
56
57  def self_entropy_rate(source, get_lambdas = False):
58      """
59      Args:
60          source: The source is an array of ints.
61      Returns:
62          The non-parametric estimate of the entropy rate based on match ↘
                lengths.
63
64
65      $$
66      \hat{h}(S)=\frac{N \log _{2} N{\sum_{i=1}^{N \Lambda_{i}(S)}
67      $$
68
69      This is described mathematically in [1] as,
70
71      [1] I. Kontoyiannis, P.H. Algoet, Yu.M. Suhov, and A.J. Wyner. ↘
            Nonparametric entropy
72      esti mation for stationary processes and random fields, with ↘
            applications to English text.
73      IEEE Transactions on Information Theory, May 1998.
74
75      """
76
77      N = len(source)
78      source = np.array(source)
79      lambdas = np.zeros(N)
80      lambdas = get_all_self_lambdas(source, lambdas)
81
82      if get_lambdas:
83          return lambdas
84      else:
85          return N*math.log(N,2) / np.sum(lambdas)
86
87
88  def text_array_self_entropy(token_source):
89          """
```

```
90          This is a wrapper for 'self_entropy_rate' to allow for raw text ↘
                to be used.
91
92          Args:
93                  token_source: A list of token strings (hint: a list of ↘
                        words).
94
95          Returns:
96          The non-parametric estimate of the entropy rate based on match ↘
                lengths.
97
98          """
99          return self_entropy_rate(np.array([fnv(word)  for word in ↘
                token_source]))
100
101 def tweet_self_entropy(tweets_source):
102         """
103         This is a wrapper for 'self_entropy_rate' to allow for raw ↘
                tweets to be used.
104
105         Args:
106                 tweets_source: A list of long strings (hint: a list of ↘
                        tweets).
107                         If it detects that you have added a list of time↘
                                , tweet pairs
108                         (as in timeseries_cross_entropy) it will recover↘
                                .
109
110         Returns:
111         The non-parametric estimate of the entropy rate based on match ↘
                lengths.
112
113         """
114         source = []
115
116         if type(tweets_source[0]) == tuple:
117                 # This is for the case of a
118                 for time, text in tweets_source:
119                         source.extend(tweet_to_hash_array(text))
120         else:
121                 for text in tweets_source:
122                         source.extend(tweet_to_hash_array(text))
123
124         return self_entropy_rate(source)
```

Listing C.3: Code for calculating predictability from the package 'ProcessEntropy' available at Github.

```
1  # This is a bonus file to help convert to predictabilties.
2
3  from scipy.optimize import fsolve
4  import numpy as np
5  import math
6
7  from ProcessEntropy.SelfEntropy import *
8  from ProcessEntropy.CrossEntropy import *
9
10 def predictability(S,N, inital_guess = 0.5):
11     """Finds the value of the predicatbility for a process with an ↘
           entropy rate S and a vocabular size N."""
12     # explodes for small values of N or large values of S :(
```

```
13      try:
14          f = lambda Pi : S + Pi*math.log(Pi,2) + (1 - Pi)*math.log(1 - Pi↘
                ,2) - (1 - Pi)*math.log(N-1,2)
15          PiMax = fsolve(f,inital_guess)
16      except:
17          PiMax = 0
18      return float(PiMax)
19
20
21  def process_predictability(process):
22      """Calculates the predictability of the process. """
23      entropy = nonparametric_entropy_estimate(process)
24      N = len(set(process))
25      return calc_predictability(entropy,N)
26
27
28
29  def cross_predictability(target,source):
30      """Calculates the predictability of the target given the information↘
             in the source."""
31      cross_entropy = timeseries_cross_entropy(target,source)
32      N = len(set(target)) # THIS IS WHERE I"M NOT SURE WHAT N TO USE
33      return predictability(entropy,N)
34
35
36  def surprise(probability):
37      """Returns surprise value for given probability"""
38      return log(1/probability,2)
```

Listing C.4: Code for preprocessing tweet and text data from the package 'ProcessEntropy' available at Github.

```
1   # This is a bonus file to help convert to predictabilties.
2
3   from scipy.optimize import fsolve
4   import numpy as np
5   import math
6
7   from ProcessEntropy.SelfEntropy import *
8   from ProcessEntropy.CrossEntropy import *
9
10  def predictability(S,N, inital_guess = 0.5):
11      """Finds the value of the predicatbility for a process with an ↘
            entropy rate S and a vocabular size N."""
12      # explodes for small values of N or large values of S :(
13      try:
14          f = lambda Pi : S + Pi*math.log(Pi,2) + (1 - Pi)*math.log(1 - Pi↘
                ,2) - (1 - Pi)*math.log(N-1,2)
15          PiMax = fsolve(f,inital_guess)
16      except:
17          PiMax = 0
18      return float(PiMax)
19
20
21  def process_predictability(process):
22      """Calculates the predictability of the process. """
23      entropy = nonparametric_entropy_estimate(process)
24      N = len(set(process))
25      return calc_predictability(entropy,N)
26
27
```

```
28
29  def cross_predictability(target,source):
30      """Calculates the predictability of the target given the information↘
            in the source."""
31      cross_entropy = timeseries_cross_entropy(target,source)
32      N = len(set(target)) # THIS IS WHERE I"M NOT SURE WHAT N TO USE
33      return predictability(entropy,N)
34
35
36  def surprise(probability):
37      """Returns surprise value for given probability"""
38      return log(1/probability,2)
```

Appendix D

Information flow influence rankings

Table D.1: The relative rankings of news-media organisations according to their influences on the information flow network as measured by four key ranking metrics.

Rank	Eigencentraility	PageRank	Mossey	Topological Sort
1	usatoday	usatoday	rightsidenews	business
2	huffpost	nytimes	usatoday	nytopinion
3	washingtonpost	bostonglobe	YahooNews	NYMag
4	bostonglobe	washingtonpost	huffpost	rawstory
5	YahooNews	huffpost	voxdotcom	theatlantic
6	voxdotcom	npr	washingtonpost	RollingStone
7	npr	voxdotcom	DeseretNews	motherjones
8	nytimes	DeseretNews	bostonglobe	time
9	DeseretNews	latimes	townhallcom	newyorker
10	rightsidenews	WestJournalism	npr	realdailywire
11	townhallcom	YahooNews	nytimes	cnbc
12	chicagotribune	rightsidenews	WestJournalism	cnsnews
13	latimes	chicagotribune	Suntimes	Jerusalem_Post
14	WestJournalism	sacbee_news	chicagotribune	CBNNews
15	Suntimes	Suntimes	sacbee_news	theintercept
16	sacbee_news	WSJ	latimes	theweek
17	WSJ	ABC	WSJ	ABC
18	NewsHour	townhallcom	PressHerald	politico
19	CBSNews	nydailynews	wfae	reason
20	DailyCaller	NewsHour	CBSNews	vanityfair

Continued on next page

Table D.1: The relative rankings of news-media organisations according to their influences on the information flow network as measured by four key ranking metrics.

Rank	Eigencentraility	PageRank	Mossey	Topological Sort
21	axios	CBSNews	NewsHour	sfchronicle
22	nydailynews	BuzzFeedNews	TheRoot	courierjournal
23	BuzzFeedNews	TODAYshow	salon	democracynow
24	TheRoot	axios	PittsburghPG	NRO
25	TODAYshow	TheLibRepublic	TODAYshow	FDRLST
26	KSLcom	TheRoot	kqednews	BuzzFeedNews
27	salon	KSLcom	DukeChronicle	WashTimes
28	FreeBeacon	DailyCaller	KSLcom	nytimes
29	kqednews	cnn	axios	businessinsider
30	ABC	ajc	Daily_Press	axios
31	WashTimes	PressHerald	ajc	IBTimes
32	wfae	kqednews	theblaze	latimes
33	PittsburghPG	PittsburghPG	FreeBeacon	theeconomist
34	ajc	salon	FDRLST	nypost
35	PressHerald	FreeBeacon	nydailynews	WSJ
36	theblaze	WashTimes	dailykos	vice
37	Daily_Press	wfae	RTDNEWS	csmonitor
38	politico	politico	sfgate	Telegraph
39	ap	theblaze	WashTimes	CBSNews
40	cnn	DukeChronicle	ap	nydailynews
41	FDRLST	FDRLST	ABC	cnn
42	dailykos	Daily_Press	nationaljournal	Forbes
43	DukeChronicle	ap	politico	usnews
44	TheLibRepublic	dailykos	TheLibRepublic	guardian
45	sfgate	sfgate	DailyCaller	truthout
46	RTDNEWS	RTDNEWS	statesman	BBCWorld
47	nationaljournal	sciencedaily	amspectator	newsweek
48	rollcall	nationaljournal	rollcall	TODAYshow
49	DailyProgress	rollcall	cnn	chicagotribune
50	amspectator	vice	vice	TheRoot
51	vice	DailyProgress	DailyProgress	TDOnline
52	statesman	amspectator	vanityfair	mercnews
53	LasVegasSun	vanityfair	reason	statesman
54	reason	statesman	democracynow	ajc
55	democracynow	democracynow	truthout	vtdigger

Continued on next page

Table D.1: The relative rankings of news-media organisations according to their influences on the information flow network as measured by four key ranking metrics.

Rank	Eigencentraility	PageRank	Mossey	Topological Sort
56	truthout	LasVegasSun	LasVegasSun	Suntimes
57	vanityfair	idsnews	MiamiHerald	ocregister
58	MiamiHerald	reason	sfchronicle	KSLcom
59	idsnews	truthout	idsnews	PittsburghPG
60	SpokesmanReview	sfchronicle	SpokesmanReview	thedailynu
61	sfchronicle	SpokesmanReview	theintercept	kqednews
62	alternet	MiamiHerald	thedailynu	sacbee_news
63	sciam	thedailynu	alternet	NewsHour
64	upworthy	alternet	vtdigger	PressHerald
65	thedailynu	upworthy	thedailyshow	YahooNews
66	sciencedaily	theintercept	upworthy	huffpost
67	michigandaily	michigandaily	michigandaily	rollcall
68	theintercept	vtdigger	businessinsider	amspectator
69	TDOnline	mercnews	theweek	nationaljournal
70	mercnews	dcexaminer	mercnews	Daily_Press
71	thedailyshow	thedailyshow	BBCWorld	salon
72	time	NRO	Forbes	thedailyshow
73	Forbes	time	time	usatoday
74	NRO	TDOnline	nytopinion	Mediaite
75	BBCWorld	nytopinion	calmatters	theblaze
76	vtdigger	delcotimes	TDOnline	DeseretNews
77	businessinsider	calmatters	curaffairs	DukeChronicle
78	nypost	BBCWorld	NRO	WestJournalism
79	theweek	businessinsider	Mediaite	alternet
80	dcexaminer	theweek	sciam	dailykos
81	nytopinion	Mediaite	Telegraph	PoliticusUSA
82	calmatters	curaffairs	PJMedia_com	FreeBeacon
83	PoliticusUSA	Forbes	usnews	LasVegasSun
84	Mediaite	memphisnews	csmonitor	voxdotcom
85	PJMedia_com	PJMedia_com	business	npr
86	curaffairs	PoliticusUSA	NYMag	bostonglobe
87	delcotimes	nypost	courierjournal	washingtonpost
88	memphisnews	sciam	RollingStone	townhallcom
89	guardian	RollingStone	guardian	DailyCaller
90	CivilBeat	courierjournal	memphisnews	ap

Continued on next page

Table D.1: The relative rankings of news-media organisations according to their influences on the information flow network as measured by four key ranking metrics.

Rank	Eigencentraility	PageRank	Mossey	Topological Sort
91	usnews	CivilBeat	PoliticusUSA	AJENews
92	Telegraph	AJENews	dcexaminer	rightsidenews
93	MailOnline	business	nypost	calmatters
94	IBTimes	TimesCall	motherjones	TheLibRepublic
95	csmonitor	ocregister	CBNNews	reuters
96	ocregister	Telegraph	msnbc	qz
97	RollingStone	cnsnews	ocregister	DailyProgress
98	AJENews	csmonitor	Jerusalem_Post	bostonherald
99	courierjournal	guardian	delcotimes	FT
100	cnsnews	usnews	IBTimes	sfgate
101	TimesCall	IBTimes	AJENews	michigandaily
102	business	NYMag	bostonherald	observer
103	theeconomist	bostonherald	Dailysignal	PJMedia_com
104	techcrunch	observer	newyorker	idsnews
105	bostonherald	CollegeFix	cnsnews	wfae
106	Jerusalem_Post	OANN	CivilBeat	CivilBeat
107	newsweek	CBNNews	observer	MailOnline
108	bustle	Jerusalem_Post	CollegeFix	techcrunch
109	OANN	msnbc	cnbc	SpokesmanReview
110	NYMag	bustle	realdailywire	RTDNEWS
111	FT	motherjones	newrepublic	sciam
112	qz	techcrunch	rawstory	MiamiHerald
113	msnbc	wgbh	theatlantic	TimesCall
114	motherjones	Dailysignal	theeconomist	upworthy
115	observer	cspan	newsweek	yesmagazine
116	CBNNews	rawstory	bustle	sciencedaily
117	wgbh	newyorker	techcrunch	OANN
118	CollegeFix	realdailywire	esquire	delcotimes
119	rawstory	cnbc	qz	verge
120	yesmagazine	FT	TeenVogue	esquire
121	theatlantic	esquire	FT	mashable
122	newyorker	bgdailynews	wgbh	Dailysignal
123	Dailysignal	newrepublic	OANN	newrepublic
124	cspan	theeconomist	yesmagazine	indyweek
125	realdailywire	qz	WSJopinion	lifehacker

Continued on next page

Table D.1: The relative rankings of news-media organisations according to their influences on the information flow network as measured by four key ranking metrics.

Rank	Eigencentraility	PageRank	Mossey	Topological Sort
126	cnbc	newsweek	cspan	The_Nation
127	reuters	theatlantic	propublica	redandblack
128	esquire	The_Nation	reuters	TheKoreaHerald
129	newrepublic	TeenVogue	BuzzFeedNews	curaffairs
130	The_Nation	yesmagazine	The_Nation	CollegeFix
131	verge	WSJopinion	grist	bustle
132	grist	TheKoreaHerald	TimesCall	memphisnews
133	bgdailynews	MailOnline	BreitbartNews	bgdailynews
134	TeenVogue	reuters	amconmag	slate
135	WSJopinion	propublica	jacobinmag	sfweekly
136	ArkansasOnline	grist	ArkansasOnline	newsmax
137	newsmax	BreitbartNews	thehill	amconmag
138	propublica	sfweekly	verge	propublica
139	BreitbartNews	ArkansasOnline	redandblack	redstate
140	mashable	newsmax	sciencedaily	cspan
141	TheKoreaHerald	verge	TheKoreaHerald	ForeignAffairs
142	redstate	redandblack	redstate	wgbh
143	sfweekly	jacobinmag	mashable	DefenseOne
144	redandblack	amconmag	MailOnline	TeenVogue
145	thehill	mashable	sfweekly	ArkansasOnline
146	jacobinmag	redstate	lifehacker	WSJopinion
147	amconmag	lifehacker	newsmax	IBDinvestors
148	lifehacker	indyweek	indyweek	jacobinmag
149	indyweek	thehill	bgdailynews	msnbc
150	IBDinvestors	DefenseOne	slate	dcexaminer
151	DefenseOne	IBDinvestors	ForeignAffairs	grist
152	ForeignAffairs	ForeignAffairs	DefenseOne	BreitbartNews
153	slate	slate	IBDinvestors	Commentary
154	Commentary	Commentary	Commentary	thehill

Bibliography

[1] J. B. Abramson, G. R. Orren, and F. C. Arterton. *Electronic Commonwealth: The Impact of New Media Technologies on Democratic Politics*. Basic Books, Inc., 1990.

[2] T. J. Allen and S. I. Cohen. Information flow in research and development laboratories. *Administrative Science Quarterly*, 14(1):12–19, 1969. ISSN 00018392. doi: 10.2307/2391357.

[3] S. Alonso, F. J. Cabrerizo, E. Herrera-Viedma, and F. Herrera. H-index: A review focused in its variants, computation and standardization for different scientific fields. *Journal of Informetrics*, 3(4):273–289, Oct. 2009. ISSN 1751-1577. doi: 10.1016/j.joi.2009.04.001.

[4] R. Baeza-Yates. Challenges in the interaction of information retrieval and natural language processing. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 445–456, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24630-5.

[5] J. P. Bagrow and L. Mitchell. The quoter model: A paradigmatic model of the social flow of written information. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075304, July 2018. ISSN 1054-1500. doi: 10.1063/1.5011403.

[6] J. P. Bagrow, X. Liu, and L. Mitchell. Information flow reveals prediction limits in online social activity. *Nature Human Behaviour*, 3(2):122–128, Feb. 2019. ISSN 2397-3374. doi: 10.1038/s41562-018-0510-5.

[7] P. Barberá, J. T. Jost, J. Nagler, J. A. Tucker, and R. Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological Science*, 26(10):1531–1542, Oct. 2015. ISSN 0956-7976, 1467-9280. doi: 10.1177/0956797615594620.

[8] D. Barrow, I. Drayer, P. Elliott, G. Gaut, and B. Osting. Ranking rankings: an empirical comparison of the predictive power of sports

ranking methods. *Journal of Quantitative Analysis in Sports*, 9(2):187–202, 2013. doi: doi:10.1515/jqas-2013-0013.

[9] T. Bell, I. H. Witten, and J. G. Cleary. Modeling for text compression. *ACM Computing Surveys*, 21(4):557–591, 1989.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117, 1998.

[11] A. D. Broido and A. Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1017, Mar. 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08746-5.

[12] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. C. Lai, and R. L. Mercer. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40, 1992.

[13] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, New York, second edition, 2002. ISBN 978-0-387-95364-9. doi: 10.1007/b97636.

[14] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, May 2009. doi: 10.1103/RevModPhys.81.591.

[15] S. Chen and J. H. Reif. Using difficulty of prediction to decrease computation: Fast sort, priority queue and convex hull on entropy bounded inputs. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 104–112. IEEE, 1993.

[16] S. Chen and J. H. Reif. Fast pattern matching for entropy bounded text. In *Proceedings DCC'95 Data Compression Conference*, pages 282–291. IEEE, 1995.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[18] E. Costenbader and T. W. Valente. The stability of centrality measures when networks are sampled. *Social Networks*, 25(4):283–307, Oct. 2003. ISSN 0378-8733. doi: 10.1016/S0378-8733(03)00012-1.

[19] T. M. Cover and R. King. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24(4):413–421, 1978.

[20] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Nov. 2012. ISBN 978-1-118-58577-1.

[21] Ł. Dębowski. Is natural language a perigraphic process? the theorem about facts and words revisited. *Entropy*, 20(2):85, Feb. 2018. doi: 10.3390/e20020085.

[22] A. Delgado-Bonal and J. Martín-Torres. Human vision is determined based on information theory. *Scientific Reports*, 6(1):36038, Nov. 2016. ISSN 2045-2322. doi: 10.1038/srep36038.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[24] A. Elo. Uscf rating formulae now reveal first scientific ranking of all living masters. *Chess Life*, 17(8):167, Aug. 1962.

[25] A. E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Pub., 1978. ISBN 978-0-923891-27-5.

[26] P. Erdos and A. Renyi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5 (1):17–60, 1960.

[27] R. M. Fano and D. Hawkins. Transmission of Information: A Statistical Theory of Communications. *American Journal of Physics*, 29:793–794, Nov. 1961. ISSN 0002-9505. doi: 10.1119/1.1937609.

[28] M. Farach, M. Noordewier, S. Savari, L. Shepp, A. Wyner, and J. Ziv. On the entropy of DNA: Algorithms and measurements based on memory and rapid convergence. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 48–57, 1995.

[29] J. Gable, S. McDonald, and J. Blades. Media bias ratings allsides. https://www.allsides.com/media-bias/media-bias-ratings, 2019.

[30] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959. ISSN 0003-4851.

[31] F. Godlee, J. Smith, and H. Marcovitch. Wakefield's article linking mmr vaccine and autism was fraudulent. *BMJ (Clinical Research Ed.)*, 342, 2011. ISSN 0959-8138. doi: 10.1136/bmj.c7452.

[32] R. L. Graham, D. E. Knuth, and O. Patashnik. Concrete mathematics: A foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

[33] P. Grassberger. Estimating the information content of symbol sequences and efficient codes. *IEEE Transactions on Information Theory*, 35(3): 669–675, May 1989. ISSN 00189448. doi: 10.1109/18.30993.

[34] J. Greenberg, editor. *Universals of language.* M.I.T. Press, Oxford, England, 1963.

[35] P. M. Greenfield. The changing psychology of culture from 1800 through 2000. *Psychological Science*, 24(9):1722–1731, 2013. doi: 10.1177/0956797613479387.

[36] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 491–501, New York, NY, USA, May 2004. Association for Computing Machinery. ISBN 978-1-58113-844-3. doi: 10.1145/988672.988739.

[37] A. Gupta, H. Lamba, and P. Kumaraguru. $1.00 per rt #boston-marathon #prayforboston: Analyzing fake content on twitter. In *2013 APWG ECrime Researchers Summit*, pages 1–12. IEEE, 2013.

[38] S. A. Hale. Net increase? cross-lingual linking in the blogosphere. *Journal of Computer-Mediated Communication*, 17(2):135–151, Jan. 2012. ISSN 1083-6101. doi: 10.1111/j.1083-6101.2011.01568.x.

[39] R. Hartley. Transmission of information. *The Bell System Technical Journal*, 7(3):535–563, 1928.

[40] H. S. Heaps. *Information retrieval, computational and theoretical aspects.* Academic Press, 1978.

[41] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences*, 102(46), 2005.

[42] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on Twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2751–2754, Austin, Texas, USA, May 2012. Association for Computing Machinery. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208672.

[43] J. Hutchins. From first conception to first demonstration: The nascent years of machine translation, 1947–1954. a chronology. *Machine Translation*, 12(3):195–252, 1997.

[44] T. Iqbal and S. Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 2020. ISSN 1319-1578. doi: https://doi.org/10.1016/j.jksuci.2020.04.001. URL https://www.sciencedirect.com/science/article/pii/S1319157820303360.

[45] D. Jones. *Censorship: A World Encyclopedia*. Taylor & Francis, 2001. ISBN 978-1-136-79864-1.

[46] P. Juola. What can we do with small corpora? Document categorization via cross-entropy. In *Proceedings of an Interdisciplinary Workshop on Similarity and Categorization, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, UK*, 1997.

[47] I. Kontoyiannis and Y. Suhov. Prefixes and the entropy rate for long-range sources. In *IEEE International Symposium on Information Theory*, pages 194–194. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1994.

[48] I. Kontoyiannis, P. Algoet, Y. Suhov, and A. Wyner. Nonparametric entropy estimation for stationary processes and random fields, with applications to English text. *IEEE Transactions on Information Theory*, 44(3):1319–1327, May 1998. ISSN 00189448. doi: 10.1109/18.669425.

[49] A. N. Langville and C. D. Meyer. *Who's #1?: The Science of Rating and Ranking*. Princeton University Press, 2012. ISBN 978-0-691-15422-0.

[50] F. Lundh. The stringlib library. http://effbot.org/zone/stringlib.htm, May 2006.

[51] K. Massey. *Statistical Models Applied to the Rating of Sports Teams*. Bachelor's thesis, Bluefield College, 1997.

[52] M. L. Mauldin. Chatterbots, tinymuds, and the turing test entering the loebner prize competition. In *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, AAAI'94, page 16–21. AAAI Press, 1994.

[53] J. Mellon and C. Prosser. Twitter and facebook are not representative of the general population: Political attitudes and demographics of british social media users. *Research & Politics*, 4(3):2053168017720008, July 2017. ISSN 2053-1680. doi: 10.1177/2053168017720008.

[54] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC press, 2018.

[55] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, , J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011. ISSN 0036-8075. doi: 10.1126/science.1199644.

[56] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[57] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., 2013.

[58] M. E. J. Newman. *Networks*. Oxford University Press, second edition, July 2018. ISBN 978-0-19-252749-3.

[59] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 903–910. Lawrence Erlbaum Associates Ltd, 2001.

[60] Q. Niu, A. Zeng, Y. Fan, and Z. Di. Robustness of centrality measures against network manipulation. *Physica A: Statistical Mechanics and its Applications*, 438:124–131, 2015. ISSN 0378-4371. doi: 10.1016/j.physa.2015.06.031.

[61] H. Nyquist. Certain factors affecting telegraph speed. *Transactions of the American Institute of Electrical Engineers*, 43:412–422, 1924.

[62] J. O'Donovan, H. F. Wagner, and S. Zeume. Value of offshore secrets: Evidence from the panama papers. *The Review of Financial Studies*, 32 (11):4117–4155, Feb. 2019. doi: 10.1093/rfs/hhz017.

[63] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[64] E. A. Pechenick, C. M. Danforth, and P. S. Dodds. Characterizing the google books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PLoS ONE*, 10(10):e0137041, 2015.

[65] R. Perera and P. Nand. Recent advances in natural language generation: A survey and classification of the empirical literature. *Computing and Informatics*, 36(1):1–32, 2017.

[66] T. Pond, S. Magsarjav, T. South, L. Mitchell, and J. P. Bagrow. Complex contagion features without social reinforcement in a model of social information flow. *Entropy*, 22(3):265, Mar. 2020. doi: 10.3390/e22030265.

[67] M. Potthast, T. Gollub, K. Komlossy, S. Schuster, M. Wiegmann, E. P. Garces Fernandez, M. Hagen, and B. Stein. Crowdsourcing a large corpus of clickbait on twitter. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1498–1507, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics.

[68] A. N. Quas. An entropy estimator for a class of infinite alphabet processes. *Theory of Probability & Its Applications*, 43(3):496–507, 1999.

[69] E. Reiter and R. Dale. *Building natural language generation systems*. Cambridge university press, 2000.

[70] T. Schürmann and P. Grassberger. Entropy estimation of symbol sequences. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 6 (3):414–427, 1996.

[71] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[72] C. E. Shannon. Prediction and entropy of printed English. *Bell system technical journal*, 30(1):50–64, 1951.

[73] E. Shearer and E. Grieco. Americans are wary of the role social media sites play in delivering the news, Oct. 2019.

[74] P. Shields and B. Weiss. Universal redundancy rates for the class of B-processes do not exist. *IEEE transactions on information theory*, 41 (2):508–512, 1995.

[75] P. C. Shields. Entropy and prefixes. *The Annals of Probability*, pages 403–409, 1992.

[76] P. C. Shields. Universal redundancy rates do not exist. *IEEE transactions on information theory*, 39(2):520–524, 1993.

[77] J. Skinner. Social media and revolution: The Arab Spring and the occupy movement as seen through three information studies paradigms. *Working Papers on Information Systems*, 11(169):2–26, 2011.

[78] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, Feb. 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1177170.

[79] T. South. News-media organisation twitter activity in 2019, 2 2021.

[80] T. South, M. Roughan, and L. Mitchell. Popularity and centrality in Spotify networks: Critical transitions in eigenvector centrality. *Complex Networks*, 2021.

[81] K. Starbird, J. Maddock, M. Orand, P. Achterman, and R. M. Mason. Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. *IConference 2014 Proceedings*, 2014.

[82] M. Tracy. The new york times tops 7.5 million subscriptions as ads decline. *The New York Times*, Feb. 2021. ISSN 0362-4331.

[83] D. Tran. The fourth estate as the final check, Nov. 2016.

[84] G. Ver Steeg and A. Galstyan. Information transfer in social media. In *Proceedings of the 21st international conference on World Wide Web*, pages 509–518, Lyon, France, 2012. Association for Computing Machinery.

[85] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat,

Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv:1907.10121 [physics]*, July 2019.

[86] F. Vis. Twitter as a reporting tool for breaking news. *Digital Journalism*, 1(1):27–47, Feb. 2013. ISSN 2167-0811. doi: 10.1080/21670811.2012.741316.

[87] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, Mar. 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aap9559.

[88] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998. ISSN 1476-4687. doi: 10.1038/30918.

[89] J. Weizenbaum. *Computer Power and Human Reason: From Judgment to Calculation.* Computer Power and Human Reason: From Judgment to Calculation. W. H. Freeman & Co, Oxford, England, 1976.

[90] J. R. Williams, J. P. Bagrow, C. M. Danforth, and P. S. Dodds. Text mixing shapes the anatomy of rank-frequency distributions. *Physical Review E*, 91(5):052811, May 2015. doi: 10.1103/PhysRevE.91.052811.

[91] F. Wu, B. A. Huberman, L. A. Adamic, and J. R. Tyler. Information flow in social groups. *Physica A: Statistical Mechanics and its Applications*, 337(1):327–335, June 2004. ISSN 0378-4371. doi: 10.1016/j.physa.2004.01.030.

[92] A. Wyner and J. Ziv. Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression. *IEEE Transactions on Information Theory*, 35(6):1250–1258, Nov./1989. ISSN 00189448. doi: 10.1109/18.45281.

[93] G. K. Zipf. The psychobiology of language, 1935.

[94] G. K. Zipf. *Human Behavior And The Principle Of Least Effort.* Addison-Wesley, 1949.

[95] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977. ISSN 0018-9448. doi: 10.1109/TIT.1977.1055714.

[96] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, Sept. 1978. ISSN 0018-9448. doi: 10.1109/TIT.1978.1055934.