UNIVERSITY OF CALIFORNIA

Los Angeles

An Architecture of Participation for Computational Social Biology

A dissertation submitted in partial satisfaction of the requirements

for the degree Doctor of Philosophy in Computer Science

by

Nicholas Williams Brown

2013

ABSTRACT OF THE DISSERTATION

by

Nicholas Williams Brown

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2013

Professor Joseph Di Stefano III, Chair

The nature and size of data on the social web are different from those used to develop most algorithms for statistics, data mining, and machine learning. The social web is far more dynamic than actuarial tables, lab tests, or historical records, in that social algorithms will affect people's behavior. Google's PageRank, for instance, has undergone hundreds of announced changes. A multibillion-dollar search engine optimization (SEO) industry has been created to adapt organizations to social algorithms. To address the adaptive nature of the social web, I present an "Architecture of Participation": a methodology and toolkit that allow one to measure, visualize, and evaluate the effectiveness of a given social-web medium and to better understand two (overlapping) sets of factors: those that influence user participation in generating content, and those that influence the quality of that content. The goal of my Architecture of Participation, then, is to better measure the dynamics of social networks in the context of these factors, with the subsidiary objective of more effectively utilizing collective intelligence in biology.

Behavior in a social network can be dynamically affected by friends or contacts, community, time of day, and many other factors that reflect the state of the network as a whole. The data coming from a social network, in short, manifests a complex, dynamic system. Hence, measurement tools must take the whole system into account, not just its pieces, as well as the extent of the social system memory summarized by its current state. To this end, the Architecture of Participation enables one to measure how system state affects social algorithms and user behavior. This Architecture includes semantic lexica, algorithms, and software that allow one to incorporate the dynamics of data and the system state into social analytics. I show that information-theoretic measures such as Rényi entropy and mutual information can be used classify an infinite set of states. I create unsupervised efficient, scalable and parallelizable algorithms to classify text in “big data.” I then develop a calculus that allows the combination these classifiers in to a search kernel in an intuitive and mathematically consistent way.

The dissertation of Nicholas Williams Brown is approved

D. Stott Parker

Matteo Pellegrini

Eleazar Eskin

Joseph Di Stefano III, Chair

University of California, Los Angeles 2013

Table of Contents

Biographical Sketch

Nicholas Williams Brown earned his Bachelor of Arts degree in Biochemistry and Molecular Biology from the University of California, Santa Cruz. He received his Master of Science degree in Computer Science in from the University of California, Los Angeles in 2006. In 2006, he joined the doctoral program at UCLA.

While pursuing his degree, he has taught computer science, programming, 3D visual effects, web programming, server administration, networking and game programming at UCLA, Santa Monica College, ITT, the Art Institutes - Hollywood and LA Film School. At UCLA, he co-authored eleven scientific publications in addition to the single author publications related to this work.

His dissertation entitled, “An Architecture of Participation for Computational Social Biology,” was supervised by Dr. Joseph DiStefano III.

Introduction

The difference between what are now popularly known as Web 1.0 and Web 2.0 is one of passive versus participatory media [O'Reilly, 2005]. In Web 1.0 the Internet is used as a data source wherein people aren't actively involved in adding their personal knowledge to the information they are consuming. Online biological databases (e.g. Entrez, Ensembl, Swiss-Prot, UCSC Genome Browser, KEGG and OMIM) and online analytical tools (e.g. Invitrogen Primer Designer, BLAST, ProteinProspector, and ProDom) are classic examples of Web 1.0. If one wants to enhance a Web 1.0 resource with one's own knowledge, the typical method is to send an e-mail to the curators and hope they accept the suggestion.

The critical algorithmic issues of the social web (Web 2.0) are dealing with the amount of the data (so-called "big data") and understanding user behavior in participatory media. The main idea behind this work is that social network data reflect complex dynamics. In a dynamic social system, one behavior is affected by one's own mood and others' behavior: that is, behavior is affected both by one's state and the state of the system as I have defined it. In any case, an individual's behavior is influenced by the behavior of another or others, and this includes participatory behavior. Whether an individual or a group is angry or amused, sated or hungry, trustful or distrustful, affects the kind and quality of user participation. The study of these phenomena is often referred to as "social dynamics"—a term that can refer to the behavior of groups that result from the interactions of individual group members as well to the relationships between individual interactions and group level behaviors. [Durlauf and Young 2004] The field of social dynamics is a subfield of complex adaptive systems or complexity science. Social dynamics is concerned with changes over time and emphasizes the role of feedbacks.

However, the laws governing Web 2.0 social networks, in particular the ways they evolve over time, are currently unclear. [Freeman, 2004] To understand the dynamics of such a network, therefore, one must measure its state over time. In turn, to create a meaningful measurement of state over time, I must be able to monitor the dynamics in their various forms: content evolution, network structure, information diffusion, and changes in state. The purpose of this research is to better measure the dynamics of social networks, particularly as they relate to social biology.

The Entropy of the Social Web

Social networks currently measure features such as user location, repeat visitors, singletons, bounce rate, visibility time, session duration, and many elements that could be considered measures of state. From a dynamical systems perspective, though, what matters are measures that reflect changes in the system state. For example, when water transitions from fluid to ice, it has gone through a state change. This research is concerned with measures of system state that define fundamentally different system behavior. Topological measures of the dynamics and structure of graphs, such betweenness [Brandes, 2001], closeness [Crucitti and Porta, 2006.], centrality [Opsahl, Agneessens, Skvoretz, 2010], node degree [Albert and Barabási, 2002], node strength [Barrat, Barth´elemy, Pastor-Satorras and Vespignani, 2004] and reach [Kadushin, 2011] are common in social network analysis.

Decades of research has improved our understanding of the quantitative analysis of the structure and dynamics of networks. The measures coming from complexity theory are primarily information-theoretic measures; the measures coming from the study of the dynamics of networks are primarily graph-theoretic measures. Currently, in fact, most research on the structure, dynamics, and state of social networks uses graph theory and is referred to as social network analysis (SNA). Many books on social network analysis present graph-theoretic–based techniques. While graph-theoretic techniques provide wonderful and interesting insight, most don’t scale to even moderately sized networks and certainly not to the scale of “big data”; trying to find complex structure and states in graphs is on this scale is computationally infeasible.

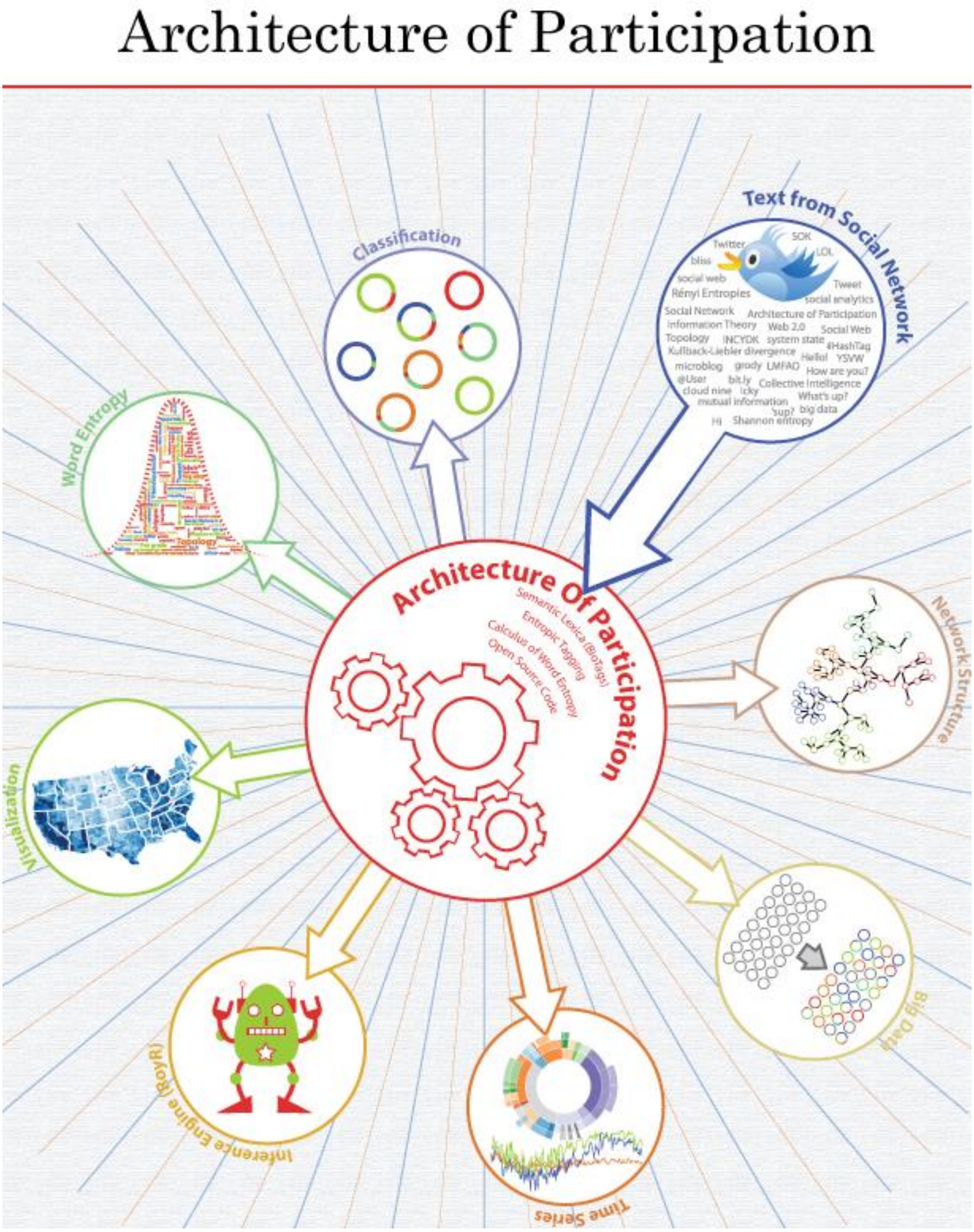
More generally, the number of possible connections in a network grows exponentially with the number of nodes in it; hence, social network analysis is typically limited to small networks. Information-theoretic techniques complement topological approaches to social network analysis by measuring aspects of state that are difficult for graph-based techniques as well as for pruning graphs; as a result, more computationally expensive approaches focus on the essential structure in a network. Using information-theoretic measures to quantify the dynamics and state within a social network provides a simple, general, and computationally efficient model that scales well to “big data.”

Ergodic theory [Denker, 1979][Walters, 1975] provides a possible computationally efficient supplement to social network analysis for understanding the state of dynamics of a network. Ergodic theory is a branch of mathematics that studies dynamical systems with invariant measures. In particular, ergodic theory applies many information-theoretic invariant measures of system state, such as the various notions of entropy for dynamical systems, to stochastic processes. There are many possible information-theoretic measures of system state to test as measures of social network dynamics: Kullback-Liebler divergence, Lyapunov exponents, Bayes's information criterion, the Hannan-Quinn criterion, Fisher information, Akaike information, Kolmogorov-Sinai entropy, and so forth. This research uses plogs, Kullback-Leibler divergence, and the Rényi entropies: Hartley entropy, Shannon entropy, collision entropy, and min-entropy. I chose these entropies because they can be efficiently calculated using text-based statistics—one of the most common features across all social networks.

In this research, I show that information-theoretic measures such as the Rényi entropies and mutual information can be used to develop classifiers that can estimate state. I then use the Rényi entropies of tags, the mutual information between tags, and heuristics to create unsupervised efficient, scalable and parallelizable algorithms to classify text in “big data.” I then use these algorithms to develop a calculus that allows us to combine any subset of classifiers in to a search kernel in an intuitive and mathematically consistent way.

I then build a number of classifiers using public datasets from Twitter [Twitter, 2013a], PubMed [PubMed/Medline, 2013], arXiv [arXiv, 2013], BBC news [Greene and Cunningham, 2006], AG's news corpus [AG's news corpus, 2013], last.fm music tags [35], the Foundational Model of Anatomy ontology (FMA) [Rosse and Mejino, 2003] and Spambase. [UC Irvine Machine Learning Repository, 2013] I use these data to build classifiers for “twitter speak, ” “medical speak,” quant speak”, “news speak”, “music speak”, ”anatomy speak” and “spam speak.”

This provides us with a simple but scalable and parallelizable set of tools to explore the entropy of the social web: semantic lexica, tags annotated with Rényi entropies, the “stickiness” between tags as calculated by mutual information, an “discriminating tags” classifier, and a calculus that allows us to combine our classifiers in a mathematically consistent way.



Graphic 2: An Architecture of Participation

The Information in a Tweet

The notion of using entropy as a measure of system state and dynamics comes both from statistical physics and from information theory. In statistical physics, entropy is a measure of disorder and uncertainty in a random variable; the higher the entropy, the greater the disorder. [Gray, 1990] In this context, the term usually refers to Gibbs entropy (Equation 1), which measures the macroscopic state of the system as defined by a distribution of atoms and molecules in a thermodynamic system. Gibbs entropy is a measure of the disorder in the arrangements of its particles. As the position of a particle becomes less predictable, the entropy increases.

Gibbs entropy

$$S = k\ln(N) = -k\sum_i p_i \log p_i \quad (1)$$

In information theory, entropy is also a measure of the uncertainty in a random variable. [Behara, Krickeberg and Wolfowitz, 1973] In this context, however, the term usually refers to the Shannon entropy (Equation 2), which quantifies the expected value of the information contained in a message (or the expected value of the information of the probability distribution). The concept was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication." [Shannon, 1948] Shannon entropy establishes the limits to possible data compression and channel capacity. That is, the entropy gives a lower bound for the efficiency of an encoding scheme (in other words, a lower bound on the possible compression of a data stream). Typically this is expressed in the number of ‘bits’ or ‘nats’ that are required to encode a given message.

*Shannon entropy**

$$E(Text) = \sum_i p_i \log p_i \quad (2)$$

*p_i is 1/frequency of tag i

For our notion of the "information of a tweet" I need to introduce three additional ideas besides entropy itself: 1) the entropy rate; 2) ergodicity; and 3) stationary stochastic processes. The entropy rate (or mean entropy rate) describes the limiting entropy over an entire probability distribution. This can be thought of as the average entropy over a sufficiently long realization of a stochastic process, whereas the entropy is relevant to a single random variable at a given point in time.

In statistics, ergodicity describes a random process wherein the average time for one sufficiently long realization of events is the same as the ensemble average. That is, the ensemble's statistical properties (such as its mean or entropy) can be deduced from a single, sufficiently long sample of the process. In other words, there are long-term invariant measures that describe the asymptotic properties of the underlying probability distribution, and they can be measured by following any single reprehensive portion if followed long enough. By "sufficiently long" I mean the sample mean converges to the true mean of the signal. For example, if I look at two particles in an ergodic system at any time, those particles may have very different states; but if I follow those particles long enough, they become statistically indistinguishable from one another. This means that statistical properties of the entire system can be deduced from a single sample of the process if followed for a sufficiently long time.

Stationarity is the property of a random process which guarantees that the aggregate statistical properties of the probability density function, such as the mean value, its moments and variance, remain the same at every point in time. A stationary process, therefore, is one whose probability distribution is the same at all times. Its statistical properties cannot necessarily be deduced from a single sample of the process. There are stochastic processes that exhibit both stationarity and ergodicity called *stationary ergodic* processes. These are random processes that will not change their statistical properties with time; hence, the properties, including the disorder (entropy) of the system, can be deduced from a single, sufficiently long sample realization of the process. There are weaker forms of the stationary condition in which the first- and second-order moments (that is, the mean and variance) of a stochastic process are constant but other properties of the probability density function can vary. Likewise, there are stationary stochastic processes that are not themselves ergodic but are composed of a mixture of ergodic components.

The ergodic notion that a sufficiently long sample of a single process represents the process as a whole allows us to use invariant measures to estimate not only the amount of information in a given tweet but the kind of information as well. How entropy can be used to classify tweets is best explained with a concrete example.

For example, let’s say I want to classify a tweet to by language. The entropy of many languages has been determined. English has 1.65 bits per word, French has 3.02 bits per word, German has 1.08 bits per word, and Spanish has 1.97 bits per word. Given the probability density function of word entropies and the average bits per word of a single tweet I could then assign probabilities that it is English, French, German, or Spanish.

Why use entropy to classify tweets by language? There are a number of techniques for text classification [Hastie, Tibshirani and Friedman, 2001][Murphy, 2012], including,: support vector machines (SVMs) , naive Bayes, latent semantic indexing , topic modeling , and artificial neural networks. In short, language classification techniques are already relatively numerous and quite good, so why add another? Here are some advantages to using entropy for classifying languages:

a. Unsupervised – No training data is required.
b. Efficient - O(n log n) in time where n is the size of the tag lexica.
c. Precise - The tags are reasonable and informative.
d. Accurate - Is not often missing tags that should be there.
e. Parallelizable -Any number of entropy frequency distributions can be used in a single pass over the text.
f. Has Explicit Tagging (using Rényi entropies)
g. Has Implicit Tagging (using mutual information)
h. Has Word Sense Disambiguation (using mutual information)
i. Generates Entropy Signatures (using Kullback-Leibler divergence)
j. Allows an infinite numbers of states to be classified using the same simple calculations.

That said, perhaps the single greatest advantage is a) above: the simplicity, scalability and consistency of nearly all entropy calculations. The basic algorithm is to find tags in text, look up entropies, and then sum. They take the form of – *log (p)* where p is typically a frequency count. We’ll go through the various entropies that I use and their differences at the bottom of this section, but they are all simple calculations. Most of the work involves pre-computing the counts for all of the features (in our case tags) used. Once that is done, determining the entropy involves retrieving a pre-computed value for each tag in a tweet and adding them up.

One may note that the computation process is similar to a naive Bayes classifier or simply counting how many times a set of keywords of interest occurs in text. So why use entropy? By using word entropy, like probabilistic approaches, the weight the keywords are adjusted more appropriately than by simply counting. Unlike probabilistic approaches there is a lot of evidence that the use of words in natural language is an ergodic process. Researchers have shown entropy rate constancy in text [Genzel and Charniak, 2002], ergodic signatures in tagging distributions [Kontoyiannis, 1996], and even a complexity and entropy of literary styles [Plotkin and Nowak, 2000]. These ergodic natural signatures provide an additional dimension that help detect language features such as keyword spamming even when the spammers are not using "spam words." Information theoretic approaches have the simplicity of keyword counting; the keyword weight adjustment and can detect not only the words used but the style in which they are used through complexity and entropy of natural language.

Of course, the language of a tweet may not be the only entropic feature of system state. It is quite possible that language is used differently in tweets according whether one is child or adult, movie star or scientist, angry or happy, or thousands of other variations in state. In the next section, "Adverts, Small Talk, Big Words, and Newspeak" we'll expand on this idea in detail for two states that I believe are particularly important for encouraging scientific collaboration.

It is quite likely that classifiers optimized to a particular task are more accurate than using general information-theoretic measures for the same task. For example, support vector machines classify the language of text quite well. This extra precision, however, almost always comes at a cost of increased computational complexity. Support vector machines don't scale well to "big data." In this case, therefore, a hybrid approach would make sense. Those tweets whose information signatures are almost certainly English are classified as English, but those whose signatures are as likely to be German as English can be handed off to more computationally expensive approaches for further analysis.

Unlike most supervised classification approaches, the use of information-theoretic measures may discover novel hidden states. This is because I can easily visualize clusters when measuring the entropy tweet by tweet or article by article. For example, when determining the information signatures of children versus adults, I find three (or more) distinct clusters and eventually find that the unknown cluster corresponds to teenagers.

One of the most intriguing aspects of using information-theoretic measures is that the same calculations can be used to create an infinite set of classifiers. For example, let's say I want classify a tweet as "hungry" or "not hungry." I could have a model in which I believe that people include tags that are food items (such as *coffee*, *pizza*, or *ice cream*) in their tweets when they're hungry. There are a couple of ways in which this classifier could be created: a) the entropies of only the food tags could be stored as a distribution ,with the entropies of the non-food tags being set to zero; or b) the tags in the tweet could be filtered for food tags before calculating the entropy of the tweet. Binary classifiers like "hungry" or "not hungry" would then check the difference between the tweet's "food entropy" and zero. Further, certain terms such as "I'm full" might indicate the "not hungry" state, in which case they might negatively contribute to the "hunger entropy." The precision and recall of the classifier could then be evaluated if one had a sufficiently large set of tweets that were validated as reflecting hunger.

Alternatively, if one had a set of tweets reflecting hunger, I could use an extension of a supervised Bayesian technique developed by Mosteller and Wallace to find which tags best discriminated the state "hungry" from the state "not hungry." [Mosteller and Wallace, 1984] Mosteller and Wallace answered the historical problem of who wrote each of the disputed Federalist papers—Madison or Hamilton—by finding which words were most distinctive to Madison and Hamilton respectively and using those as the basis of their classifier. We'll discuss this in more detail in the section "Classifying an Infinite Set of States." Other supervised approaches that generate a set of words, such as naïve Bayes, or our "discriminating tags" could also be used.

A Calculus of Classification

To calculate the “information of a tweet,” one need to decide which of the many entropies to use as well as any other heuristics that can help estimate the information content in text. One also needs some way of adding, subtracting and scaling the information estimates. For example, how can one add lists of informative tags to create a search kernel that would filter for computational biology related text that is low on chit-chat and spam? Before discussing our calculus for estimating the information of a tweet” I need to summarize of the entropies that I use and their differences. I chose the following information-theoretic measures: plog, Shannon entropy, Rényi entropy, Hartley entropy, collision entropy, min-entropy, Kullback-Leibler divergence and the information dimension.

Plog

Plog (pronounced ‘plog, ’ for positive log) (Equation 3) is simply the negative log of the frequency. As the value of plog increases, the frequency decreases.

*Plog**

$$E(Text) = -\sum_i \ln p_i \qquad (3)$$

*p_i is 1/frequency of tag i

frequency	plog (base 2)
0.5	1
0.25	2
0.125	3
1/16	5
1/1024	10

1/1,048,576	20

Big plog means low frequency.

Rényi entropies

The Rényi entropies (Equation 4) generalize the Shannon entropy, the Hartley entropy, the min-entropy, and the collision entropy. As such, these entropies as an ensemble are often called the Rényi entropies (or the Rényi entropy, even though this usually refers to a class of entropies). The difference between these entropies is in the respective value for each of an order parameter called alpha: the values of alpha are greater than or equal to zero but cannot equal one. The Renyi entropy ordering is related to the underlying probability distributions and allows more probable events to be weighted more heavily. As alpha approaches zero, the Rényi entropy increasingly weighs all possible events more equally, regardless of their probabilities. A higher alpha (α) weighs more probable events more heavily. The base used to calculate entropies is usually base 2 or Euler's number base *e*. If the base of the logarithm is 2, then the uncertainty is measured in ‘bits’. If it is the natural logarithm, then the unit is ‘nats’.

*Rényi entropies**

$$E(Text) = \frac{-1}{(\alpha - 1)} \log \sum_i p_i^{\alpha} \quad (4)$$

*p_i is 1/frequency of tag i

Hartley entropy

The Hartley entropy [Gray, 1990] (Equation 5) is the Rényi entropy with an alpha (α) of zero.

Hartley entropy*

$$E(Text) = -\log |X| \quad (5)$$

*p_i is 1/frequency of tag i

Shannon entropy

The Shannon entropy [Gray, 1990] (Equation 6) is the Rényi entropy with an alpha (α) of one. The Shannon entropy is a simple estimate of the expected value of the information contained in a message. It assumes independence and identically distributed random variables, which is a simplification when applied to word counts. In this sense it is analogous to naïve Bayes, in that it is very commonly used and thought to work well in spite of violating some assumptions upon which it is based.

*Shannon entropy**

$$E(Text) = -\sum_i p_i \log p_i \quad (6)$$

*p_i is 1/frequency of tag i

collision entropy

The collision entropy [Gray, 1990] (Equation 7) is the Rényi entropy with an alpha of two and is sometimes just called "Rényi entropy."

collision entropy*

$$E(Text) = -\log \sum_i p_i^2 \quad (7)$$

*p_i is 1/frequency of tag i

min-entropy

The min-entropy [Gray, 1990] (Equation 8) is the Rényi entropy as the limit of alpha (α) approaches infinity. The name min-entropy stems from the fact that it is the smallest entropy measure in the Rényi family of entropies.

min-entropy*

$$E(Text) = -\log(\max(\ p_i\ )) \quad (8)$$

*p_i is 1/frequency of tag i

Kullback-Leibler divergence

Kullback-Leibler divergence [Gray, 1990] (Equation 9) is a non-symmetric measure of the difference between two probability distributions. The Kullback-Leibler measure goes by several names: relative entropy, discrimination information, Kullback-Leibler (KL) number, directed divergence, informational divergence, and cross entropy. Kullback-Leibler divergence is a measure of the difference between the observed entropy and its excepted entropy. I calculate the KL divergence by weighting one distribution (like an observed frequency distribution) by the log of probabilities of some other distribution D_2 (Equation 9)

KL divergence*

$$\sum_{i=1} D_1(p_i) \log \frac{D_1(p_j)}{D_2(p_i)} \qquad (9)$$

*p_i and p_j are 1/frequency of a

tag in texts i and j.

Kullback-Leibler divergence is useful to identify tags whose distributions (and therefore entropies) differ across two distributions. For example, KL divergence can be used to evaluate the cross-entropy of tags between Twitter and PubMed.

Mutual Information

Mutual information [Gray, 1990] quantifies the mutual dependence of the two random variables. It is a measure of the "stickiness" between two items. It measures how much knowing one of these variables reduces uncertainty about the other. I can use mutual information to quantify the association between two tags. Mutual information (Equation 10) is given by:

$$MI(I,J)=\sum_{i=1}^{m}\sum_{j=1}^{n}p_{ij}\log\left(\frac{p_{ij}}{p_{i.}\cdot p_{j}}\right) \quad (10)$$

$MI(I,J)$, the mutual information between two tags I and J, is calculated using p_i is 1/frequency of tag I, p_j is 1/frequency of tag j and p_{ij} is 1/co-occurrence frequency of tags I and J.

Weighted Information Content Parameterization

A search kernel is generated from a list of tags, to “add” kernels, I simply combine lists. If one wants to weigh individual tags “information content” (IC) or a list of tags this scaling is easily done by the properties of logs. The sum the log n times is the same as n times the log, so I can use the properties of logs to scale. However, “subtracting” information content is tricky as IC estimates are based in the entropy equation listed above and the minimum entropy is zero.

To understand our approach to subtract information, let’s revisit our example of adding positive “quant speak,” and “medical speak” word entropies while subtracting negative “spam,” and “chit-chat” entropies to create a search kernel that would filter for computational biology related text that is low on chit-chat and spam.” The simplest way to do this would be to have negative weights for the undesirable tags. However, doing so would violate basic properties of entropy. The entropy value E is non-negative. The minimum possible entropy value is zero corresponding to the case in which one event is certain (Equation 11):

$$E_{\min}=1\cdot\ln\left(\frac{1}{1}\right)=0 \quad (11)$$

When all states are equally probable ($p_i=\frac{1}{n}$), the entropy value is maximum (Equation 12):

$$E_{\max} = \sum_{i=1}^{n} \frac{1}{n} \ln(n) = n \frac{1}{n} \ln(n) = \ln(n) \quad (12)$$

A proof the minimum and maximum values for entropy is given by Theil in *Statistical Decomposition Analysis*. [Theil, 1972] Negative weights would allow us to have "negative" entropy. While this might be acceptable if these calculations still allow us to create simple, efficient, scalable, and precise classifiers it would violate the three basic properties of entropy: 1) a minimum value of zero, 2) a maximum value of log (n) and 3) monotonicity. Instead I treat "negative" entropy as if it is increasing the overall uncertainty. That is, I view the undesirable entropy as increasing the number of states n. The "negative" entropy then increases n to a new number of states n_{pseudo} which equals the original n plus an estimate of the increase in uncertainty $n_{negative}$ ($n_{pseudo} = n + n_{negative}$). I detail our justification for "negative" entropy in the paper "A Calculus of Classification" [Brown, 2013c].

Unsupervised Tagging

Our development of "discriminating tags" was the our experience with standard topic modeling algorithms as implemented in the R packages 'tm' and 'lda' were not suited to our research. Specifically, these modeling algorithms:

a. Use eigen-value decomposition on dense matrices, which is O(n^3) in time
b. Tag articles with words that have low marginal information like "I,""new","Wednesday".
c. Don't scale to big data.
d. Don't handle polysemy (word disambiguation)
e. Don't infer topics not explicitly in the text but are implied from context.

Discriminating tags [Brown, 2013b] has properties well suited to our research. Specifically, discriminating tags is:

a. Unsupervised – No training data is required.
b. Efficient - O(n log n) in time where n is the size of the tag lexica.
c. Precise - The tags are reasonable and informative.
d. Accurate - Is not often missing tags that should be there.

e. Parallelizable -Any number of entropy frequency distributions can be used in a single pass over the text.
f. Has Explicit Tagging (using Rényi entropies)
g. Has Implicit Tagging (using mutual information)
h. Has Word Sense Disambiguation (using mutual information)
i. Generates Entropy Signatures (using Kullback-Leibler divergence)

Given a tag set with associated frequency distributions, the discriminating tags approach is summarized as follows:

1. Explicit Tag Algorithm ("explicit tagging")

   a. Filter the tags set by an information threshold to create an unsupervised tag list using an information theoretic measure such as one of the uses the Rényi entropies (Shannon entropy, collision entropy, min-entropy) or plog

   b. For each article/tweet/web page find tags matching the unsupervised tags and their counts.)

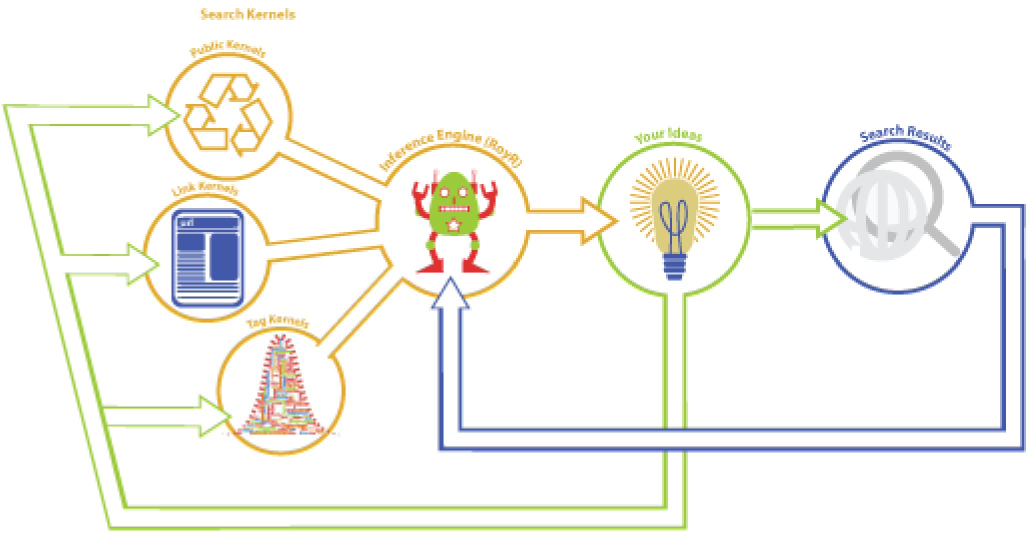
   c. Keep those tags above a count threshold. These called the explicit tags (ET).

2. Implicit Tag Algorithm

   a. For every tag in a tag set calculate a set of associated tags. Any reasonable association measure and threshold can be used.
   b. Create an associative array between a tag and its associated tags.
   c. For each tag in the explicit tags append list of associated tags using the associative array. This expanded list is the putative tags.
   d. Create an associative array to keep track of duplicate tags in the putative tag list.
   e. For each distinct putative tag calculate the sum of the mutual information between it and each of the explicit tags. An associative array is used to skip over replicate putative tags.
   f. Keep those putative tags above a mutual information threshold. These are the implicit tags (IT).

Roll your own Rank (RyoR)

The use of word tag entropy allows users to create search kernels that they can share, tweak and re-mix. As discussed in section in the section and publication "A Calculus of Classification"; the use of entropic search classifiers can be exploited to remix these kernels and classifiers. A classifier can be created in at least three ways: 1) creating a tag list, called a "tag-bag", 2) by presenting a list of links or text of interest and using "discriminating tags" to automatically generate a tag-bag from the training set or 3) tweaking and re-mixing existing tag-bags and search kernels. A wide range of users can “roll their own rank” without a deep mathematical expertise. Users can also view the existing public search kernels on the site, clone them and tweak them allowing for a simple intuitive mechanism for aggregating search ideas. Further the use of a technique called “Interleaved Search Evaluation” [Chapelle, Joachims, Radlinski and Yue, 2012] allows users to evaluate search kernels simply by using a site.



Classifying an Infinite Set of States

As discussed in “The Information in a Tweet,” information-theoretic measures can be used to create an infinite set of classifiers for system state. For the classifiers, as noted, the basic algorithm is to find tags in text, look up entropies, and sum; hence using hundreds or thousands of measures of state is feasible, even on “big data.” The primary difference between the classifiers is the set of discriminating tags used and the thresholds used for binning. Rule-based inference engines with many rules can easily be built simply by creating a search kernel that weights positively the entropic classifiers of interest and negatively those of disinterest.

Building the state classifiers is straightforward. The first step is building a set of discriminating tags for a state. For example, if one wants to classify a tweet as “angry” then one could just generate a list of anger words such as *furious*, *berserk*, *enraged*, *fuming*, *incensed*, *infuriated*, *irate*, *livid*, *maniacal*, *outraged*, etc. one then calculate the “anger entropy” of tweets and compare that with human annotation of “angry tweets.” The primary tools I use for the evaluation are receiver operating characteristic (ROC) curves which I discuss in detail in the methods section.

Another approach to building a set of discriminating tags is when one has a data set representing the state of interest. Some may quip that one can find a large set of “angry tweets” simply by logging in to Twitter; but there may be many groups already collecting this data. For example, a large portion of tweets to the complaint department of a large organization is likely to be angry.

The approach used in this research to find a set of discriminating tags when given a training set is an extension of a supervised Bayesian technique developed by Mosteller and Wallace to determine the authorship of the Federalist Papers. The approach is detailed in the Mosteller and Wallace’s book *Applied Bayesian and Classical Inference: The Case of The Federalist Papers* [Mosteller and Wallace, 1984] as well as the in the methods section, but the basic intuition is straightforward: given a training set in which some of the classification is known, a set of “marker words” can be ranked by their low variance within a set and high variance between sets. In Mosteller and Wallace’s work, once a set of marker words was found, they were used in a Bayesian calculation to determine the likelihood of any unknown text belonging to a class (“Madison” or “Hamilton”). One difference between our approach and that of Mosteller and Wallace is that I use only tags and not all words; hence I call them discriminating tags. (The difference between a tag and a word is described in detail in the Methods section.) The second difference is that I filter our discriminating tags based on our intuition. For example, Mosteller and Wallace found that words like *enough, while, whilst,* and *upon* were marker words that discriminated between writing by Madison and writing by Hamilton. Though it makes sense that one author uses *while* and the other *whilst*, the word *enough* may be more subject- than author-dependent. In our case, if *furious, berserk, enraged,* and *upon* were determined to be discriminating tags, I might still eliminate *upon* or else further investigate how *upon* reflects a state of anger. Another approach used in this research to find a set of discriminating tags when given a training set is an to auto-tag the text using our *discriminating tags* algorithm and using the resulting tag set as a tag-bag.

Adverts, Small Talk, Big Words, and Newspeak

One particularly interesting aspect of the social web is the nature of text made available for public consumption. From the time of the Gutenberg printing press until the advent of Web 2.0, nearly all text presented in public was written by professionals. [Shirky, 2010] [Shirky, 2008] Whether it was a book, a business or government record, a sermon, a news story or opinion article, a scientific paper, or an advertisement, it was written by a professional with the intent to communicate information and/or ideas. Not until the social web and Twitter did musings about what a non-celebrity ate for breakfast or whether someone likes naps was widely available for public consumption.

Musings about one’s foot fungus or a statement like “ Hahahahahaha!!!! You should have come to NKLA!!! So many beautiful pitties! And pittie lovers....” are what I call *small talk*. Small talk is light, intimate banter, often understandable only by the authors’ close friends. A lot of the communication on the social web is small talk, even though it was very rare in public writing prior to the social web.

Adverts are messages intended to sell something: for example, “Sleek, Thin, State Of The Art And Stunning, LG's Flat Screen TVs 50% OFF!” *Big words* refer to highly edited, jargon-filled text intended to communicate ideas or information to peers, like scientific or technical papers. *Newspeak* refers to highly edited text intended to communicate ideas or information to a broad public, like news articles.

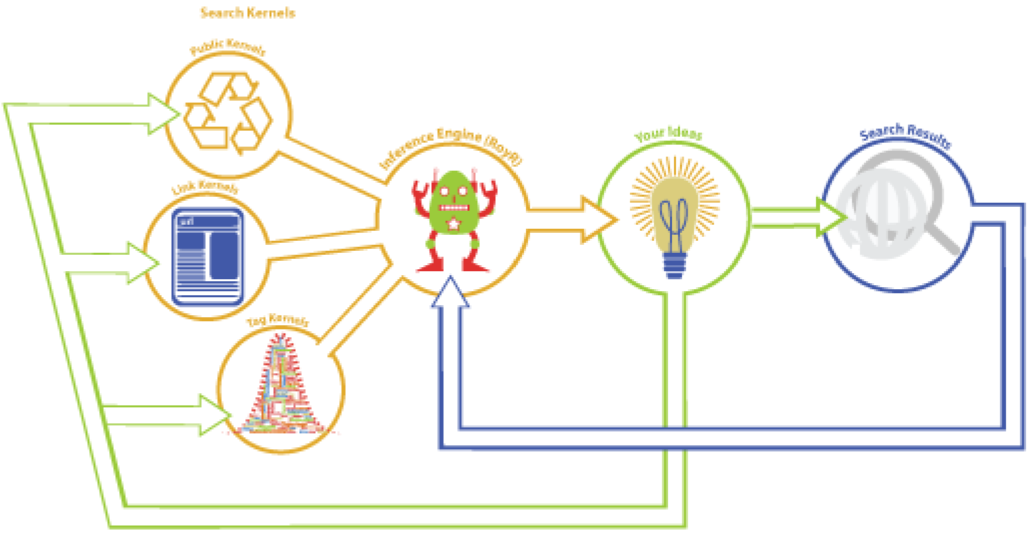
Our reason for focusing on classifying these states for the purpose of promoting participation is our belief that the nature of scientific collaboration networks primarily involves “big words” and that adverts, and small talk should be filtered and newspeak needs to come from authoritative sources. We’ll discuss the “Adverts, Small Talk, Big Words and Newspeak” model as it relates to encouraging scientific collaboration in more detail in the section "Experiments in Computational Social Biology."

Building Information-theoretic Inference Engines

The generation of hundreds of information-theoretic classifiers could easily be adapted to inductive, deductive , and/or fuzzy reasoning in its inference engine. For Inductive reasoning, the entropy calculations would need to be converted to probabilistic measures. For deductive reasoning, the entropy distributions would need to be converted to hard thresholds like “big words” or “not big words.” For fuzzy reasoning the entropy distributions would need to be converted to fuzzy set membership. In future research, I believe all of these approaches should be investigated in depth.

One can also use geometric methods as the basis of inference. One can treat a set of *n* classifiers as an *n*-dimensional Hilbert space and determine whether a given item is within one or more *n*-dimensional hypervolumes of interest using a set of *n* linear discriminants. For example, if there are two classifiers, “Big Words” and “Clinician,” I would plot the distribution of those who took and those who didn’t take an action of interest and fit a set of two linear discriminants that best fit the data using least squares. Of course, there are more sophisticated ways in which one can construct the hyperplane, such as support vector machines.

However, the purpose of this research is a proof of concept that information-theoretic measures are useful for characterizing system state and dynamics and that those measures can help in understanding the factors that influence user participation. In fact one want information-theoretic inference engines that are so intuitive that users can build their own search kernels, share them and the system will automatically evaluate them simply through their use. I discuss the details of how to do this in the section on the "Roll your own Rank” tool (RyoR). The short version of how this works is that I can use the calculus of word entropies to allow a user to create a search kernel choosing any available entropy of interest and disinterest and seeing if it gets results that they like. Users can also view the existing search kernels on the site, clone them and tweak them. I evaluate the search kernels by a technique called “Interleaved Search Evaluation”. [Chapelle et al., 2012] The idea of this approach is to interleave multiple search ranks to be “fair,” so that users’ clicks can be interpreted as unbiased judgments about the relative quality of the rankings. This way the users just need to click of results that interest them to evaluate the search kernels.



The Structure and Dynamics of Information Networks

Graph theory has its origins in the 1700s, but much of the research using it to understand the dynamics of networks has been in the past decade or so. [Watts, 2003] An aspect of this research is on the question of whether there are topological properties of networks that can be used as measures of system state. In particular, when information-theoretic measures are mixed with graph theory, they can be used to better understand the nature of interactions between two nodes. For example, people may be connected in a social network like Facebook or Twitter for a variety of reasons; they may for instance be friends, collaborators, acquaintances, political or religious co-thinkers, fans of an entertainer, or supporters of a cause. I want to use entropy to better understand the interactions so that I can create collaboration networks, friendship networks, and the like.

Our basic approach is to classify the nature of the interactions between two nodes in the same way as I measure the information of a tweet. For example, if someone likes, favorites, or re-tweets something, I calculate the entropies of the interactions that generate an entropy signature between two nodes. If those interactions are primarily "small talk," then they are part of a small talk network. If the entropy signature has two prominent modes—“intimate speak” and “big words”—then they may be connected in both friendship and collaboration networks.

Our approach is an exploratory analysis of how the use of information-theoretic measures can affect standard social-network analysis. As noted above, I chose measures that are commonly used for social network analysis and are available in open-source software the R packages ‘statnet’ and ‘tnet’.

The basic idea is to run an analysis on a full network and on the information-theoretic sub-network. For example, I calculate the influence of a node on the full network and on a collaboration sub-network to determine whether there are nodes that are influential on the sub-network but not on the network as a whole. Limiting analysis to sub-networks in this way can both improve the quality and efficiency of analysis. Because the number of potential interactions in a network grows exponentially with the number of nodes in it, effective tools to prune networks early on are very important to social network analysis.

Big Data, Efficiency and System State

“Every day, we create 2.5 quintillion bytes of data—so much that 90% of the data in the world today has been created in the last two years alone.” [IBM, 2012] Due to the growing complexity of digital social networks and the huge quantity of data they produce daily, it’s important that I deal with “big data” efficiently. A number of tools and technologies (Map-Reduce, NoSQL, Hadoop, Hive, cloud computing, parallel processing, clustering, MPP, virtualization, large grid environments, and so on) have been developed to store and process big data. While big-data technologies have established the ability to collect and process large amounts of data, most organizations struggle with understanding the data and taking advantage of its value. According to an *Economist* report: “Extracting value from big data remains elusive for many organizations. For most companies today, data are abundant and readily available, but not well used.”[Economist, 2012]

A central issue with “big data” is that not all of the data will be relevant or useful in solving a particular problem and will often add noise instead. However, the purpose of the present research is to better measure the system state of social networks, and this knowledge of state can help with a central issue:

which data and algorithms are relevant to a particular problem? As our basic algorithm is to find tags in text, look up entropies, and sum, it is easily parallelizable and scales well to "big data."

Measuring state in unstructured or semi-structured historical data is virtually identical to monitoring state in real time. The major difference between characterizing historical data and doing the same for real-time data is that I can often use more computationally intensive estimates of system state when the real-time constraint is lifted. Therefore, having classified system state in unstructured or semi-structured historical data, I can filter that data for a subset that is most relevant to a particular problem.

Stationarity and Social Analytics

Most algorithms used for machine learning assume that data has the same probability distribution at all times. [Quiñonero-Candela, 2008] However, this stationarity condition is often violated in real-world problems—particularly in dynamical systems. Whether it's appropriate to use a simple stationary model to describe a complex real-world dynamical system depends on whether that model is useful or correct. Determining whether an algorithm on the social web is useful or correct is often very difficult, as I are missing the counterfactual. I can do algorithm-versus algorithm-comparisons for a given site on a given day for a given audience, but I rarely know whether an algorithm is correct according to some "gold standard." A measurement of the system state allows us to perform a sensitivity analysis of a given algorithm on the stationarity condition without the need of a gold standard. The sensitivity analysis would require three components:

1) the classification of data on system state;

2) a set of algorithms to test; and

3) a measure of distance for comparing the output (e.g. correlation).

Given these components, then for each state X and algorithm Y, I would measure distance of output across states to find outliers. This is particularly important for the "big data" of the social web because techniques for adapting machine-learning algorithms to dynamic probability distributions such as covariate shift adaptation, weighted empirical risk minimization, weighted cross-validation, and direct importance estimation are computationally expensive.

The Entropy and Topology of Participation

As explained above, an aspect of this research is to better measure the system state of a social web. In particular, it aims to evaluate measures that reflect the dynamics of data in a social system. However, to encourage participation, I need to better understand how system state influences the adoption and quality of user-generated content. In this section, I will review what is known from the social sciences on how groups form, collaborate, deliberate, and aggregate knowledge. Then I discuss how I might measure these processes in a way that I can incorporate into social analytics.

What factors may influence adoption and quality of aggregated knowledge?

Decades of research in the social sciences tell us that the statistical aggregation of knowledge from groups often outperforms deliberating groups and individual experts. However, certain conditions must be met for the statistical aggregation of human knowledge to work well [Sunstein, 2006] [Sunstein, 2008] [Surowiecki, 2005]:

1. Diversity of opinion: A range of viewpoints is incorporated.

2. Independence: People's opinions aren't determined by the opinions of those around them.

3. Decentralization: People are able to specialize and draw on local knowledge.

4. Common Knowledge: Uniquely held information is widely dispersed.

5. Expertise: Those contributing have some knowledge of the subject.

6. Low Transaction Costs: It is easy or automatic for people to contribute.

7. Aggregation: Some mechanism exists for turning private judgments into a collective decision.

There is frequently a tradeoff amongst these conditions. If one selects for a group of only experts, for instance, one often decreases the diversity of opinion (through selection bias of who determines expertise and reduction of group size) and increases transaction costs (through the effort of finding experts). A fundamental question in the computational aggregation of human knowledge is: Who should one aggregate knowledge from for a particular topic? Should one have the largest group possible? Or a diverse group of people with some knowledge? A group of experts? Or the single “most expert” person?

A simple model that provides insight about the proper group size is Condorcet's jury theorem [Ladha, 1992]: If the probability that a voter being correct is greater than 1/2, then adding more voters increases the probability that the majority decision is correct. The implication of Condorcet's theorem is that one should add not just the experts but all those with a little relevant knowledge. The jury theorem also raises the question of how somebody can perform worse than randomly—that is, with a less than 50% likelihood of getting a yes-or-no question right. Studies indicate that when groups perform worse than a bunch of randomly guessing individuals, this is often due to a systematic bias or “groupthink.” Groupthink has many causes, including the following:

1. Amplification of Cognitive Errors: Groups have been found to amplify, rather than to attenuate sensational and recent evidence even if it is wrong.

2. Hidden Profiles: Uniquely held information is not dispersed.

3. Informational Cascades: Individuals ignore their own beliefs out of deference to popular opinions of the group.

4. Group Polarization: Members of a deliberating group often end up adopting a more extreme version of the position toward which they tended before deliberation began.

Our question is whether our information-theoretic and topological measures of system state can help visualize the properties of a social network that I have just summarized: diversity of opinion, independence, decentralization, common knowledge, expertise, low transaction costs, groupthink, amplification of cognitive errors, hidden profiles, informational cascades, and group polarization. For example, can low transaction costs be inferred from the velocity of information flow through the network? Can group polarization be seen by the clustering of measures of system state? Can expertise be inferred from the velocity of information flow in a particular domain of knowledge? Is expertise a state that can be inferred from topology or entropy?

A specific example is whether “diversity of opinion” can be inferred from the entropy signatures of those contributing to a wiki. It turns out there is quite a bit of research on how one can use entropy to create what is called a “diversity index.” [Jost, 2006] A diversity index is a quantitative measure that reflects how many different types (such as states) there are in a dataset, and simultaneously takes into account how evenly the basic entities (such as individuals) are distributed among those types. The value of a diversity index increases both when the number of types increases and when evenness of the distribution across the individuals increases. If I think about when entropy reaches its maximum value, I can see how it can be used as a measure of diversity. The maximum of the entropy function is the log of the number of possible events, and occurs when all the events are equally likely. In fact, one of the most commonly used diversity measures is to employ the Shannon entropy of a probability distribution as a diversity index. This is called Shannon's diversity index, or the Shannon-Wiener index. Actually, all the Rényi entropies are commonly used as diversity indices.

It turns out, then, that “diversity of opinion” can be inferred from the entropy quite easily. I just need to identify tags that represent the opinions of interests: tags that represent opinion 1, opinion 2, etc. For example, if I want to quantify the diversity of opinion concerning Mitt Romney versus Barack Obama, I can create lists of tags that represent Mitt Romney (Mitt Romney, Romney, Mitt, Gov. Romney) and Barack Obama (Barack Obama, Barack, Obama, POTUS, President of the United States, United States President, US President). (A group of tags that are synonyms I call “synsets,” as does Wordnet [Miller, 1995][Fellbaum, 1998].) One then compares the observed entropy with the theoretical maximum entropy to generate a diversity index in the range (0,1). If one is concerned that tags for the current US President may be far more frequent those of a presidential contender, one can use one of the higher order Rényi entropies to weigh the relative distributions more appropriately.

Once one has a diversity index, one can use it to measure “groupthink.” Groupthink can be thought of as the decrease of diversity of opinion over time. To determine whether there has been a significant change in the diversity of opinion, one can plot this decrease or one can perform an analysis of variance (ANOVA) with time-series data.

The independence of opinion can also be measured using information theory. Mutual information quantifies the mutual dependence of the two random variables. It is a measure of the “stickiness” between two items: that is, how much knowing one of these variables reduces uncertainty about the other. I can use mutual information to quantify the question of how well I can predict one person’s opinions while only knowing another person’s.

The Entropy of Sentiment

Sentiment refers to feelings and emotions. Sentiment analysis aims to determine the emotional state of a user at the time s/he contributes some content. Our approach to measuring sentiment is to find a set of discriminating tags related to sentiment and then calculate the “sentiment entropy” of tweets. However, sentiment analysis is a very active field. Hence, rather than our standard of comparing automated “sentiment tweets” with human-annotated sentiment tweets, I can compare our algorithm with other algorithms for sentiment.

An Architecture of Participation

The code for our Architecture of Participation is written primarily in Python and JavaScript and is packaged into libraries. As discussed in the preface, the “Architecture of Participation” is not just a scientific work but the open source code base that implements the methodology. It is meant to be used by Open Source developer’s not just scientists. This Architecture includes semantic lexica, algorithms, and software that allow one to incorporate the dynamics of data and the system state into social analytics. The code and data are released under the two Open Source licenses: the Apache License Version 2.0 [The Apache Software Foundation, 2013], and the Creative Commons Attribution license [Creative Commons, 2013]. These licenses allow others to use the Architecture for any purpose. They may distribute, remix, tweak, and build upon this work, even commercially, as long as the original creation is credited. The code, its documentation, licenses, example usage, and application programming interface (API) are available at http://stochasticity.org. The code is also available as a GitHub repository.

A social network and blog dedicated to measurement and visualization of qualitative differences in the dynamics of social networks is at http://stochasticity.net. Stochasticity.net's blog will focus on the "Entropy of the Social Web ", as well as on the use of and improvement of this Architecture. The Inspiration for the code and visualization for http://stochasticity.net comes from many sources. Any registered user can contribute news, discussion, bugs, links and feature suggestions. Stochasticity.net also has a bot that continually scours the net for videos, links, and articles related to information theory, entropy, the dynamics of social networks, social network analysis, measurement of system state in dynamical systems, and the nature of randomness.

Experiments in Computational Social Biology

The many successes of the social web (e.g. PageRank, BitTorrent, Digg, Flickr, Twitter, Wikipedia, Google Maps, blogs, eBay, Facebook, YouTube, del.icio.us , etc.) have led to calls for Science 2.0—the use of social media for science. [Shneiderman, 2008] Christopher Surridge, editor of PLOS ONE, believes that "Web 2.0 fits so perfectly with the way science works, it's not whether the transition will happen but how fast" [Waldrop, 2008]. David Crotty, executive editor at Cold Spring Harbor Laboratory Press, in his talk "Why Web 2.0 is failing in Biology" says:

> Most of it boils down to the tools not being well designed for the desired audience. The hype is right in some ways—the potential is there, and it is something I should be excited about, but we're failing to channel that potential into compelling tools that will catch on with the community. A successful tool will address a need of the user, and will do so in the context of the culture of the user. You're unlikely to get a well-established culture to change just to suit your tool, no matter how much promise it shows. While most of the tools available are clever ideas, or seem useful on the surface, their lack of traction should be telling you that something's not quite right. [Crotty, 2008]

A recent survey of the UK research community indicates that "most researchers use well-known generic tools such as Google Scholar (73%) and Wikipedia (69%). They also indicate that a significant minority of researchers also use other well-known social networking services such as YouTube (29%), Facebook(24%) and Twitter (10%). Overall, however, the survey indicates that use by the UK research community of Web 2.0-based services for novel forms of scholarly communication is relatively low." [Williams, 2007] The same report, however, indicated that most researchers were open to the use of the social web for research: "A small minority (14%) of non-users expressed themselves skeptical or uninterested, with the great majority (86%) either neutral or enthusiastic." The following quote sums up the report: "Despite an increasing interest in Web 2.0 as a platform and enabler for e-research, I have limited understanding of the factors influencing adoption." [Crotty, 2008]

In order to assess the possible use of the Architecture of Participation to better understand the factors influencing user contribution in the culture of biological research, I used the Architecture to test whether using these information-theoretic measures of system state and our inference engine can increase user participation. Our focus is on improving engagement for a wiki (Disiki) and a social bookmarking site (Tagic), two of the most common types of sites in social biology. These two biological social networks, 1) Disiki (a wiki), and 2) Tagic (a social bookmarking site) allowed us analyze the dynamics of social networks much smaller-scale than Twitter. The basic idea for both of these sites is to use the "Roll your own Rank" tool (RyoR), allow users to create their own or tweak the best existing search kernels and to use collective intelligence to determine which factors best promoted contributing to wiki or sharing bookmarks. I discuss the details of how to do this in the section on the "Roll your own Rank" tool (RyoR).

The wiki model, Disiki, was created In response to a September, 2008 *Nature* article, "Big Data: Wikiomics" [Waldrop, 2008]. The article expressed the idea that more mechanisms for harnessing community intelligence were needed in biology, but that "[co]ommunity intelligence is a new concept for biology" and the practical details of how to implement community intelligence to annotate big data are still very experimental. In a 2007 publication, the curators of OMIM state:

> A challenge OMIM already faces is how to catalog complex phenotypes and complex genotypes and their functional relationships to each other and to include epigenetics (and epigenomics), the interaction of genes and gene products, the interaction with and influence of environment, and

> the emergent phenotypes resulting from these interactions—no small undertaking[McKusick, 2007].

IBM’s Center for Social Software believes that Web 2.0 technologies can be particularly effective in involving those outside of academia in medical research: “The next generation of Web tools has the potential to significantly enhance our ability to understand and communicate what is happening to patients in the real world, ” says William Marder, PhD, senior vice president for research at the Healthcare business of Thomson Reuters. [IBM, 2008]

Disiki is an open-source disease encyclopedia of around 6, 000 diseases. It is intended to help researchers aggregate, share, and syndicate information likely to be useful in modeling and understanding relationships that may cause complex diseases. Currently, Disiki collects basic information about each disease (etiology, symptoms, related conditions, complications, diagnosis, environmental factors, epidemiology, genetics, prevention, prognosis, risk factors) as well as associations between the disease and genes/proteins, pathways, chemicals, and phenotypes. The major difference between Disiki and a standard disease wiki is that it uses information-theoretic and topological measures of system state in its recommendation engine and then tests whether those recommendations increase the contribution rate for wiki articles.

The social bookmarking model, Tagic, was created In response to the loss of Connotea. Tagic is a link/bookmark sharing site for BioMedicine. It is very similar to social bookmarking sites like Connotea, de.li.ci.ous, Digg, StumbleUpon, Reddit, ReadCube, CiteULike, and Papers, except that in addition to user tagging it employs an automated tagging system. Moreover, Tagic uses information-theoretic and topological measures of system state to increase the sharing rate for bookmarks.

Algorithms, Data and Methods used in the Architecture of Participation

This section is intended to provide enough detail of the algorithms, data, and methods used to build the Architecture of Participation in sufficient detail that a user can understand, extend, and employ it. Some of the methods are novel, while others are well established. For well-known methods, our focus is on how I adapt the methodology to the measurement and visualization of qualitative differences in the dynamics of social networks.

BioTags: A Semantic Lexical Database for the Social Tagging of BioMedical Text

Tagging is a process in which end users use free-form keywords to manually index content in an organic and distributed manner. The popularity of tagging has led some to claim that it is the primary classification scheme of the Internet. [Smith, 2008] A tag can be thought of as an informative keyword. A user is very unlikely to tag an article with a word like "this" because it conveys very little information. Rather, they'll often tag with a subject or sentiment.

Problems with tagging are well-known. Users often present idiosyncrasies, inaccuracies, inconsistencies, and other irregularities when tagging. Specifically, four areas are critical to tagging. The first three areas are straightforward enough:

1) tag misspelling;

2) tag heterogeneity, (that is, different tags denoting the same content, such as "Ziagen" and "abacavir sulfate," which both refer to the same drug);

3) tag polysemy (i.e. identical tags that denote different meanings, such as, Apple may refer to fruit or a company. and;

4) semantic annotation of tags (i.e. abacavir sulfate is a drug).

The fourth area, often called "semantic enrichment" is a particularly difficult problem. Lexical resources are often used to annotate terms. As Boguraev and Pustejovsky state, "In computational linguistics research, it has become clear that, regardless of a system's sophistication or breadth, its performance must be measured in large part by the computational lexicon associated with it." [Boguraev and Pustejovsky, 1996] The purpose of BioTags is to create a lexical database to help resolve issues with tag misspelling, tag heterogeneity, tag polysemy, and semantic annotation. The semantic enrichment is particularly focused on concepts related to mining biomedical text: diseases, symptoms, genes, anatomy, risk factors, genes, proteins, markers, pathways, chemicals, drugs, ligands, HLA numbers, alleles, antigens, SNPs, loci, enzymes, organisms, species, and phenotypes. In addition, I annotated the tags with frequency and entropy statistics. The calculations of entropy are discussed in their own section below. The raw tag frequency signatures and tag contingency tables are provided as a data dump so others can easily generate additional statistics that I are missing. The details of generating the frequency signatures and tag contingency tables are discussed in their own section below. Further places are annotated with its longitude and latitude, people with their gender and date of birth, and institutions with their addresses.

I built BioTags by extracting terms from one non-biomedical encyclopedia, Wikipedia [Wikipedia, 2013] and several biomedical databases and ontologies, including: DARPA Cognitive hierarchy [DARPA Cognitive hierarchy, 2013],the Foundational Model of Anatomy [Rosse and Mejino, 2003], the Colin Brain Atlas [Collins, Holmes, Peters and Evans, 1995], the BrainMap functional neuroimaging database [Laird, Lancaster and Fox, 2005], the Talairach atlas [Talairach and Tournoux, 1998], the Mai atlas [Mai, Assheuer and Paxinos, 1997], Entrez-Gene [Maglott, Ostell, Pruitt and Tatusova, 2005], NINDS [The National Institute of Neurological Disorders and Stroke Index, 2013], NIH Office of Rare Diseases Research (ORDR) [NIH Office of Rare Diseases Research (ORDR), 2013], BioLexicon [BioLexicon, 2013], UniProt [UniProt, 2013], and Gene Ontology [Gene Ontology, 2013]. I also extracted hashtags from Twitter.

I use the notion that a tag can be thought of as an informative keyword in order to distinguish tags from terms. I first eliminate the very-low-frequency terms, then calculate how important the term is to PubMed. Specifically I use tf–idf (term frequency–inverse document frequency) [Salton and McGill, 1986] to select a subset of tags from set of terms. Tf–idf is a numerical statistic that reflects how important a word is to a collection of documents. Tf–idf ranks tags by their low variance within a set and high variance between sets. The following steps summarize how tags were identified from a large text corpus:

1. Use Biological databases, Wikipedia, and Twitter hashtags to find terms.

2. Find those terms whose frequency is above a threshold exist in over 11 million PubMed articles.
3. Calculate tf–idf for each frequent term across 11 million PubMed articles.
4. Select those above a tf–idf threshold.

Once the tag set was identified a number of heuristics were developed to annotate the tags. A primary basis of tag annotation is the source itself. For example, a tag coming from the NIH Office of Rare Diseases would be annotated with the meta-tag “disease” and a tag coming from Foundational Model of Anatomy would be annotated with the meta-tag “anatomy.” Another source annotation of a tag is the structure of its Wikipedia article. For example, if the “info-box” of the article has a longitude and latitude, I can infer that the tag is a “place” in addition to knowing the quantitative values of its longitude and latitude.

WordNet is another source of tag annotation. WordNet is a lexical database of English nuns, verbs, adjectives, and adverbs that are grouped into sets of cognitive synonyms (synsets). The WordNet synsets are further characterized by hyperonymy, hyponymy, or ISA relationships. Tags found in WordNet were annotated with their synonym relationships. Finally, tags with annotation have allowed us to develop heuristics to exploit patterns in the structure of words (prefixes, suffixes, and roots) to annotate additional tags. For example, disease tags often end with the words “syndrome,” “deficiency,” or “disease.” Likewise, tags that represent small molecules often have prefixes like “alkyl-“ or suffixes like “–mide”, “-dehyde” or “–hol”. The statistics and entropy of a given tag were derived by counting its frequencies over various corpora. The following steps summarize how tags were annotated:

1) the source of the tag
2) its Wikipedia "info-box"
3) Wordnet synset annotation
4) word-structure annotation heuristics
5) calculation of statistics and entropy from tag counts

Finally, I exported the BioTags in wiki format, creating a wiki page for each tag/synset at BioTags.org. The intent of the wiki is to use crowdsoucing to add missing tags, flag errors, and improve annotation. The BioTags lexical database, the website, and all component lists are available at http://BioTags.org.

Further details are found in our paper ”BioTags: A Semantic Lexical Database for the Social Tagging of BioMedical Text.”

Mapping Twitter Hashtags to Words and Phrases

The social tagging of a tweet in is done by placing a hash mark in front of a word or phrase, such as #BCSM, #Lyphoma, #BrainTumorThursday, #BreastCancer, #Infertility, #Diabetes, #lymphoedema, #RareDiseaseDay, #RareDisease, #ADHD, #Anorexia, #MultipleSclerosis, #Depression, #OzDOC, or #MedEd. Finding hastags in tweets is very simple; one just looks for one-grams that begin with a hash mark. However, as tags cannot contain spaces and there is a 140 character limit a number unusual morphological change to words and phrases occur in Twitter. To map the Twitter hashtags to our tag set I created a number of mapping heuristics.

Our first mapping rule I call the Scrunched Word Hashtag Heuristic. A multiple word phrase like Rare Disease Day is scrunched in to the hashtag #RareDiseaseDay. To map these tags to our synsets in our lexical dictionary I scrunched all of our tags converted all of the tag and tweet text to lower case, and looked for exact matches with Twitter hashtags. For example, the tag Multiple Sclerosis is converted to #multiplesclerosis and matched with the text in a tweet using the same tokenization that I describe in the section “Counting Tags in Text.”

Our second mapping rule I call the Capitalized Word Phrase Hashtag Heuristic. There is a strong tendency to capitalize multiple word phrases to make them easier to read. For example, it’s more common to see the tag #BreastCancer, or #BrainTumorThursday included in a tweet rather than #breastcancer, or #braintumorthursday. These tags become a source of novel tags for our lexical dictionary. If a tag like #BreastCancer already exists then I skip it . If a tag like #BrainTumorThursday doesn’t exist then I add spaces before each capital letter and convert it to lower case;

#BrainTumorThursday becomes “brain tumor thursday” and a candidate for a novel tag. Standard collocation tests described in detail in Chapter 5 of Chris Manning and Hinrich Schütze’s book, Foundations of Statistical Natural Language Processing, [Manning and Schuetze, 1999] are used to determine if that phrase in used in a corpora other than Twitter (i.e. PubMed , Common Crawl [Common Crawl, 2013], and Wikipedia). If it passes the collocation test then the tag is added to the lexical dictionary and the semantic, syntactic and statistical annotation described in the section “BioTags: A Semantic Lexical Database for the Social Tagging of BioMedical Text” is performed. If a novel tag is very common in Twitter but cannot be validated in another corpora then it is added as a “stub,” and users of BioTags are encouraged to annotate it.

Counting Tags in Text

Tags are counted by loading a lexical dictionary, usually BioTags, in to a hash table. The lexical dictionary has separate entries for case, spelling variations and mis-spelling of tags. For example, Color, color, coluor and colour would all have separate entries but would belong to the same synset. These various tags would be counted separate but then aggregated into the synset for color. A flag can be set to ignore case in which the text for the tags would be converted to lower case before inserting in to the hash and the text to be counted would be converted to lower case before tag frequencies were determined. The python counting script has two more optional parameters for a frequency threshold and a check rate. For example, if I set my frequency threshold to 5 and my check rate to 1,000,000 then the script would check that a tag occurred at least 5 times in 1,000,000 entities (e.g. tweets, abstracts, webpages, etc.), then at 2,000,000 entities the tag would be checked to see if it occurred at least 10 times and so on. Those tags that didn’t meet a threshold are eliminated from the hash table reducing its size.

Individual entities of text (e.g. tweets, abstracts, webpages, etc.) use a local hash for counting. Some punctuation is converted to white space using three regular expressions. The first regular expression '[\.][ ]+' gets converted to ' ,.' . (single white space, period, single white space). The second regular expression '[\,][ ]+' gets converted to ' , ' . (single white space, comma, single white space). These two regular expressions - implemented as the single regular expression [\.|\,][ ]+' to ' , ' - allows us to remove commas and periods that end words but keep words with commas and periods like 3,7-Dihydro-1,3,7-trimethyl-1H-purine-2,6-dione. The last regular expression r'[_+:;=!@$%^&*\"\?\/]' to ‘ ‘ allows us to remove punctuation that can interfere with tag matching. Once punctuation is converted to white space the text is tokenized by white space and all n-grams of to five grams are generated. A local dictionary is used to count the ngrams that match tags in the lexical dictionary. Those local tag counts are added to the overall tag counts after the text is processed.

Twitter Data

I wrote python scripts to access both the Using the Twitter Search API [Twitter Search API, 2013] and the Twitter Streaming APIs. [Twitter Streaming APIs, 2013] I used these API’s to extract Tweet data, Twitter user profiles and friends & follower relationships between Twitter users. I parsed the tweets for retweets, urls and hastags using regular expressions. I saved all of the data provided by Twitter for tweets and users and as specified Twitter REST API v1.1. I extracted millions of records of tweet and user profile data and stored them in a MySQL database. Tag frequencies for Twitter were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. The raw tag counts were then used to calculate information theoretic measures.

PubMed Data

I download over 11 million PubMed records for the National Library of Medicine (NLM) as XML over ftp. This represented Medline data through December 2012. I wrote a python script that converted the XML files to tab delimited text files. I keep the PubMed id, title, abstract, MeSH terms, and author list. All subsequent counting and text processing was done using the tab delimited text files. Tag frequencies for PubMed were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. . The raw tag counts were then used to calculate information theoretic measures.

Wikipedia Data

I downloaded the April 4th, 2013 English Wikipedia XML dump file (pages-articles.xml.bz2). [Wikipedia Data, 2013] The file has the current revisions only, with no talk or user pages. The size of the

April 4th, 2013 dump is 9 GB compressed, 42 GB uncompressed . I wrote python scripts to convert the XML in to tab delimited text files and converted the Wiki formatting to a standard formatting. I also wrote python scripts to find those articles with "Info-boxes" and extract that Info-box data in to individual data fields associated with the article. Tag frequencies for Wikipedia were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. . The raw tag counts were then used to calculate information theoretic measures.

Wordnet Data

WordNet is a lexical database of English nuns, verbs, adjectives and adverbs which are grouped into sets of cognitive synonyms (synsets). The WordNet synsets are further characterized by hyperonymy, hyponymy or ISA relationships. I downloaded the WordNet database files and parsed them. Permission to use, copy, modify and distribute WordNet for any purpose and without fee or royalty is hereby granted, WordNet provided by WordNet as long as proper attribution is given to WordNet and any derivative products don't use the WordNet trademark.

Common Crawl Data

Common Crawl is a openly accessible web crawl data that is freely available. As of April 2013 the crawl has 6 billion pages and associated metadata. The crawl data is stored on Amazon's Public Data Sets, allowing it to be directly accessed for map-reduce processing in EC2. To process Common Crawl I set up a Hadoop cluster on Amazon EC2. I then ran various MapReduce jobs to process the content, with python scripts and then transferred the results to S3 buckets within the Amazon cloud. Tag frequencies for Common Crawl were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. The raw tag counts were then used to calculate information theoretic measures.

Crawling the Web

Our web crawler consists of python scripts and has four main aspects: 1) A crawler - a tool that fetches urls and feeds and extracts the text and links. 2) The indexer – a tool that creates an index of the text given to it so that patterns can be found in the text. 3) The ranker – a tool that allows one to apply algorithms and heuristics to the index to classify the text. 4) Garbage collection that flags spam, stale and possibly malicious site.

For the crawler I wrote a set of python scripts and MySQL database fetch and store urls. Urls that have not been visited are selected randomly. be crawled are selected have a content type of 'text/html', and have no url error codes. The order of the selection is by url priority. When more than one crawler is used the urls are randomly assigned to a number of tranches that equals the number of crawlers.

An attempt to fetch the web page is then made. If a DNS error is returned this information is database and the next url is then crawled. If a web server error is returned the crawler checks whether it has server authentication for the url and if it does and second attempt is made with the server authentication information. If either on the first or second attempt a web server error is still returned this information is database and the next url is then crawled. Our crawler also has Url authentication capabilities; both for server and browser authentication. Urls that require passwords are typically run as a separate robot process that is checking each url for the user, password and type of authentication. The reason that “authentication needed” and “no authentication” robots are run as separate processes is that a very small percentage of web pages require authentication and I only have credentials for a few. Forcing a robot to check authentication creates unnecessary overhead.

Our indexer extracts links, creates link relationships, extracts tags, and counts tags once given the html of a web page. It also coverts “tiny urls” like http://bit.ly/12h5UfB and variant forms of urls like http://clipartist.net/, http://clipartist.net/index.php, http://www.clipartist.net/, canonical standard of the form http://clipartist.net. The canonical standard url I call a “slug” and the indexer uses it to keep track of the occurrence of a url in its various forms across webpages and tweets.

Our ranker calculates the entropy of the text for the url. It calculates the plog, Kullback-Liebler divergence and the Rényi entropies (Hartley entropy, Shannon entropy, collision entropy, min-entropy) based on our four major corpora: 1) PubMed, 2) Twitter, 3) Common Crawl and 4) Wikipedia. Entropies for individual urls as well as entropies for the domain are calculated. These entropies, a domains or pages Google PageRank or a random selection can be used by the robot to determine the crawl order.

Our garbage collection does not delete data or urls but rather flags urls as potentially problematic. The three criteria I use to flag a url or domain is: 1) "badware" 2) spam 3) poor response time. I determine "badware" (usually web-based malware/virusus) by a list of over a million domains kept by the StopBadware not for profit organization. [StopBadware, 2013] More details on badware can be found at the StopBadware (https://www.stopbadware.org) site in their publications section. Spam is determined determining the "spam entropy" text. Response time is recorded through a measurement of the socket time when fetching a url.

Creating Term-Document Matrices from Text

Small term-document matrices are created using the R text mining package 'tm'. For processing larger amounts of text, tag counts are generated as described in the section "Counting Tags in Text." However rather than summing the tag counts over the individual entities of text (e.g. tweets, abstracts, webpages, etc.) the tag-count data for each entity is first stored in associative array, denoted A. The mapper emits key-value pairs with an entity id as the key and the corresponding associative array of tag counts as values. The source already has an entity id like a tweet id or a PubMed id then that id is used otherwise a unique id is created.

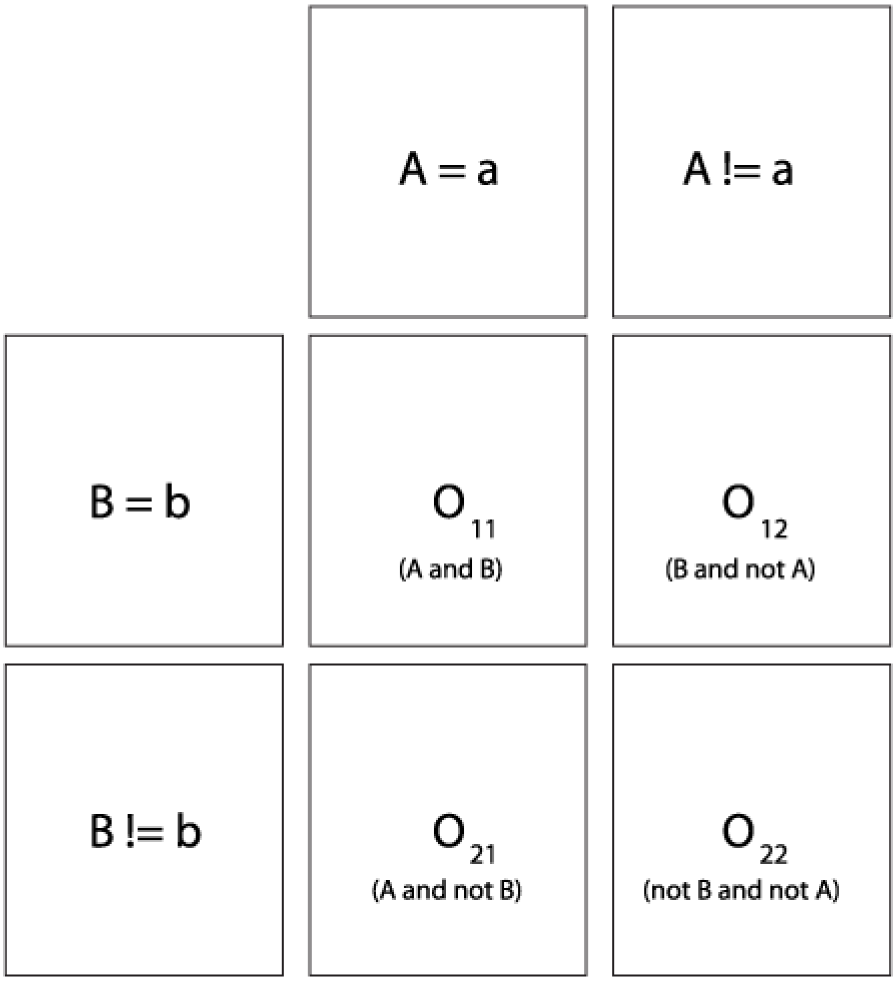
Most of our processing uses the Bag-of-words model in which the text of a document is represented as an unordered collection of words. [Youngjoong, 2012] As such, the reducer can take the output of the mapper without further processing as I already have an entity id as the key and the corresponding associative array (Bag-of-words) as its model. When a full n-tag by n-tag term-document matrix is needed the reducer fills in zeros for all missing tags. For large Hadoop jobs both the "pairs" and "stripes" MapReduce algorithms to count co-occurrence as described in "Data-Intensive Text Processing with MapReduce" [Lin, Dyer and Hirst 2010] was used.

Creating Tag Co-occurrence Matrices

For small amounts of text the tag co-occurrence matrix was created using the same Bag-of-words mapper that emits key-value pairs with an entity id as the key and the corresponding associative array of tag counts as values as described in the section "Creating Term-Document Matrices from Text."

For larger data I used a “frequency signature” approach to convert the Bag-of-words output to a format that I can use to calculate tag co-occurrence associations and mutual information. Frequency signatures are described in detail in Stefan Evert’s PhD dissertation “The Statistics of Word Cooccurrences Word Pairs and Collocations.” [Evert, 2004]

To calculate tag co-occurrence associations and mutual information for two tags, A and B, I need four items of data. The co-occurrence count of A and B, the count of A but not B, the count of B but not A, and the total number of tags in a corpus. This co-occurrence frequency data for a word pair (A,B) are usually organized in a contingency table show below. The contingency table stores the observed frequenciesO_{11} … O_{22}. The table below (adapted from Evert’s dissertation) shows an observed contingency table.



	A = a	A != a
B = b	O_{11} (A and B)	O_{12} (B and not A)
B != b	O_{21} (A and not B)	O_{22} (not B and not A)

Contingency Table

Contingency table : O_{11} is co-occurrence count of A and B, O_{12} is the count of A but not B, O_{21} is the count of B but not A, and O_{22} is the count of not B and not A.

However, while the co-occurrence count of A and B, and the total number of tags in a corpus are efficiently and easily counted the count of A but not B, the count of B but not A are tricky and computationally expensive. The insight and advantage of frequency signatures is that they calculate the count of A but not B, the count of B but not A by just counting A and B and the co-occurrence count of A and B. That is, the count of A but not B is equal to count of A minus the co-occurrence count of A and B. Likewise, the count of B but not A is equal to count of B minus the co-occurrence count of A and B.

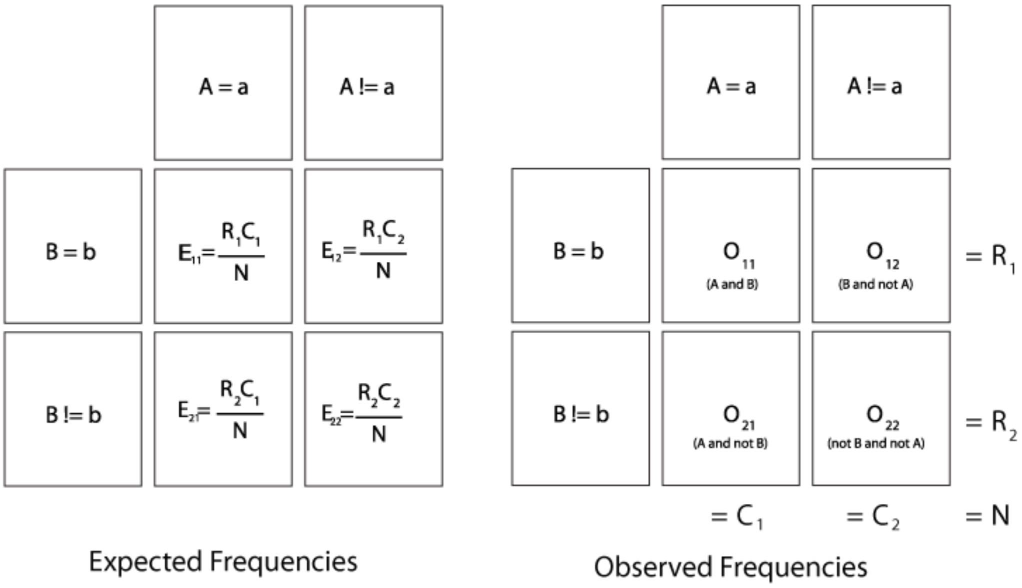
The frequency signature of a tag pair (A, B) is usually written as (f, f1, f2,N). Where f is the co-occurrence count of A and B, f1 is the count of A but not B, f2 is the count of B but not A, and N is the total counts. Notice that the observed frequencies O_{11}, ..., O_{22} can be directly calculated from the frequency signature by the equations below:

1. $O_{11} = f$
2. $O_{12} = f1 - f$
3. $O_{21} = f2 - f$
4. $O_{22} = N - f1 - f2 + f$

Generating all of the data tag co-occurrence association and mutual information calculations using this approach can be generated using a single pass of the data and two associative arrays; one of the tag counts and another for the tag co-occurrence counts.

Calculating Associations and Mutual Information from Frequency Signatures

Evert shows the many association and mutual information statistics can be calculated from the observed frequencies O_{11}, ..., O_{22} if I can generate the expected frequencies E_{11}, ..., E_{22}. [Evert, 2004] The table below (adapted from Evert's dissertation) shows the expected versus observed contingency tables.



	A = a	A != a
B = b	$E_{11}=\frac{R_1C_1}{N}$	$E_{12}=\frac{R_1C_2}{N}$
B != b	$E_{21}=\frac{R_2C_1}{N}$	$E_{22}=\frac{R_2C_2}{N}$

Expected Frequencies

	A = a	A != a	
B = b	O_{11} (A and B)	O_{12} (B and not A)	$= R_1$
B != b	O_{21} (A and not B)	O_{22} (not B and not A)	$= R_2$
	$= C_1$	$= C_2$	$= N$

Observed Frequencies

The sum of all four observed frequencies (called the sample size N) is equal to the total number of pair tokens extracted from the corpus. R_1 and R_2 are the row totals of the observed contingency table, while C_1 and C_2 are the corresponding column totals. The expected frequencies can be directly calculated from observed frequencies O_{11}, ..., O_{22} by the equations below:

a. $R_1 = O_{11}+O_{12}$
b. $R_2 = O_{21}+O_{22}$
c. $C_1 = O_{11}+O_{21}$
d. $C_2 = O_{12} + O_{22}$
e. $N = O_{11}+O_{12}+O_{12} + O_{22}$

Evert went on to show that several association measures can be easily calculated once one has the expected and observed contingency tables. For example, the pointwise mutual information (MI) is calculated by (Equation 15) below.

pointwise mutual information $$MI = \log(\frac{O_{11}}{E_{11}}) \qquad (15)$$

The Likelihood measures that can be calculated using the expected and observed contingency tables are: multinomial-likelihood, binomial-likelihood, Poisson-likelihood, the Poisson-Stirling approximation, and hypergeometric-likelihood. The exact hypothesis tests that can be calculated using the expected and observed contingency tables are: binomial test, Poisson test, and Fisher's exact test. The asymptotic hypothesis tests that can be calculated using the expected and observed contingency tables are: z-score, Yates' continuity correction, t-score (which compares O_{11} and E_{11} as random variates), Pearson's chi-squared test, and Dunning's log-likelihood (a likelihood ratio test). The measures from information theory that can be calculated using the expected and observed contingency tables are: MI (mutual information, mu-value), logarithmic odds-ratio logarithmic relative-risk, Liddell's difference of proportions, MS (minimum sensitivity), gmean (geometric mean) coefficient, Dice coefficient (aka. "mutual expectation"), Jaccard coefficient, ,MIconf (a confidence-interval estimate for the mu-value), MI (pointwise mutual information), local-MI (contribution to average MI of all co-occurrences), average-MI (average MI between indicator variables).

Stefan Evert also developed a Perl library called UCS toolkit [Evert, 2013] for the statistical analysis of co-occurrence data with association measures and their evaluation in a collocation extraction task. I chose to rewrite these measures in python rather than use Perl to maintain consistency with all of our other code.

LDA

To implement latent Dirichlet allocation (LDA) and related models I used the R Package ‘lda’. In LDA, each document may be viewed as a mixture of various topics. This is similar to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior. The pLSA model is equivalent to the LDA model under a uniform Dirichlet prior distribution, therefore I only use LDA for our tagging comparisons and not pLSA and LDA. Details of its usage are provided in its online documentation.

Topic Modeling

To implement topic models I used the R Package ‘topicmodel’. The R Package topicmodels has tools for fitting topic models. The fittted model can be used to estimate the similarity between documents as well as between a set of keywords using an additional layer of latent variables which are referred to as topics. Details of its usage are provided in its online documentation.

Receiver operating characteristic (ROC) curves

I used R Package pROC to display and analyze ROC curves. The R Package pROC is a set of tools to visualize, smooth and compare receiver operating characteristic (ROC curves). (Partial) area under the curve (AUC) can be compared with statistical tests based on U-statistics or bootstrap. Confidence intervals can be computed for (p)AUC or ROC curve.

The Unsupervised Classification and Tagging of Free Text

The classification of free text is a fundamental task for information retrieval. Every day, we create 2.5 quintillion bytes of data, much of that text from social networks. Twitter alone produces 12 terabytes of Tweets each day. The automated tagging of text in the era of “big data” require methods that are not only of high precision and recall but are also unsupervised, efficient, scalable and parallelizable. To this end I created discriminating tags, an unsupervised efficient, scalable and parallelizable algorithm to classify and tag biomedical text based on the Rényi entropies of tags and the mutual information between tags. The algorithm classifies by generating two types of tags:1) Explicit tags and 2) Implicit tags. Explicit tags are the “most informative” keywords that explicitly occur in the text as based on estimates of their Rényi entropies. Implicit tags are keywords that should have occurred as estimated by context but did not. For example, I might tag an article with the keyword “Protein Interaction” when that term is not in the text because other keywords such as “Two-Hybrid”, “Binding Site” and “Protein Domain” imply its existence.

Implicit tags are determined by expanding the explicit tags set with a set of associated tags or by matching the explicit tags with a predetermined set of related tags that I call a “tag-bag” to create a set of putative implicit tags. An estimation of the mutual information between a putative implicit tag and the set of explicit tags is used to determine whether to tag some text with that implicit tag. I compared discriminating tags to other unsupervised topic modeling algorithms such as: latent Dirichlet allocation (LDA), latent semantic indexing, independent component analysis, probabilistic latent semantic indexing, and non-negative matrix factorization. I validated the quality our approach by generating receiver operating characteristic (ROC) curves to analyze the precision and recall of recovering PubMed MeSH tags as compared to common topic modeling algorithms. I also ran the different approaches on increasing large datasets to compare the scaling properties of the various topic modeling algorithms.

The first algorithm to extract the explicit tags assumes that one is using a set of tags and not words. This could be a set of empirically validated tags from a social network or be generated using tf-idf . Further it assumes that the tags have been annotated with at least one of the Rényi entropies or plog as described in the section Entropy of Tags.

Explicit Tag Algorithm

Given a tag set with associated entropies the Explicit Tag Algorithm is as follows:

1. Filter the tags set by an entropy threshold to create a discriminating tag list.
2. For each article/tweet/web page find tags matching the discriminating tags and their counts.
3. For each tag with at least one count divide the discriminating tag count by number of tags in text.
4. Keep those tags above a threshold. These called the explicit tags.

The second algorithm uses the explicit tags to create a list of putative implicit tags then uses then checks each tag in the expanded list for its “stickiness” with the explicit tags using mutual information.

Implicit Tag Algorithm

1. For every tag in a tag set calculate a set of associated tags. Any reasonable association measure and threshold can be used.

2. Create an associative array between a tag and its associated tags.
3. For each tag in the explicit tags append list of associated tags using the associative array. This expanded list is the putative tags.
4. Create an associative array to keep track of duplicate tags in the putative tag list.
5. For each distinct putative tag calculate the sum of the mutual information between it and each of the explicit tags. An associative array is used to skip over replicate putative tags.
6. Keep those putative tags above a mutual information threshold. These called the implicit tags.

Please note that some explicit tags may be removed if they don't meet the mutual information threshold.

The third algorithm using "tag-bags" is identical to the Implicit Tag Algorithm above except it looks up predefined putative tag lists called "tag-bags" rather than create them by concatenating the associated tag lists.

Entropy of Tags

As discussed earlier, entropy is also a measure of the uncertainty in a random variable. The discussion of how word entropy is calculated are in the sections: "The Information in a Tweet," and "A Calculus of Classification" and detailed in the papers "The Unsupervised Classification and Tagging of Free Text" and "A Calculus of Classification."

The Topological Measures

As discussed in the section The Structure and Dynamics of Information Networks graph theory are the primary tools for social network analysis. Graph theory based algorithms, while wonderful, are typically computationally expensive. In a world that generates 2.5 quintillion bytes of data a day - much of that coming from social networks – even small differences in computational cost can be critical. Our methods demonstrate that information-theoretic techniques complement topological approaches to social network analysis by measuring aspects of state that are difficult for graph-based techniques as well as for pruning graphs; as a result, more computationally expensive approaches focus on the essential structure in a network. Using information-theoretic measures to quantify the dynamics and state within a social network provides a simple, general, and computationally efficient model that scales well to "big data."

I use a consistent approach for all of our graph based social network analysis. First I prune a network using information theoretic classification; then perform a standard graph based social network analysis with the pruned network and full network and compare the analysis. The questions I are asking is how much of essential structure in a network is maintained in the pruned analysis and how did it affect the computational cost?

The tools that are using are the R and python. I use the R social network analysis packages tnet, statnet, and igraph. I use the python network analysis library NetworkX. The pruned and full networks are generated using python scripts.

Visualization

I use the javascript package D3 , R graphics and the python library matplotlib for visualization. The following are the graphics that are used in the Architecture of Participation code:

a) Entropy Donuts (Pie Charts); b) Multi-Series Line Chart; c) Tag Clouds; d) Force-Directed Graphs and e) Histograms

Details of these charts can be found on the example pages for the javascript package D3 and the python library matplotlib.

3D Visualization

I use the 3D modeling and animation packages Autodesk Maya [Autodesk Maya, 2013] and SideFX Houdini [SideFX Houdini, 2013] for 3D Visualization.

Creating tag-bags

A "tag-bag" is just a set of tags. They are the basis of all of the entropy calculations. For example, if one wanted to create "sentiment entropy" one could compile a list of "sentiment words." That list of sentiment words would be considered a tag-bag. Alternatively, if one data that represented sentiment such as love letters, complaints, hate mail, and extract the most informative tags from it to create a sentiment list. Once created a tag-bag can always be further edited to remove bad tags and add missing tags.

All tag-bags for this research are created by extracting the most informative tags from existing data. The tag extraction procedure that I use has five steps:

1. Count the frequency signatures of n-grams matching terms in a lexical dictionary for a corpora.
2. Calculate the td-idf of the matching terms. and keep those above a threshold.
3. Calculate the plog of the matching terms.
4. Keep the terms that are above both thresholds. These terms are called tags.
5. The resulting tag set is called a tag-bag.

I use a number of datasets to create tag-bags. Tag bags were created using the above procedure with the following datasets:

a. Twitter: I created a tag-bag for “twitter speak” using a random sample of a million tweets.

b. PubMed: I created a tag-bag for “medical speak” using a random sample of a million PubMed articles.

c. PubMed Journals: I created a tag-bags that represent sub-disciplines biomedicine using discipline specific journals. For example, I used the Journal of Biochemistry to generate “biochemistry speak”, the journal Genetics to generate “genetics speak”, etc.

d. arXiv: I used a bulk download of 29,000 physics, mathematics, computer science, quantitative biology, quantitative finance and statistics arXiv articles for “Quant speak.”

e. BBC news: I used a bulk download 2225 news articles for “BBC news speak.”

f. AG's corpus: I used a random sample of AG's corpus of over one million news articles for news speak.”

g. Hu and Liu sentiment words: I used a list of positive and negative opinion words for English (around 6800 words) for our sentiment tag-bag.

h. Last.fm Music Tags: I used the Last.fm Music tags as music tag-bag.

i. The Foundational Model of Anatomy ontology (FMA): I used the FMA ontology as an anatomy tag-bag.

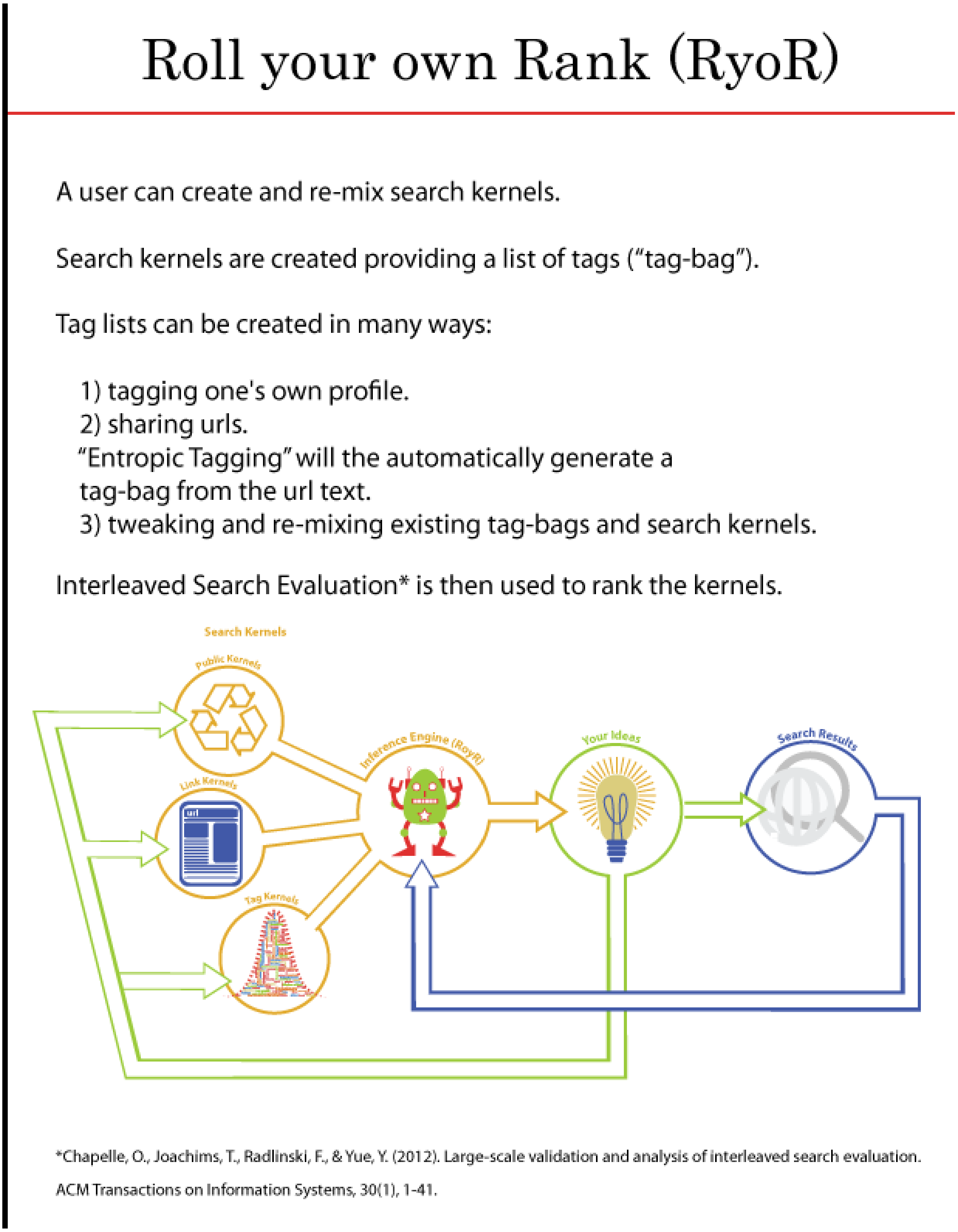
j. Spambase: Spambase is a collection of spam e-mails which a diverse set of individuals who had filed as “spam.” I used Spambase to create our spam tag-bag.

There are thousands of other datasets on the web one can use to create tag-bags.

Roll your Own Search Engine

A strength of this system is that users can easily “Roll their Own Search Engine.” A user simply needs to choose which entropies they think are relevant and what weights to give those entropies. For example, if someone felt relevant content should be medical and quantitative but not sentimental or spammy then they could choose the PubMed and arXiv based etropies with positive weights and sentiment word and spam entropies with negative weights and an entropy (e.g. the Shannon entropy). The inference engine would then calculate rank R of an item I using the set of weights w and set of entropies e.

If a user likes the search results then they can save the search kernel. Multiple search kernels can be compared using the interleaving algorithm described in the section Building Information Theoretic Inference Engines. The Roll your Own Search Engine flow is shown in the diagram below:



Roll your own Rank (RyoR)

A user can create and re-mix search kernels.

Search kernels are created providing a list of tags (“tag-bag”).

Tag lists can be created in many ways:

1) tagging one's own profile.
2) sharing urls.
“Entropic Tagging” will the automatically generate a
tag-bag from the url text.
3) tweaking and re-mixing existing tag-bags and search kernels.

Interleaved Search Evaluation* is then used to rank the kernels.

*Chapelle, O., Joachims, T., Radlinski, F., & Yue, Y. (2012). Large-scale validation and analysis of interleaved search evaluation. ACM Transactions on Information Systems, 30(1), 1-41.

Web Frameworks for Disiki and Tagic

I use Amazon web services, python, linux, apache and the django web framework to build our “experiments in computational social biology” Disiki and Tagic as well as stochasticity.net.

Architecture of Participation Evaluation

BioTags

I created a semantic lexical database for the social tagging of biomedical text. I extended the MeSH keyword set by several hundred thousand keywords. Unlike a controlled vocabulary, such as MeSH, where a user must use one of the controlled terms to annotate text, social tagging applications allow a user to classify with any tag. Even if those tags are misspelt or two different tags refer to the same concept. The primary purpose of BioTags is to provide a lexical resource to correct for tag misspelling and tag heterogeneity. The secondary purpose of BioTags is to provide a lexical resource to help disambiguate tag polysemy (i.e. identical tags that denote different meanings). BioTags provides some semantic enrichment of tags. Our semantic enrichment has focused on annotation relevant to information retrieval in biomedical test. That is annotating terms as diseases, symptoms, genes, anatomy, risk factors, genes, proteins, markers, pathways, chemicals, drugs, ligands, HLA numbers, alleles, antigens, SNPs, loci, enzymes, organisms, species, and phenotypes.

These heuristics extracted 474,165 tag synsets from 11 million PubMed abstracts. This resource is case and punctuation sensitive so “benefit-cost analysis”,”benefit cost analysis”, “Benefit-cost analysis” would ne three separate entities in the same synset tags. A synset has a display term (the most common tag in a synset) and its spelling variants and synonyms. For example, the tag with the display term “Australian frog” belongs to the synset “Litoria aurea”, “Australian frog” or “Green and Golden Bell Frog.”

The potential semantic enrichment is unlimited. For example, that’s say a researcher wants tags associated with cognitive processes (e.g. abstract thought, abductive reasoning, word recognition, spatial working memory, puzzle solving, nonverbal intelligence) and BioTags doesn’t have this semantic class. I put BioTags in a wiki so a list of cognitive processes tags could be started and the community can enhance it. The wiki also allows the collective intelligence of the community to correct mistakes, remove tags and add tags.

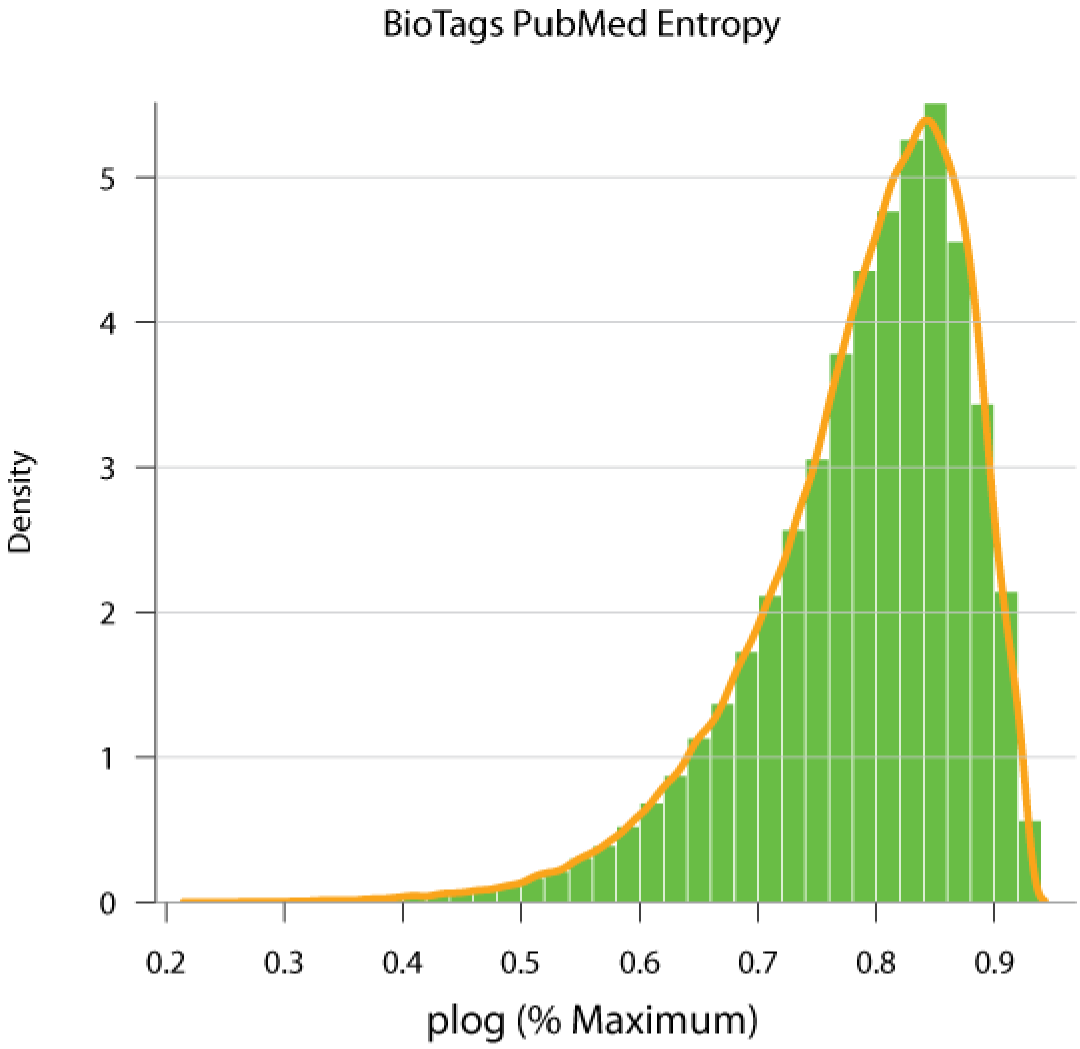
BioTags is currently a useful resource for developing social tagging applications, information retrieval and text mining but certainly incomplete. For example, even though I annotate tags as “drugs,” I don’t have subclasses of drugs such as antidepressants, antipyschotics, antibacterials, pain killers, etc. While “cartilage disease”, and “cartilage diseases” are mapped as spelling variants they aren’t mapped as synonyms of “disease of cartilage” or “disease of the cartilage” by our heuristics. As discussed in the methods I randomly sampled tags and checked them versus existing databases to estimate precision. The precision of the tag extraction was 91.2%.

The precision of the spelling variants was 98.3%. I used list inspection to help guesstimate the recall of our spelling variant and synonym matching and believe the recall to be low. Specifically I seemed to have issues in a number of areas. First, I missed tags that were acronyms are mixed words. For example, I found “omega-3 Fas”,”omega-3 FA”, and “omega-3-FA” were all spelling variants but did not match these with “omega-3 Fatty Acids.”

I missed tags in which the synonyms were reordering of words. For example, “cartilage disease”, and “cartilage diseases” are mapped as spelling variants but they aren’t mapped as synonyms of “disease of cartilage” or “disease of the cartilage.”

The precision of the semantic annotation was 95.7%. Again I feel the semantic enrichment was accurate but very incomplete. Our annotation, while useful, is not very deep. For example, I annotate tags as “drugs,” but I don’t have subclasses of drugs such as antidepressants, antipsychotics, antibacterials, pain killers, etc.

The BioTags lexical database, website and all of its component lists are available at http://BioTags.org. Further directions for this resource include writing robots that use BioTags to query popular social tagging sites like Twitter, del.icio.us, reddit, digg, flickr, StumbleUpon etc. to generate statistics on how these tags are used on the web and further cleaning, refining and expanding the annotation heuristics. I also aren’t sure if the Wiki format is ideal for getting researchers to help semantically annotate these tags and will look to build or explore other tools that might do this.



plog distribution for 474,165 BioTags (1,211,701,821 total tag counts). The histogram shows that Biotags is composed of primarily medium to high entropy tags.

Explicit discriminating tags

I randomly sampled 119,995 PubMed abstracts and calculated the exact matches with 973,586 MeSH tags. The low entropy *explicit discriminating tags* generated 4,185,771 tags of which 664, 329 of the 973,586 MeSH tags were recovered for a recall rate of 68%.

A random sample of 1,000 the additional tags were evaluated of which 853 were found to be relevant by hand for a recall rate of 85%. The relevance of the additional tags is subjective and best explained with some concrete examples. The tables below show the PMID (PubMed ID), Title (of the abstract), MeSH Tags, Low Entropy (*explicit discriminating tag with an entropy of greater than 25% of maximum entropy*) and Medium Entropy (*explicit discriminating tag with an entropy of greater than 50% of maximum entropy*)

PMID	13258677
Title	the effect of hysterectomy on ovarian function in the rabbit.
MeSH	Uterus;Ovary
Low Entropy	uterus;ovary;hysterectomy;ovarian
Medium Entropy	

PMID	13258702
Title	hyperinsulinism and premenstrual tension: report of a case of hyperplasia of the islets of langerhans.
MeSH	Menstruation;Hyperinsulinism
Low Entropy	tension;hyperplasia;menstruation;premenstrual;langerhans;report case;islets;hyperinsulinism
Medium Entropy	menstruation;premenstrual;report case;hyperinsulinism report case

PMID	13258683
Title	elective induction of labor using pituitrin; an evaluation of routine elective induction on a private obstetrical service.
MeSH	Labor, Induced
Low Entropy	service;private;labor;elective;routine;induction
Medium Entropy	

PMID	13258681
Title	the midforceps operation; preliminary study of 351 cases

MeSH	Delivery, Obstetric
Low Entropy	delivery;operation;obstetric;preliminary study;preliminary
Medium Entropy	preliminary study

PMID	13258684
Title	intravenous ethyl alcohol analgesia with intravenous pitocin induction of labor.
MeSH	Labor, Induced;Oxytocin;Anesthesia and Analgesia;Ethanol
Low Entropy	ethyl alcohol;anesthesia;labor;anesthesia and analgesia;ethanol;analgesia;ethyl;induction;intravenous;alcohol
Medium Entropy	ethyl alcohol;anesthesia and analgesia;ethyl

PMID	13258696
Title	plasma cholinesterase activity in normal pregnance and in eclamptogenic toxemias.
MeSH	Pregnancy;Pre-Eclampsia;Blood;Cholinesterases
Low Entropy	cholinesterases;normal;eclampsia;plasma cholinesterase;cholinesterase activity;pregnancy;plasma;cholinesterase
Medium Entropy	cholinesterases;eclampsia;plasma cholinesterase;cholinesterase activity;cholinesterase

PMID	13258698
Title	cold-knife conization and residual preinvasive carcinoma of the cervix.
MeSH	NO MeSH Tags
Low Entropy	cold knife;preinvasive carcinoma;knife;preinvasive;conization
Medium Entropy	preinvasive carcinoma

PMID	13258704
Title	lutein cysts in normal twin pregnancy leading to erroneous diagnosis of hydatid mole; a case report.

MeSH	Pregnancy, Multiple;Hydatidiform Mole
Low Entropy	cysts;lutein;twin pregnancy;hydatidiform mole;normal;case report;hydatid;twin;erroneous;diagnosis;pregnancy;hydatidiform;mole
Medium Entropy	twin pregnancy;hydatidiform mole;hydatid;erroneous;hydatidiform;mole

PMID	13258701
Title	an instrument for self-examination of the cervix.
MeSH	Gynecology
Low Entropy	gynecology;instrument;self;cervix
Medium Entropy	Gynecology

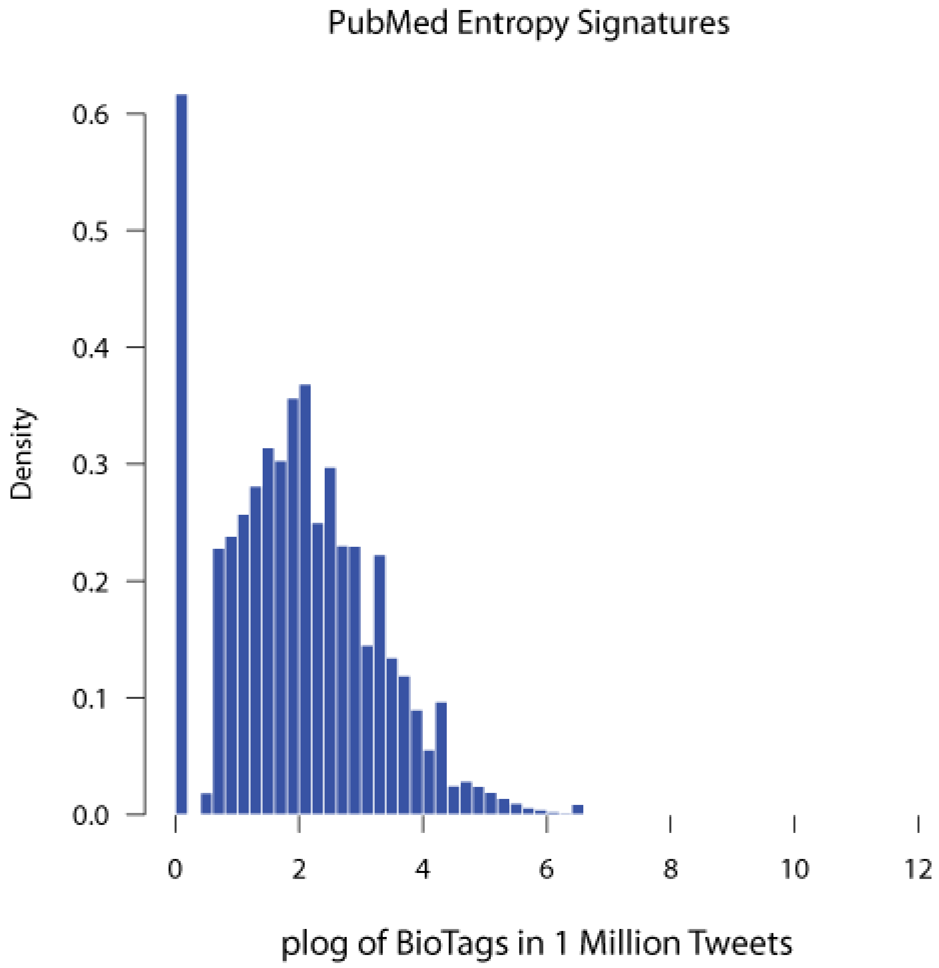
Nearly all of the errors are generated from errors in the lexical dictionary. That is, tags like "delivery" and "preliminary" are considered non-informative and scored as errors. Whereas the tag "preliminary study" is considered informative and a useful tag to add to the MeSH term "Delivery, Obstetric." The failure to recover "Delivery, Obstetric" is due to the tag "obstetric delivery" being missing from the BioTags lexica. Our inspection guesstimates that the recall rate of recovering MeSH tags could be improved by a few percent by improving the base lexica. In PMID 13258696 "plasma cholinesterase activity in normal pregnance and in eclamptogenic toxemias" the MeSH terms were pregnancy, pre-eclampsia, blood and cholinesterases. The discriminating tags missed pregnancy and blood which could be recovered if the counts and entropy were used rather than just the entropy of the tag. We'll explore these weighting adjustments in future work.

Complexity analysis

The explicit tag algorithm requires a single pass over n tags. Each step requiring a ln(n) hash look up. The memory required is a single associative array for a tag set of size n.

Classification Using Entropy

I ran the classification analysis on one million tweets gathered from the Twitter streaming API. To see if I could tag tweets. As the histogram below shows using the Biotags lexicon filters the million tweets by generating plog of zero for about 60% of one million tweets. I found for the noisy small text of Twitter the ergodic signatures acted like a filter rather than a classifier.



Biotags lexicon generates plog of zero for about 60% of one million tweets.

Inspections of the tweets with entropy zero by hand indicate that zero entropy tweets are very rarely relevant to biology or medicine.

The tables below show the results of using BioTags to classify 1 million tweets. The classifier selects tweets with relevant words but doesn't handle keyword spam well. For example, "sniffing glue sniffing glue sniffing glue," "Wobble base wobble base, " and "Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice" does very well. Further a tweet like "...Itai...itai..." is probably not referring to Itai-itai (or ouch-ouch) bone mineral disease. From our results, I view Twitter classification using entropy as more of a Twitter filter than a classifier

Using BioTags to Classify 1 Million Tweets. The tables below show the Hartley entropy and the tweet.

Tweet	**plog**
Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice	12.255
#quoteyourteacher sniffing glue sniffing glue sniffing glue sniffing glue	12.08
Wobble base wobble base wobble wobble wobble.	10.159
cof cof cof tuber cof cof cof culose cof cof cof	10.12
...Itai...itai...	10.083
MOU MOU MOU(8)	9.747
Stochastic stochastic stochastic	9.656
RT @ellliekime: Wobble base wobble base wobble wobble wobble.	9.143
Stratum corneum. Stratum lucidum. Stratum granulosum. Stratum spinosum. Stratum germinativum. #5LayersOfTheEpidermis #LEARNING	8.968
Anomic aphasia. Global aphasia. Isolation aphasia. Broca's aphasia. Wernicke's aphasia.	8.768
Erythroblastosis fetalis! Hydrops fetalis!	8.749
wavelet... oh wavelet..	8.736
Causes of Boutonniere Deformity: Causes of Boutonniere Deformity:	8.701
@eminemguevara electroencephalogram, magnetic resonance imaging,magnetic resonan angiography...=))	8.673
Atrial septal defects ..	8.66

Christmas Carol for Obsessive Compulsive Disorder ---Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells...	8.646
Fine needle aspiration biopsy	8.626
@DaveLaidig ordinal logistic regression/binary logistic regression.	8.616

Tweet	**plog**
RT @Barrowice _Nature_ editorial dashes alarmist hopes of linking extreme weather events to global... http://t.co/5SrQZIa0 #extremeweather	4.734
RT @grbeaton_psf 7/10 of worlds fastest growing economies in Sub-Saharan Africa in next 5 years | http://t.co/C7aOMtBM | PSF opportunities+	4.734
im an endothermic reaction rn. Whos the EXOTHERMIC HERE then	4.734
Neck dissection through a facelift incision http://t.co/Ljy9r7Fq	4.734
#Job - Postdoctoral Fellows Positions -- Cancer Nanomedicines and Phage Nanobiotechnology : Auburn University http://t.co/3zAwsJrq #HigherEd	4.734
Wildfowl Infectious mononucleosis: Inevitable Sweeping Recognition cream Whiz-Mutu regard-The-Snicker at Infor: .eIP	4.734
Incident 14/10/12 18:17 Light unit on fire in store room - Kettering	4.734
Professional jealousy amongst religious scholars is one of the major causes of dis unity amongst believers	4.734
Humor is to life what shock absorbers are to automobiles.	4.734
Branched Chain Amino Acids BCAAs http://t.co/8Rg9iZoN	4.734
3D Rendering Programmer / Ubisoft Reflections / Newcastle upon Tyne, Tyne and Wear, United Ki... http://t.co/bQeYrZth See @GamesJobBoard	4.734
@MissGendered chamber music? sonnets? somersaults? scuba diving?	4.734
SATURNUS: ""Saturn In Ascension"" RELEASE DATE, TRACKLISTING ANNOUNCED: Danish funeral doom warriors SATURNUS hav... http://t.co/dqRd0hH9	4.734
BMC Bioinformatics | Abstract | Accuracy of RNA-Seq and its ...: RNA-Seq is becoming common tech... http://t.co/2csYwZMn #biocodershub	4.734
Ammonium,Hydroxide and Nitrate are from valency 1	4.734
I uploaded a @YouTube video http://t.co/OEnqEVo8 kinky /curly /RE-twist using aloe vera gel	4.734
@toddstrade I scratched it. Making oil free buckwheat pancakes	4.734

HEMA timee.. sdh lama ga makan Dutch cuisine.. :)	4.734

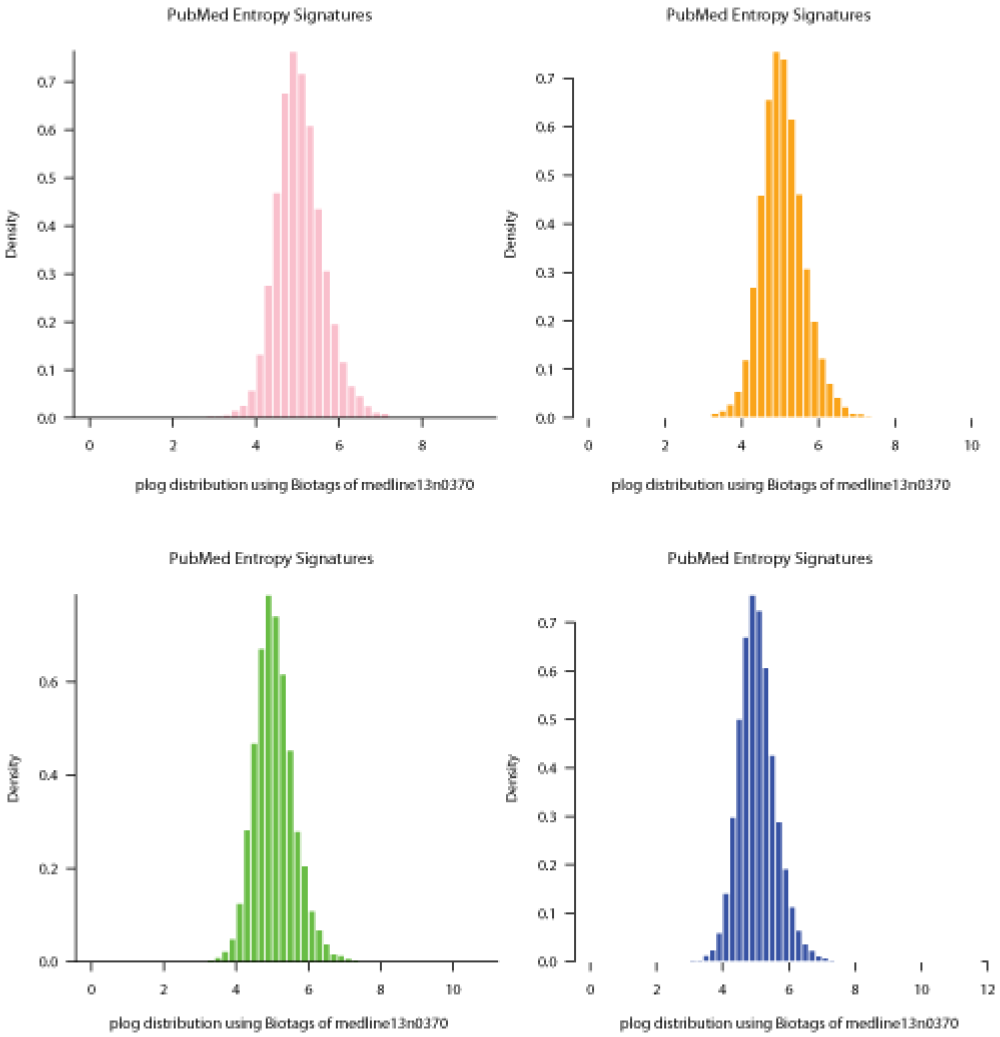
Tweet	**plog**
RT @frackinfrell: That Republicans seem to want to unravel 100 years of social progress so that corporations can sell off America. #ThingsThatOffendObama	2.613
Hurricane Sandys Breakout Star: Bloombergs Sign Language Interpreter via @digg http://t.co/VxC0dBEj	2.613
Awh. Geee I love Corbin &lt;333 thanks footlong. LOL	2.613
RT @JohnQuiggin: Dylan Byers joins the war on probability theory http://t.co/VFiJetaf v @POLITICO	2.613
RT @MackMcHaleACN: That's Romney's fiscal policy@ninatypewriter: Romney's taxes ""He claimed 47% of Americans as dependents."" ~ Weekend ...	2.613
What's your opinion on crowd-funding? A new era of democratically funded projects and altruism? A short-lived... http://t.co/YqNLNfN4	2.613
news- 'Jesus Wife' Papyrus Deemed a Fake http://t.co/nl2heRZ5	2.613
'Nexus 2: The Gods Awaken' Crowd Funding: http://t.co/5RuIIyjn	2.613
Hiii it's KDD just seeing how's it going dahhling humans	2.613
In other news though, Dexter's martingale has come #yey	2.613
@Breathedreamgo I hear oregano oil ia helpful for preventing it. How ling are you there?	2.613
Dems going to use filibuster to flatten McConnell in 2014. GOPTea fear the utterance because they know it is toxic. #GOP	2.613
2570-80 Foxfield Road - Twin Class A Medical office Buildings offering individual units as well as executive suites.. http://t.co/lvG6O5ZF	2.613
Guys are trying to propagate the cell line #labtalk	2.613
RT @gabeinformatics GRC: CDC27 example of gene with several hundred paralogs missing from reference! Show to be problematic for exome...	2.613
Also the woman opposite is knitting. KNITTING! It's like 1979. She can use a needle to perform a tracheotomy on me when I keel over.	2.613
Great point about same sex couples raising children: http://t.co/QLC0tsLe http://t.co/Nh0WU8fp	2.613
@Amazonnewsmedia Yes :) Non evidence based support of the LCP. Doctors coming ou in droves to say "" I say so it must be right"" :)	2.613

Tweet	**plog**

New job: 3d modeling designs of cars by Shahrukhkhan512 http://t.co/NlJENLRR	0
@TimetoPlay Meon Deluxe Animation Studio #timetoplaylive	0
@P0RTERalltheway I doubt they care go to places like an animation studio maybe you ca get a job there	0
Being able to be an intern at Disney's animation studio or Pixars studio would be too AMAZING.	0
yaaaaay 3 more days of this character modeling B.S, cant wait to get into fun stuff next week :)	0
Character Modeling Lecture by Pixar Character Designer is Starting! @ Tokyo Big Sight	0
I'm listening to Friend Killer by Delly Ranx on #CloudPlayer http://t.co/t683mXvc via @amazonmp3	0
RT @cartoonsmart: Job posting! CartoonSmart is always seeking awesome teachers for Actionscript 3, AS3, iOS programming, Flash, Dreamweaver, etc. Let me know.	0
Job posting! CartoonSmart is always seeking awesome teachers for Actionscript 3, AS3, iOS programming, Flash, Dreamweaver, etc. Let me know.	0
iPhone programming http://t.co/UPvQjlrM http://t.co/bssfmc13	0
Digital Filmmaking Daily is out! http://t.co/gRtrhtuA Top stories today via @m24instudi @nickhealy @ccshriver	0
@ShortTompkins I are looking at adding some XNA content (especially in Windows Phone), but will also consider XBLA now - thanks!	0
New #Job: Bible Game Developer http://t.co/8YkFIfiN	0
IC Bungie speaks... on Behavioral Game Design http://t.co/qy2aZRTv #ian_crossland	0
A small real time strategy game by charleshon http://t.co/ecHX6B18	0
It's pretty awesome how much cool game programming I can do with the math I learned i highschool 8 years ago	0
blog to get you started on game programming http://t.co/D7Q0jgGy	0
AS3 BitmapData.lock for better performance discussion on Kongregate http://t.co/ZQeAEAuz	0

Short tweets with one informative word and words that repeat informative words do well with our entropy filter. Twitter has a number of characteristics that might allow us to convert our entropy filter in to a classifier. Using Kullback-Leibler divergence to test and down weight tweets that differ from the "real" biomedical tweets would prevent keyword spamming. Keeping track of a user's tweet history and making a rank adjustment based on a history of biomedical tweets might also help.

I also found a consistency in the probability distributions in PubMed text as indicated by the plog graphs below. The indicates that further work is warranted on whether there is an entropy rate constancy in text along the lines of Genzel and Charniak's work *Entropy Rate Constancy in Text*. [Genzel and Charniak, 2002]



The graphic above shows remarkably consistent plog distributions in four separate XML PubMed files. Each XML file is around 30,000 articles.

.Discussion

In this research, I show that information-theoretic measures such as the Rényi entropies (Hartley entropy, Shannon entropy, collision entropy, min-entropy), plog and mutual information can be used classify an Infinite set of states. I developed a calculus of word entropy to create search kernel's that are intuitive, mixable, precise, sharable, scalable and parallelizable. I developed discriminating tags , an unsupervised efficient, scalable and parallelizable algorithm to classify and tag biomedical text based on the Rényi entropies of tags, plogs and the mutual information between tags. I developed, BioTags, a semantic lexical database for social tagging.

These tools - semantic lexica, tags annotated with Rényi entropies, the "stickiness" between tags as calculated by mutual information, an "discriminating tags" classifier, and a calculus that allows us to combine our classifiers in a mathematically consistent way - provide us with a simple but scalable and parallelizable set of tools to explore the entropy of the social web. These information theoretic methodologies and tools allow one to measure, visualize is to the dynamics of social networks, and more effectively utilize collective intelligence. The methodology and tools developed in the research have been submitted for review in the scientific literature: 1) *BioTags: A Semantic Lexical Database for the Social Tagging of BioMedical Text* 2) *The Unsupervised Classification and Tagging of Free Text* and 3) *A Calculus of Classification*.

I have shown that information theoretic techniques are very well suited to classifying longer highly-curated text such as PubMed. For messier, shorter text such as tweets it is probably better thought of as enrichment rather than classification. Low entropy tweets are mostly irrelevant; but high entropy tweets still have more false positives than is needed for inclusion in some further downstream analysis. The direction of future work includes:

a. measuring the irregularity, volatility, periodicity and the entropy of time series.
b. identifying essential network structure using information theoretic measures.
c. identifying relevant big data using information theoretic measures.
d. visualization using information theoretic measures.
e. sensitivity analysis of machine learning algorithms using entropy.
f. entropic inference engine creation.

g. the measurement influence, sentiment, tone, spam, engagement, groupthink, share of voice, share of conversation, reach, buzz, and virality.
h. applicability to common “power user” queries. (e.g. How viral are my tweets? How influential are my tweets? Who should I be following? What is the best time to tweet? What hashtags should I add to my tweet? How can I filter chatter? How can I create buzz?
i. measurement of ergodic signatures and analysis of the complexity and entropy of literary styles.

The use of information theoretic measures to classify “big data” may turn out to be one of the most useful aspects of this work and a central focus of future work. As stated before, 90% of the data in the world today has been created in the last two years. A major issue with “big data” filtering the wheat from the chaff. The Rényi entropies (Equation 4) are very simple to use, and easily parallelizable with most the computational cost up front counting frequency signatures. These properties make them ideally suited for classifying “big data.”

*Rényi entropies**

$$E(Text) = \frac{-1}{(\alpha \text{-} 1)} \log \sum_i p_i^{\alpha} \quad (4)$$

*p_i is 1/frequency of tag i

The use of a word tag calculus to supplement social network analysis for understanding the dynamics of networks is also very intriguing for future directions of this research. The number of possible connections in a network grows exponentially with the number of nodes in it; hence, social network analysis is typically limited to small networks. The use information-theoretic techniques to complement topological approaches to social network analysis by measuring aspects of state that are difficult for graph-based techniques as well as for pruning graphs will be a central aspect of future work. This should allow computationally expensive approaches to expand the size of the networks they can handle by focusing on the essential structure in a network.

The use of a word tag calculus that allows users to create search kernels that they can share, tweak and re-mix may be the most fun aspect of future work and this works most important contribution to participation. As discussed in section on the "Roll your own Rank" tool (RyoR) and in the section and publication “A Calculus of Classification”; the use of entropic search kernels are entropies and discriminating tag classifiers provides a natural and intuitive way for users to participate in creating their own search. Since the entropic search kernels are entropies and discriminating tag classifiers are entropies sixty years of research characterizing mathematical properties of entropy can be exploited to remix these kernels and classifiers. Further, as a classifier can be created by creating a tag list (i.e. a “tag-bag”) and the kernels and classifiers can be visualized as tags a wide range of users can “roll their own rank” without a deep mathematical expertise. Users can also view the existing public search kernels on the site, clone them and tweak them allowing for a simple intuitive mechanism for aggregating search ideas. Further the use of a technique called “Interleaved Search Evaluation” allows users to evaluate search kernels simply by using a site.

The Architecture allows for the measurement of ergodic signatures and analysis of the complexity and entropy of literary styles at a “big data” scale. For decades researchers have shown entropy rate constancy in text, ergodic signatures in tagging distributions , and even a complexity and entropy of literary styles . The implication of that is that individual Twitter users should have an ergodic signature if sampled for a sufficiently long realization. Millions of Twitter users may have tweeted enough to provide a long-enough sample for an ergodic signature which the Architecture measures as part of its measurement of system state. As the Architecture continues to run we’ll make these data available to any researchers interested in the analysis of the complexity and entropy of tweets as well as analyze it ourselves.

Appendix

BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text

Nik Brown

Computer Science Department, University of California Los Angeles

nik@ucla.edu

ABSTRACT

Social tagging has rapidly become a popular practice in which users add free-form keywords to content for categorization. Social websites such as del.icio.us, Digg, Flickr, facebook, Google+ and thousands of others use social tagging to supplement search. This presents a fundamental challenge, because the effectiveness of tagging is limited by the lack of syntactic, semantic and statistical information about those tags. Lexical resources for enriching tag-bsased search are often missing terms as well as missing associated lexical, semantic and statistical information. In this paper, we build BioTags - a large lexical database of over 450,000 tags for biomedical text. BioTags annotates tags with their synonyms, alternate spellings, acronyms, frequency statistics, and semantic annotation. The frequency statistics include frequency signatures and the following tag-tag association measures: multinomial-likelihood, binomial-likelihood, Poisson-likelihood, the Poisson-Stirling approximation, and hypergeometric-likelihood,binomial test, Poisson test, Fisher's exact test, z-score, Yates' continuity correction, t-score, Pearson's chi-squared test, Dunning's log-likelihood, mutual information, logarithmic odds-ratio logarithmic relative-risk, Liddell's difference of proportions, Dice coefficient and Jaccard coefficient. The BioTags lexical database, website and all of its component lists are available at http://BioTags.org.

Keywords— Social Tagging, Semantic Web, Lexical Database, Text Mining, Information Retrieval

I. INTRODUCTION

Tagging is a process in which end users use free-form keywords to manually index content in an organic and distributed manner. The popularity of tagging has led some to claim that it is the primary classification scheme of the Internet. [1] A tag can be thought of as an informative keyword. A user is very unlikely to tag an article with a word like “this” because it conveys very little information. Rather, they’ll often tag with a subject or sentiment.

Problems with tagging are well-known. Users often present idiosyncrasies, inaccuracies, inconsistencies, and other irregularities when tagging. Specifically, four areas are critical to tagging: 1) tag misspelling; 2) tag heterogeneity, (that is, different tags denoting the same content, such as “Ziagen” and “abacavir sulfate,” which both refer to the same drug); 3) tag polysemy (i.e. identical tags that denote different meanings, such as, Apple may refer to fruit or a company. and; 4) semantic annotation of tags (i.e. abacavir sulfate is a drug).

The fourth area, often called “semantic enrichment” is a particularly difficult problem. Lexical resources are often used to annotate terms. As Boguraev and Pustejovsky state, "In computational linguistics research, it has become clear that, regardless of a system's sophistication or breadth, its performance must be measured in large part by the computational lexicon associated with it." [2] The purpose of BioTags is to create a lexical database to help resolve issues with tag misspelling, tag heterogeneity, tag polysemy, and semantic annotation. The semantic enrichment is particularly focused on concepts related to mining biomedical text: diseases, symptoms, genes, anatomy, risk factors, genes, proteins, markers, pathways, chemicals, drugs, ligands, HLA numbers, alleles, antigens, SNPs, loci, enzymes, organisms, species, and phenotypes. In addition, we annotated the tags with frequency statistics. The raw tag frequency signatures and tag contingency tables are provided as a data dump so others can easily generate additional statistics that we are missing.

I built BioTags by extracting terms from one non-biomedical encyclopedia, Wikipedia [3] and several biomedical databases and ontologies, including: DARPA Cognitive hierarchy [4], the Colin Brain Atlas [5], the BrainMap functional neuroimaging database [6], the Talairach atlas [7], the Mai atlas [8], Entrez-Gene [9], NINDS [10], NIH Office of Rare Diseases Research (ORDR) [11], BioLexicon [12], UniProt [13], Gene Ontology [15] and the Foundational Model of Anatomy [16]. I also extracted hashtags from Twitter. [17,18]

II. METHODS

A. What is a tag?

I use the notion that a tag can be thought of as an informative keyword in order to distinguish tags from terms. Many of the corpus processing approaches were adapted from Manning et. al's books on statistical natural language processing. [19,20]

B. Tf–idf

I first eliminate the very-low-frequency terms, then calculate how important the term is to PubMed. Specifically we use tf–idf (term frequency–inverse document frequency) [21-24] to select a subset of tags from set of terms. Tf–idf is a numerical statistic that reflects how important a word is to a collection of documents. Tf–idf ranks tags by their low variance within a set and high variance between sets. The following steps summarize how tags were identified from a large text corpus:

1. Use Biological databases, Wikipedia, and Twitter hashtags to find terms.

2. Find those terms whose frequency is above a threshold exist in over 11 million PubMed articles.

3. Calculate tf–idf for each frequent term across 11 million PubMed articles.

4. Select those above a tf–idf threshold.

Once the tag set was identified a number of heuristics were developed to annotate the tags. A primary basis of tag annotation is the source itself. For example, a tag coming from the NIH Office of Rare Diseases would be annotated with the meta-tag “disease” and a tag coming from Foundational Model of Anatomy would be annotated with the meta-tag “anatomy.” Another source annotation of a tag is the structure of its Wikipedia article. For example, if the “info-box” of the article has a longitude and latitude, we can infer that the tag is a “place” in addition to knowing the quantitative values of its longitude and latitude.

C. WordNet

WordNet [25,26] is another source of tag annotation. WordNet is a lexical database of English nuns, verbs, adjectives, and adverbs that are grouped into sets of cognitive synonyms (synsets). The WordNet synsets are further characterized by hyperonymy, hyponymy, or ISA relationships. Tags found in WordNet were annotated with their synonym relationships. Finally, tags with annotation have allowed us to develop heuristics to exploit patterns in the structure of words (prefixes, suffixes, and roots) to annotate additional tags. For example, disease tags often end with the words “syndrome,” “deficiency,” or “disease.” Likewise, tags that represent small molecules often have prefixes like “alkyl-“ or suffixes like “–mide”, “-dehyde” or “–hol”. The statistics and entropy of a given tag were derived by counting its frequencies over various corpora. The following steps summarize how tags were annotated:

1) the source of the tag

2) its Wikipedia "info-box"

3) Wordnet synset annotation

4) word-structure annotation heuristics

5) calculation of statistics and entropy from tag counts

D. Mapping Twitter Hashtags to Words and Phrases

The social tagging of a tweet in is done by placing a hash mark in front of a word or phrase, such as #BCSM, #Lyphoma, #BrainTumorThursday, #BreastCancer, #Infertility, #Diabetes, #lymphoedema, #RareDiseaseDay, #RareDisease, #ADHD, #Anorexia, #MultipleSclerosis, #Depression, #OzDOC, or #MedEd. Finding hastags in tweets is very simple; one just looks for one-grams that begin with a hash mark. However, as tags cannot contain spaces and there is a 140 character limit a number unusual morphological change to words and phrases occur in Twitter. To map the Twitter hashtags to our tag set we created a number of mapping heuristics.

Our first mapping rule we call the *scrunched word hashtag heuristic*. A multiple word phrase like Rare Disease Day is scrunched in to the hashtag #RareDiseaseDay. To map these tags to our synsets in our lexical dictionary we scrunched all of our tags converted all of the tag and tweet text to lower case, and looked for exact matches with Twitter hashtags. For example, the tag Multiple Sclerosis is converted to #multiplesclerosis and matched with the text in a tweet using the same tokenization that we describe in the section "Counting Tags in Text."

Our second mapping rule we call the *capitalized word phrase hashtag heuristic*. There is a strong tendency to capitalize multiple word phrases to make them easier to read. For example, it's more common to see the tag #BreastCancer, or #BrainTumorThursday included in a tweet rather than #breastcancer, or #braintumorthursday. These tags become a source of novel tags for our lexical dictionary. If a tag like #BreastCancer already exists then we skip it. If a tag like #BrainTumorThursday doesn't exist then we add spaces before each capital letter and convert it to lower case; #BrainTumorThursday becomes "brain tumor thursday" and a candidate for a novel tag. Standard collocation tests described in detail in Chapter 5 of Chris Manning and Hinrich Schütze's book, Foundations of Statistical Natural Language Processing, [19] are used to determine if that phrase in used in a corpora other than Twitter (i.e. PubMed, Twitter and Wikipedia). If it passes the collocation test then the tag is added to the lexical dictionary and the semantic, syntactic and statistical annotation described in the section "BioTags: A Semantic Lexical Database for the Social Tagging of BioMedical Text" is performed. If a novel tag is very common in Twitter but cannot be validated in another corpora then it is added as a "stub," and users of BioTags are encouraged to annotate it.

E. Counting Tags in Text

Tags are counted by loading them in to a hash table using a python script. A flag can be set to ignore case in which the text for the tags would be converted to lower case before inserting in to the hash and the text to be counted would be converted to lower case before tag frequencies were determined. The python counting script has two more optional parameters for a frequency threshold and a check rate. For example, if I set my frequency threshold to 5 and my check rate to 1,000,000 then the script would check that a tag occurred at least 5 times in 1,000,000 entities (e.g. tweets, abstracts, webpages, etc.), then at 2,000,000 entities the tag would be checked to see if it occurred at least 10 times and so on. Those tags that didn’t meet a threshold are eliminated from the hash table reducing its size.

Individual entities of text (e.g. tweets, abstracts, webpages, etc.) use a local hash for counting. Some punctuation is converted to white space using three regular expressions. The first regular expression '[\.][ ]+' gets converted to ' ,.' . (single white space, period, single white space). The second regular expression '[\,][ ]+' gets converted to ' , ' . (single white space, comma, single white space). These two regular expressions - implemented as the single regular expression [\.|\,][ ]+' to ' , ' - allows us to remove commas and periods that end words but keep words with commas and periods like 3,7-Dihydro-1,3,7-trimethyl-1H-purine-2,6-dione. The last regular expression r'[_+:;=!@$%^&*\"\?\/]' to ‘ ‘ allows us to remove punctuation that can interfere with tag matching. Once punctuation is converted to white space the text is tokenized by white space and all n-grams of to five grams are generated. A local dictionary is used to count the ngrams that match tags in the lexical dictionary. Those local tag counts are added to the overall tag counts after the text is processed.

F. Creating Tag Co-occurrence Matrices

I used a “frequency signature” approach to convert a bag-of-words output to a format that we can use to calculate tag co-occurrence associations and mutual information. Frequency signatures are described in detail in Stefan Evert’s PhD dissertation “The Statistics of Word Cooccurrences Word Pairs and Collocations.” [28,29]

To calculate tag co-occurrence associations and mutual information for two tags, A and B, we need four items of data. The co-occurrence count of A and B, the count of A but not B, the count of B but not A, and the total number of tags in a corpus. This co-occurrence frequency data for a word pair (A,B) are usually organized in a contingency table. The co-occurrence count of A and B, and the total number of tags in a corpus are efficiently and easily counted the count of A but not B, the count of B but not A are tricky and computationally expensive. The insight and advantage of frequency signatures is that they calculate the count of A but not B, the count of B but not A by just counting A and B and the co-occurrence count of A and B. That is, the count of A but not B is equal to count of A minus the co-occurrence count of A and B. Likewise, the count of B but not A is equal to count of B minus the co-occurrence count of A and B. The frequency signature of a tag pair (A, B) is usually written as (f, f1, f2,N). Where f is the co-occurrence count of A and B, f1 is the count of A but not B, f2 is the count of B but not A, and N is the total counts.

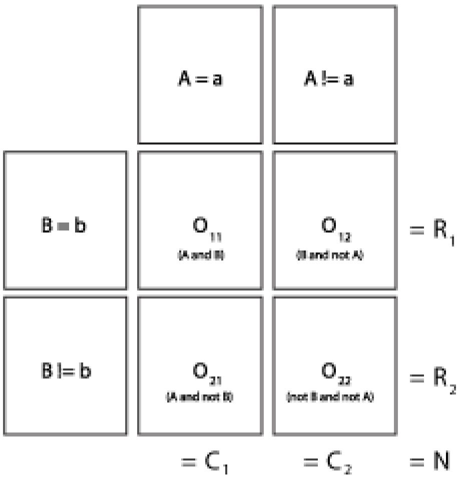
The frequency signature of a tag pair (A, B) is usually written as (f, f1, f2,N). Where f is the co-occurrence count of A and B, f1 is the count of A but not B, f2 is the count of B but not A, and N is the total counts. Notice that the observed frequencies O_{11}, ..., O_{22} can be directly calculated from the frequency signature by the equations below:

5. $O_{11} = f$
6. $O_{12} = f1 - f$
7. $O_{21} = f2 - f$
8. $O_{22} = N - f1 - f2 + f$

Generating all of the data tag co-occurrence association and mutual information calculations using this approach can be generated using a single pass of the data and two associative arrays; one of the tag counts and another for the tag co-occurrence counts.

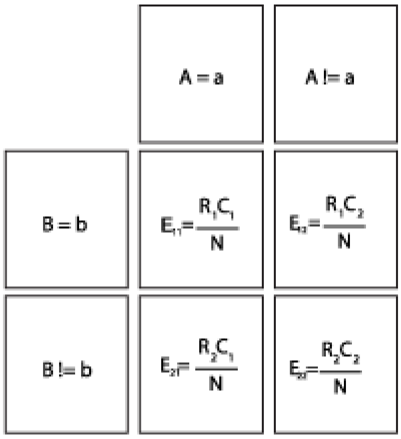
G. Calculating Associations and Mutual Information from Frequency Signatures

Evert shows the many association and mutual information statistics can be calculated from the observed frequencies O_{11}, ..., O_{22} if we can generate the expected frequencies E_{11}, ..., E_{22}. [28,29] The table below (adapted from Evert's dissertation) shows the expected versus observed contingency tables.



	A = a	A != a	
B = b	O_{11} (A and B)	O_{12} (B and not A)	$= R_1$
B != b	O_{21} (A and not B)	O_{22} (not B and not A)	$= R_2$
	$= C_1$	$= C_2$	$= N$

Observed Frequencies



	A = a	A != a
B = b	$E_{11}=\frac{R_1C_1}{N}$	$E_{12}=\frac{R_1C_2}{N}$
B != b	$E_{21}=\frac{R_2C_1}{N}$	$E_{22}=\frac{R_2C_2}{N}$

Expected Frequencies

The sum of all four observed frequencies (called the sample size N) is equal to the total number of pair tokens extracted from the corpus. R1 and R2 are the row totals of the observed contingency table, while C1 and C2 are the corresponding column totals. The expected frequencies can be directly calculated from observed frequencies O_{11}, ..., O_{22} by the equations below:

f. $R1 = O_{11}+O_{12}$

g. $R2 = O_{21}+O_{22}$

h. $C1 = O_{11}+O_{21}$

i. $C2 = O_{12} + O_{22}$

j. $N = O_{11}+O_{12+}O_{12} + O_{22}$

Evert went on to show that several association measures (multinomial-likelihood, binomial-likelihood, Poisson-likelihood, the Poisson-Stirling approximation, and hypergeometric-likelihood,binomial test, Poisson test, Fisher's exact test, z-score, Yates' continuity correction, t-score, Pearson's chi-squared test, Dunning's log-likelihood, mutual information, logarithmic odds-ratio logarithmic relative-risk, Liddell's difference of proportions, Dice coefficient and Jaccard coefficient.) can be easily calculated once one has the expected and observed contingency tables. ." [28,29] For example, the pointwise mutual information (MI) is calculated by (Equation 1) below.

pointwise mutual information

$$MI = \ln(\frac{O_{11}}{E_{11}}) \qquad (1)$$

H. Extracting terms for databases

I extracted terms from one non-biomedical encyclopedia, Wikipedia, and several biomedical databases and ontologies including: the Foundational Model of Anatomy, Entrez-Gene, NINDS, NIH Office of Rare Diseases Research (ORDR), BioLexicon, UniProt, and Gene Ontology. I used encyclopedia and ontologies rather than dictionaries like Wordnet and Wikitionary because we wanted keywords. Words that appear in dictionaries like "apparently", "this" or "false" are too vague to be used as keywords. Terms in ontologies are often written in ways that don't appear in text (e.g. "Typhoid, purified polysaccharide antigen" or "Espin, mouse, homolog of") so we also generated permutations of terms. For example, generating "homolog of mouse Espin" and "mouse homolog of Espin" from"Espin, mouse, homolog of". Case, brackets, parenthesis, periods and commas are often important in biological text (e.g. (6aR,11aR)-9-methoxy-6a,11a-dihydro-6H[1]benzofuro[3,2-c]chromen-3-ol) so we kept case and only removed periods or commas when they are the first or last character in a term. I then matched a list of a couple million possible terms with 11 million PubMed abstracts and kept those terms that matched at least 3 times.

I. Using patterns, capitalization and symbols to find tags in free text

Terms that can be used for tagging often have unusual patterns, capitalization or symbols. For example, the tags: cisplatin-free, Igk-V , inositol 1,4,5-trisphosphate, orfE, (6aR,11aR)-9-methoxy-6a,11a-dihydro-6H[1]benzofuro[3,2-c]chromen-3-ol(R)-2,3-butanediol dehydrogenase all have usual capitalization or symbols. Even though CTGGGA is not a word it refers to an important cis-regulatory element and can be used as a tag. To determine capitalized terms we looked for n-gram in which each word of the n-gram contained at least one capitalized letter other than the first letter. If the frequency of the capitalized n-gram pattern was at least 10 then it was kept. To determine symbol terms we looked for n-gram in which each word of the n-gram contained at least one unusual symbol (i.e. ,-.()[]) other than the first letter or the last letter. If the frequency of the capitalized n-gram pattern was at least 10 then it was kept.

Tags also often follow patterns. For example, EC numbers follow the regular expression ^\d+\.-\.-\.-|\d+\.\d+\.-\.-|\d+\.\d+\.\d+\.-|\d+\.\d+\.\d+\.(n)?\d+$ (e.g. 1.1.1.1) and chromosomal locations follow the pattern [0-9]+[p|q] [0-9]+ (e.g. 3p22 or 5q31-q33) Whether a pattern effectively discriminated a tag we ran the pattern on a sample of a few hundred thousand abstracts and only kept the pattern if its precision was greater than 95%. Patterns also allowed semantic annotation. For example, we can annotate the tag 3p22 with the annotation, "chromosomal location."

J. Post filtering of tags

Regular expressions were used to clean extracted tags. The extracted terms were inspected for mistakes and mistakes that could be written as regular expressions were used to clean the tag lists. For example, the symbols test found terms like 6,7-Dinitroquinoxaline-2,3-dione and 2-Pyridinylmethylsulfinylbenzimidazoles, but it also incorrectly found terms like 5—15 or fertilization—a. The terms removed by a pattern were checked to make sure that it is primarily removing mistakes.

K. Stemming of Terms

A stemmed version of each tag was created by removing all punctuation and spacing, making the tag lower case, replacing British English with American English spelling variants (i.e. our to or, ce to se, re to er, xion to ction, ise to ize, yse to yze, and ogue to og) and Porter stemming [29]. If two tags have the same stem then they were considered spelling variants.

L. Determining polysemy

Most tags found in biomedical literature were jargon and therefore were unambiguous (e.g. anhydrotic ectodermal dysplasia, nasal mucosae, mung beans, Amblyomma cajennense, etc.). I used Wikipedia disambiguation pages to determine the possible senses of common (i.e. those popular enough to have Wikipedia pages) terms like “apple.”

M. Twitter Data

I wrote python scripts to access both the Using the Twitter Search API [16] and the Twitter Streaming APIs. [17] I used these API’s to extract Tweet data, Twitter user profiles and friends & follower relationships between Twitter users. I parsed the tweets for retweets, urls and hastags using regular expressions. I extracted millions of records of tweet and user profile data and stored them in a MySQL database.

N. PubMed Data

I download over 11 million PubMed records for the National Library of Medicine (NLM) as XML over ftp. This represented Medline data through December 2012. I wrote a python script that converted the

XML files to tab delimited text files. I keep the PubMed id, title, abstract, MeSH terms, and author list. All subsequent counting and text processing was done using the tab delimited text files. Tag frequencies for PubMed were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. . The raw tag counts were then used to calculate information theoretic measures.

O. Wikipedia Data

I downloaded the April 4th, 2013 English Wikipedia XML dump file (pages-articles.xml.bz2). [25] The file has the current revisions only, with no talk or user pages. The size of the April 4th, 2013 dump is 9 GB compressed, 42 GB uncompressed . I wrote python scripts to convert the XML in to tab delimited text files and converted the Wiki formatting to a standard formatting. I also wrote python scripts to find those articles with “Info-boxes” and extract that Info-box data in to individual data fields associated with the article. Tag frequencies for Wikipedia were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. . The raw tag counts were then used to calculate information theoretic measures.

III. RESULTS

I created a semantic lexical database for the social tagging of biomedical text. I extended the MeSH keyword set by several hundred thousand keywords. Unlike a controlled vocabulary, such as MeSH, where a user must use one of the controlled terms to annotate text, social tagging applications allow a user to classify with any tag. Even if those tags are misspelt or two different tags refer to the same concept. The primary purpose of BioTags is to provide a lexical resource to correct for tag misspelling and tag heterogeneity. The secondary purpose of BioTags is to provide a lexical resource to help disambiguate tag polysemy (i.e. identical tags that denote different meanings). BioTags provides some semantic enrichment of tags. Our semantic enrichment has focused on annotation relevant to information retrieval in biomedical test. That is annotating terms as diseases, symptoms, genes, anatomy, risk factors, genes, proteins, markers, pathways, chemicals, drugs, ligands, HLA numbers, alleles, antigens, SNPs, loci, enzymes, organisms, species, and phenotypes.

These heuristics extracted 474,165 tag synsets from 11 million PubMed abstracts. This resource is case and punctuation sensitive so “benefit-cost analysis”,”benefit cost analysis”, “Benefit-cost analysis” would ne three separate entities in the same synset tags. A synset has a display term (the most common tag in a synset) and its spelling variants and synonyms. For example, the tag with the display term “Australian frog” belongs to the synset “Litoria aurea”, “Australian frog” or “Green and Golden Bell Frog.”

The potential semantic enrichment is unlimited. For example, that’s say a researcher wants tags associated with cognitive processes (e.g. abstract thought, abductive reasoning, word recognition, spatial working memory, puzzle solving, nonverbal intelligence) and BioTags doesn’t have this semantic class. I put BioTags in a website so a list of cognitive processes tags could be started and the community can enhance it. Once this list gets big enough the technique described in the methods “Supervised Bayesian Approach for the Semantic Annotation of Tags” can be used annotate new tags as “cognitive processes.” The wiki also allows the collective intelligence of the community to correct mistakes, remove tags and add tags.

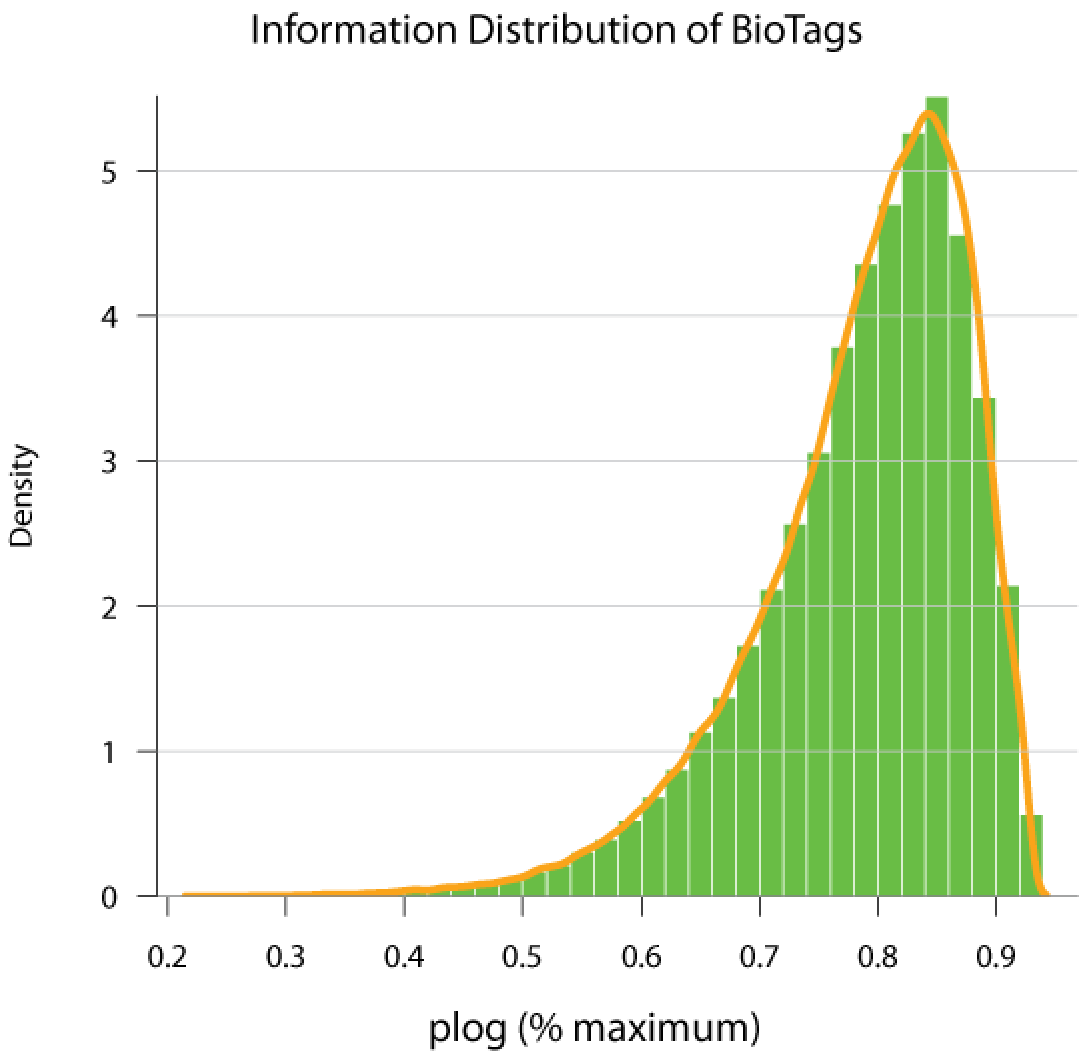
BioTags is currently a useful resource for developing social tagging applications, information retrieval and text mining but certainly incomplete. For example, even though we annotate tags as “drugs,” we don’t have subclasses of drugs such as antidepressants, antipyschotics, antibacterials, pain killers, etc. While “cartilage disease”, and “cartilage diseases” are mapped as spelling variants they aren’t mapped as synonyms of “disease of cartilage” or “disease of the cartilage” by our heuristics. As discussed in the methods we randomly sampled tags and checked them versus existing databases to estimate precision. The precision of the tag extraction was 91.2%.

The precision of the spelling variants was 98.3%. I used list inspection to help guesstimate the recall of our spelling variant and synonym matching and believe the recall to be low. Specifically we seemed to have issues in a number of areas. First, we missed tags that were acronyms are mixed words. For example, we found “omega-3 Fas”,”omega-3 FA”, and “omega-3-FA” were all spelling variants but did not match these with “omega-3 Fatty Acids.”

I missed tags in which the synonyms were reordering of words. For example, “cartilage disease”, and “cartilage diseases” are mapped as spelling variants but they aren’t mapped as synonyms of “disease of cartilage” or “disease of the cartilage.”

The precision of the semantic annotation was 95.7%. Again we feel the semantic enrichment was accurate but very incomplete. Our annotation, while useful, is not very deep. For example, we annotate tags as “drugs,” but we don’t have subclasses of drugs such as antidepressants, antipsychotics, antibacterials, pain killers, etc.

Finally we analyzed the information distribution of the tag set as estimated by its normalized plog density. That is, plog divided by maximum plog. As the histogram below shows, Biotags is composed of primarily medium to high information tags.



Graph 1: plog distribution for 474,165 BioTags (1,211,701,821 total tag counts). The histogram shows that Biotags is composed of primarily medium to high information tags.

IV. DISCUSSION

I created a semantic lexical database for the social tagging of biomedical text. I extended the MeSH keyword set by several hundred thousand keywords. Unlike a controlled vocabulary, such as MeSH, where a user must use one of the controlled terms to annotate text, social tagging applications allow a user to classify with any tag. Even if those tags are misspelt or two different tags refer to the same concept. The primary purpose of BioTags is to provide a lexical resource to correct for tag misspelling and tag heterogeneity. The secondary purpose of BioTags is to provide a lexical resource to help disambiguate tag polysemy (i.e. identical tags that denote different meanings). Finally, BioTags provides some semantic enrichment of tags. Our semantic enrichment has focused on annotation relevant to information retrieval in biomedical test. That is annotating terms as diseases, symptoms, genes, anatomy, risk factors, genes, proteins, markers, pathways, chemicals, drugs, ligands, HLA numbers, alleles, antigens, SNPs, loci, enzymes, organisms, species, and phenotypes.

The potential semantic enrichment is unlimited. For example, that's say a researcher wants tags associated with cognitive processes (e.g. abstract thought, abductive reasoning, word recognition, spatial working memory, puzzle solving, nonverbal intelligence) and BioTags doesn't have this semantic class. I put BioTags in a website so a list of cognitive processes tags could be started and the community can enhance it. Once this list gets big enough the technique described in the methods "Supervised Bayesian Approach for the Semantic Annotation of Tags" can be used annotate new tags as "cognitive processes." The website also allows the collective intelligence of the community to correct mistakes, remove tags and add tags.

BioTags is currently a useful resource for developing social tagging applications, information retrieval and text mining but certainly incomplete. For example, even though we annotate tags as "drugs," we don't have subclasses of drugs such as antidepressants, antipyschotics, antibacterials, pain killers, etc. While "cartilage disease", and "cartilage diseases" are mapped as spelling variants they aren't mapped as synonyms of "disease of cartilage" or "disease of the cartilage" by our heuristics.

The BioTags lexical database, website and all of its component lists are available at http://BioTags.org.

V. REFERENCES

[1] G. Smith, Gene. 2008. Tagging: People-powered Metadata for the Social Web

[2] Boguraev, Branimir., Pustejovsky, James. 1996. Corpus Processing for Lexical Acquisition (Language, Speech, and Communication) The MIT Press

[3] Wikipedia. 2013. Retrieved from http://en.wikipedia.org/

[4] The original solicitation of the DARPA Cognitive hierarchy Retrieved from http://www.darpa.mil/ipto/solicitations/open/05-18_PIP.htm

[5] Collins L, Holmes C, Peters TM, Evans AC 1995. "Automatic 3-D model-based neuroanatomical segmentation." *Human Brain Mapping;* 3 (3): 190-208.

[6] Laird AR, Lancaster JL, Fox PT. 2005. "BrainMap: The social evolution of a functional neuroimaging database." *Neuroinformatics* 3, 65-78.

[7] Talairach J, Tournoux P. 1988. "Co-planar stereotaxic atlas of the human brain." *Thieme,* New York.

[8] Mai JK, Assheuer J, Paxinos G. 1997. "Atlas of the Human Brain." *Academic Press*

[9] Maglott D, Ostell J, Kim D, KD, Tatusova, T. 2005. "Entrez Gene: gene-centered information at NCBI." *Nucleic Acids Res*. 33(Database Issue): D54–D58.

[10] The National Institute of Neurological Disorders and Stroke (NINDS) Disorder Index Retrieved from http://www.ninds.nih.gov/disorders/disorder_index.htm

[11] NIH Office of Rare Diseases Research (ORDR) Retrieved from http://rarediseases.info.nih.gov/

[12] BioLexicon Retrieved from http://www.ebi.ac.uk/Rebholz-srv/BioLexicon/biolexicon.html

[13] UniProt Retrieved from http://www.uniprot.org/downloads

[14] Gene Ontology Retrieved from http://www.geneontology.org/

[15] Rosse C, Mejino JVL. 2003. "A reference ontology for biomedical informatics: the Foundational Model of Anatomy." *J Biomed Inform.* 36:478-500.

[16] Twitter Search API Retrieved from https://dev.twitter.com/docs/api/1/get/search

[17] Twitter Streaming APIs. Retrieved from https://dev.twitter.com/docs/streaming-apis

[18] Jones KS 1972. "A statistical interpretation of term specificity and its application in retrieval". *Journal of Documentation* 28 (1): 11–21. doi:10.1108/eb026526.

[19] Manning, Christopher D. ; Schuetze, Hinrich. 1999. “Foundations of Statistical Natural Language Processing.” *The MIT Press*

[20] Manning, Christopher D. ; Raghavan, Prabhakar; Schuetze, Hinrich. 2008. Introduction to Information Retrieval The MIT Press

[21] Salton G, Buckley C. 1988. "Term-weighting approaches in automatic text retrieval". *Information Processing and Management* 24 (5): 513–523. doi:10.1016/0306-4573(88)90021-0. Also available at CiteSeerX.

[22] Wu HC, Luk RWP, Wong KF, Kwok KL 2008. "Interpreting tf–idf term weights as making relevance decisions". *ACM Transactions on Information Systems* 26 (3): 1–37. doi:10.1145/1361684.1361686.

[23] Salton G; McGill MJ 1986. “Introduction to modern information retrieval.” *McGraw-Hill*. ISBN 0-07-054484-0.

[24] Salton G, Fox EA, Wu H. 1983. "Extended Boolean information retrieval". *Communications of the ACM* 26 (11): 1022–1036. doi:10.1145/182.358466.

[25] Wikipedia. 2013. Data Dump Retrieved from http://en.wikipedia.org/wiki/Wikipedia:Database_download

[26] George A. Miller 1995. “WordNet: A Lexical Database for English.” *Communications of the ACM* Vol. 38, No. 11: 39-41.

[27] Christiane Fellbaum 1998. “WordNet: An Electronic Lexical Database”. *MIT Press*. Cambridge, MA:

[28] Evert, Stefan 2004. “The Statistics of Word Cooccurrences: Word Pairs and Collocations. “ *PhD dissertation, University of Stuttgart.* (www.collocations.de)

[29] Stefan Evert. 2013. “UCS toolkit.” Retrieved from http://www.collocations.de/software.html

[30] Porter MF. 1980. “An algorithm for suffix stripping.” *Program* v14 no. 3, pp 130-137

Unsupervised Classification and Tagging of Free Text

Nik Brown

Computer Science Department, University of California Los Angeles

nik@ucla.edu

ABSTRACT

The classification of free text is a fundamental task for information retrieval. We present an unsupervised algorithm to classify and tag social network text based on information theory. This algorithm uses information theory based heuristics to determine whether a tag found in text should be considered a keyword. We call the algorithm explicit tagging (ET). The goal of ET is to auto-tag text with relevant keywords. ET has several desirable properties for the auto-tagging of social network text. It is: a) unsupervised, b) efficient, c) precise, d) accurate, and e) parallelizable. We validate ET by using it to recover MeSH keywords from over 11 million PubMed abstracts as well as filter Twitter for Biomedical tweets.

Keywords— Topic Model, Entropy, Twitter, Kullback-Leibler divergence, Social Tagging

I. INTRODUCTION

Tagging is a process in which end users use free-form keywords to manually index content in an organic and distributed manner. [1] Social tagging has rapidly become a popular practice in which users add free-form keywords to content in order to organize and categorize it. Automated tagging can complement social tagging in several ways. It is difficult to get enough people to annotate a fraction of the more than 11 million PubMed abstracts by hand. The social tagging of the millions of biological and medical tweets, blogs, and newsfeeds is dependent of the popularity of those resources. Popular social websites that use social tagging such as del.icio.us, Digg, Flickr, facebook, Google+ can have many tags whereas many less popular but important resources may have none. Automated tagging is essential when real-time tagging is needed. Due to the growing complexity of digital social networks and the huge quantity of data they produce daily, it's important that we deal with "big data" efficiently. "Every day, we create 2.5 quintillion bytes of data—so much that 90% of the data in the world today has been created in the last two years alone." [2] While big-data technologies have established the ability to collect and process large amounts of data, most organizations struggle with understanding the data and taking advantage of its value. According to an Economist report: "Extracting value from big data remains elusive for many organizations. For most companies today, data are abundant and readily available, but not well used."[3]

A central issue with "big data" is that not all of the data will be relevant or useful in solving a particular problem and will often add noise instead. However, the purpose of the present research is to better measure the system state of social networks, and this knowledge of state can help with a central issue: which data and algorithms are relevant to a particular problem? Efficiently tagging big data from social networks can help determine which subset of data is most relevant to a particular problem.

ET is very well suited to tagging "big data" from social networks as it is O(n log n) where n is the size of the tag set as opposed to O(n^3) [4] for topic modeling algorithms that use eigen-value decomposition on dense matrices. Specifically, unsupervised tagging is:

a. Unsupervised – No training data is required.

b. Efficient - O(n log n) in time where n is the size of the tag lexica.

c. Precise - The tags are reasonable and informative.

d. Accurate - Is not often missing tags that should be there.

e. Parallelizable -Any number of entropy frequency distributions can be used in a single pass over the text.

f. Has Explicit Tagging (using Rényi entropies)

g. Has Implicit Tagging (using mutual information)

h. Has Word Sense Disambiguation (using mutual information)

i. Generates Entropy Signatures (using Kullback-Leibler divergence)

The recipe for ET is very simple. Given a tag set with associated entropy frequency distributions, the discriminating tags approach is summarized as follows:

1. Explicit Tag Algorithm ("explicit tagging")

   d. Filter the tags set by an information threshold to create an unsupervised tag list using an information theoretic measure such as one of the uses the Rényi entropies (Shannon entropy, collision entropy, min-entropy) or plog

   e. For each article/tweet/web page find tags matching the unsupervised tags and their counts.)

   f. Keep those tags above a count threshold. These called the explicit tags (ET).

The ET algorithm can be calculated in parallel with a single pass through the text. That pass also generates an ergodic signature. Ergodic signatures are a particularly interesting aspect of this work. The use of words in natural language is an ergodic process. Researchers have shown entropy rate constancy in text [5], ergodic signatures in tagging distributions [6], and even a complexity and entropy of literary styles [7]. These ergodic natural signatures provide an additional dimension that help detect language features such as keyword spamming even when the spammers are not using "spam words." Information theoretic approaches have the simplicity of keyword counting; the keyword weight adjustment and can detect not only the words used but the style in which they are used through complexity of natural language.

We validate explicit tagging by comparing its algorithmic tagging to MeSH keywords carefully chosen by their authors in over 11 million PubMed articles. We also use it to filter biomedical tweets from

Twitter to determine whether the approach can be used to categorize the noisy, short text generated by social media.

II. METHODS

A. BioTags

We used “BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text.” [8] for our base tag set.

B. PubMed

We download over 11 million PubMed [9] records for the National Library of Medicine (NLM) as XML over ftp. This represented Medline data through December 2012. We wrote a python script that converted the XML files to tab delimited text files. We keep the PubMed id, title, abstract, MeSH terms, and author list. All subsequent counting and text processing was done using the tab delimited text files. Tag frequencies for PubMed were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. The raw tag counts were then used to calculate the Rényi entropies (plog, Shannon entropy, collision entropy, min-entropy) , min-entropy) and plog scores.

C. Twitter

We wrote python scripts to access both the Using the Twitter Search API [13] and the Twitter Streaming APIs. We used these API’s to extract Tweet data, Twitter user profiles and friends & follower relationships between Twitter users. We parsed the tweets for retweets, urls and hastags using regular expressions. We saved all of the data provided by Twitter for tweets and users and as specified Twitter REST API v1.1. [14] We extracted millions of records of tweet and user profile data and stored them in a MySQL database. Tag frequencies for Twitter were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. The raw tag counts were then used to calculate information theoretic measures.

D. Counting Tags in Text

Tags are counted by loading a lexical dictionary in to a hash table. The lexical dictionary has separate entries for case, spelling variations and mis-spelling of tags. For example, Color, color, coluor and colour would all have separate entries but would belong to the same synset.

Individual entities of text (e.g. tweets, PubMed abstracts, webpages, etc.) use a local hash for counting. Some punctuation is converted to white space using three regular expressions. Once punctuation is converted to white space the text is tokenized by white space and all n-grams of to five grams are generated. A local dictionary is used to count the ngrams that match tags in the lexical dictionary. Those local tag counts are added to the overall tag counts after the text is processed.

E. Entropy

The Rényi entropies [15] (Equation 1) generalize the Shannon entropy, the plog, the min-entropy, and the collision entropy. As such, these entropies as an ensemble are often called the Rényi entropies (or the Rényi entropy, even though this usually refers to a class of entropies). The difference between these entropies is in the respective value for each of an order parameter called alpha: the values of alpha are greater than or equal to zero but cannot equal one. The Renyi entropy ordering is related to the underlying probability distributions and allows more probable events to be weighted more heavily. As alpha approaches zero, the Rényi entropy increasingly weighs all possible events more equally, regardless of their probabilities. A higher alpha (a) weighs more probable events more heavily. The base used to calculate entropies is usually base 2 or Euler's number base e. If the base of the logarithm is 2, then the uncertainty is measured in bits. If it is the natural logarithm, then the unit is nats.

*Rényi entropies**

$$E(Text) = \frac{-1}{(\alpha - 1)} \log \sum_i p_i^{\alpha} \quad (1)$$

*p_i is 1/frequency of tag i

F. Discriminating Tagging (ET)

ET is very simple. Given a tag set with associated entropy frequency distributions, the discriminating tags approach for ET is summarized as follows:

1. Explicit Tag Algorithm ("explicit tagging")

    a. Filter the tags set by an information threshold to create an unsupervised tag list using an information theoretic measure such as one of the uses the Rényi entropies (Shannon entropy, collision entropy, min-entropy) or plog

    b. For each article/tweet/web page find tags matching the unsupervised tags and their counts.)

    c. Keep those tags above a count threshold. These called the explicit tags (ET).

III. RESULTS

We randomly sampled the explicit tagging of 11 million PubMed articles and compared the algorithmic tags with the MeSH tags. We did this at "low," "medium", and "high" entropy. By "low," "medium", and "high" entropy we mean the 25th percentile, 50th percentile and 75th percentile of the maximum entropy. We used this scaling because the range of entropy varies. The minimum possible entropy value is zero corresponding to the case in which one event is certain (Equation 1)

$$E_{\min} = 1 \cdot \ln\left(\frac{1}{1}\right) = 0 \quad (1)$$

When all states are equally probable ($p_i = \frac{1}{n}$), the entropy value is maximum (Equation 2):

$$E_{\max} = \sum_{i=1}^{n} \frac{1}{n} \ln(n) = n\frac{1}{n}\ln(n) = \ln(n) \quad (2)$$

A proof the minimum and maximum values for entropy is given by Theil in Statistical Decomposition Analysis 1972. [13] Our analysis is focused of entropy thresholds generated using the plog and the Shannon entropy (Equations 3,5). [10]

We randomly sampled 119,995 PubMed abstracts and calculated the exact matches with 973,586 MeSH tags. The low entropy explicit tagging generated 4,185,771 tags of which 664, 329 of the 973,586 MeSH tags were recovered for a recall rate of 68%.

A random sample of 1,000 the additional tags were evaluated of which 853 were found to be relevant by hand for a recall rate of 85%. The relevance of the additional tags is subjective and best explained with some concrete examples. The tables below show the PMID (PubMed ID), Title (of the abstract), MeSH Tags, Low (explicit unsupervised tag with entropy of greater than 25% of maximum entropy) and Medium (explicit unsupervised tag with entropy of greater than 50% of maximum entropy)

PMID	13258677
Title	the effect of hysterectomy on ovarian function in the rabbit.
MeSH	Uterus;Ovary
Low	uterus;ovary;hysterectomy;ovarian
Medium y	

PMID	13258702

Title	hyperinsulinism and premenstrual tension: report of a case of hyperplasia of the islets of langerhans.
MeSH	Menstruation;Hyperinsulinism
Low	tension;hyperplasia;menstruation;premenstrual; langerhans;report case;islets;hyperinsulinism
Medium	menstruation;premenstrual;report case;hyperinsulinism

PMID	13258683
Title	elective induction of labor using pituitrin; an evaluation of routine elective induction on a private obstetrical service.
MeSH	Labor, Induced
Low	service;private;labor;elective;routine;induction
Medium	

PMID	13258681
Title	the midforceps operation; preliminary study of 351 cases
MeSH	Delivery, Obstetric
Low	delivery;operation;obstetric;preliminary study;preliminary
Medium	preliminary study

PMID	13258684
Title	intravenous ethyl alcohol analgesia with intravenous pitocin induction of labor.
MeSH	Labor, Induced;Oxytocin;Anesthesia and Analgesia;Ethanol
Low	ethyl alcohol;anesthesia;labor;anesthesia and analgesia;ethanol;analgesia;ethyl;induction;intravenous;alcohol
Medium	ethyl alcohol;anesthesia and analgesia;ethyl

PMID	13258696
Title	plasma cholinesterase activity in normal pregnance and in eclamptogenic toxemias.
MeSH	Pregnancy;Pre-Eclampsia;Blood;Cholinesterases
Low	cholinesterases;normal;eclampsia;plasma cholinesterase;cholinesterase activity;pregnancy;plasma;cholinesterase
Medium	cholinesterases;eclampsia;plasma cholinesterase;cholinesterase activity;cholinesterase

PMID	13258698
Title	cold-knife conization and residual preinvasive carcinoma of the cervix.
MeSH	NO MeSH Tags
Low	cold knife;preinvasive carcinoma;knife;preinvasive;conization
Medium	preinvasive carcinoma

PMID	13258704
Title	lutein cysts in normal twin pregnancy leading to erroneous diagnosis of hydatid mole; a case report.
MeSH	Pregnancy, Multiple;Hydatidiform Mole
Low	cysts;lutein;twin pregnancy;hydatidiform mole; normal;case report; hydatid;twin;erroneous;diagnosis; pregnancy;hydatidiform;mole
Medium	twin pregnancy;hydatidiform mole;hydatid;erroneous;hydatidiform;mole

PMID	13258701
Title	an instrument for self-examination of the cervix.
MeSH	Gynecology
Low	gynecology;instrument;self;cervix
Medium	Gynecology

Nearly all of the errors are generated from errors in the lexical dictionary. That is, tags like “delivery” and “preliminary” are considered non-informative and scored as errors. Whereas the tag “preliminary study” is considered informative and a useful tag to add to the MeSH term “Delivery, Obstetric.” The failure to recover “Delivery, Obstetric” is due to the tag “obstetric delivery” being missing from the BioTags lexica. Our inspection guesstimates that the recall rate of recovering MeSH tags could be improved by a few percent by improving the base lexica.

plog $$E(X) = -\sum_i \ln p_i \qquad (3)$$

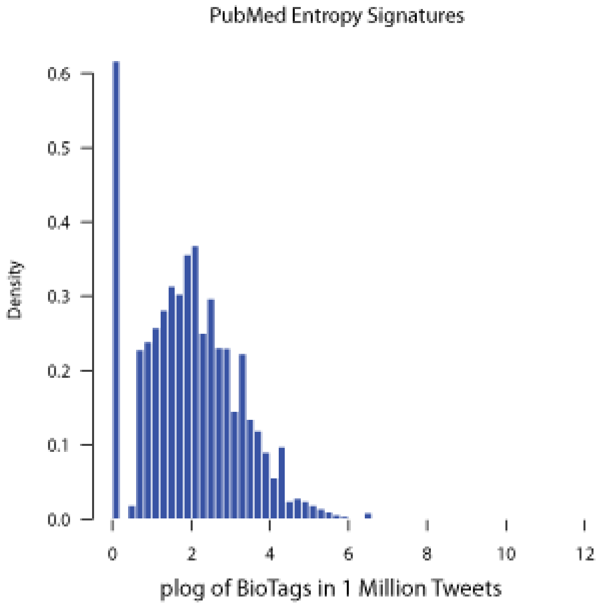
Hartley entropy $$E(X) = -\ln |X| \qquad (4)$$

Shannon entropy $$E(X) = -\sum_i p_i \ln p_i \qquad (5)$$

Compexity analysis

The explicit tag algorithm requires a single pass over n tags. Each step requiring a ln(n) hash look up. The memory required is a single associative array for a tag set of size n.

We ran the classification analysis on one million tweets gathered from the Twitter streaming API. To see if we could tag tweets. As the histogram below shows using the Biotags lexicon filters the million tweets by generating plog of zero for about 60% of one million tweets. We found for the noisy small text of Twitter the ergodic signatures acted like a filter rather than a classifier.



Biotags lexicon generates plog of zero for about 60% of one million tweets.

Inspections of the tweets with entropy zero by hand indicate that zero entropy tweets are very rarely relevant to biology or medicine. The tables below shows the results of using BioTags to classify 1 million tweets. The classifier selects tweets with relevant words but doesn't handle keyword spam well. For example, "sniffing glue sniffing glue sniffing glue," "Wobble base wobble base, " and "Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice Mice" does very well. Further a tweet like "...Itai...itai..." is probably not referring to Itai-itai (or ouch-ouch) bone mineral disease. From our results, we view Twitter classification using entropy as more of a Twitter filter than a classifier.

Tweet	**plog**
Mice Mice Mice Mice Mice Mice Mice Mice Mic Mice Mice Mice Mice Mice Mice Mice Mice Mic Mice Mice Mice Mice Mice Mice Mice Mice Mic Mice	12.255

#quoteyourteacher sniffing glue sniffing glue sniffing glue sniffing glue	12.08
Wobble base wobble base wobble wobble wobble	10.159
Stratum corneum. Stratum lucidum. Stratum granulosum. Stratum spinosum. Stratum germinativum. #5LayersOfTheEpidermis #LEARNING	8.968
Anomic aphasia. Global aphasia. Isolation aphasia Broca's aphasia. Wernicke's aphasia.	8.768
Erythroblastosis fetalis! Hydrops fetalis!	8.749
Causes of Boutonniere Deformity: Causes of Boutonniere Deformity:	8.701
@eminemguevara electroencephalogram, magneti resonance imaging,magnetic resonance angiography...=))	8.673
Atrial septal defects ..	8.66
Christmas Carol for Obsessive Compulsive Disorder ---Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells, Jingle Bells...	8.646
Fine needle aspiration biopsy	8.626
Posterior tubercle of posterior arch of c2	8.592
Bronchiolitis obliterans with organizing pneumon	8.537

(BOOP): Bronchiolitis obliterans with organizing pneumonia (BOOP) affects your lun...	
Hypothalamic-Pituitary-Adrenal axis.... Okayyy.	8.41
Nicotinamide adenine dinucleotide and flavin adenine dinucleotide	8.405

IV. DISCUSSION

In this research we describe explicit unsupervised tagging (ET), an unsupervised efficient, scalable and parallelizable algorithm to classify and tag social network text based on information theory called explicit tagging (ET). This tool provide us with a simple but scalable and parallelizable set of tools to classify big text data produced by the social web. The use of information theoretic measures to classify “big data” is a central focus of this work. As stated before, 90% of the data in the world today has been created in the last two years. A major issue with “big data” is filtering the wheat from the chaff. The Rényi entropies (Equation 1) are very simple to use, and easily parallelizable with most the computational cost up front counting frequency signatures. These properties make them ideally suited for classifying “big data.” Specifically, unsupervised tagging is:

a. Unsupervised – No training data is required.

b. Efficient - O(n log n) in time where n is the size of the tag lexica.

c. Precise - The tags are reasonable and informative.

e. Parallelizable -Any number of entropy frequency distributions can be used in a single pass over the text.

f. Has Explicit Tagging (using Rényi entropies)

g. Has Implicit Tagging (using mutual information)

h. Has Word Sense Disambiguation (using mutual information)

i. Generates Entropy Signatures (using Kullback-Leibler divergence)

We validate explicit tagging by comparing its auto-tagging of PubMed abstracts with the human annotated MeSH terms. We found that it accurately recovered human annotated tags. We also used ET to filter Twitter found biomedical data. We found for the noisy small text of Twitter we relied on the ergodic signatures acting like a filter rather than tagging for classification. Short tweets with one informative word and words that repeat informative words do well with our entropy filter. Twitter has a number of characteristics that might allow us to convert our entropy filter to aid our tagging classifier. Using Kullback-Leibler divergence to test and down weight tweets that differ from the “real” biomedical tweets would prevent keyword spamming. Keeping track of a user’s tweet history and making a rank adjustment based on a history of biomedical tweets might also help.

REFERENCES

[1] G Smith. 2008. Tagging: People-powered Metadata for the Social Web

[2] IBM. 2012. What is big data? Retrieved from http://www-01.ibm.com/software/data/bigdata/

[3] Economist. 2012. Economist Intelligence Unit Retrieved from http://www.eiu.com/

[4] Cai, Deng., He, Xiaofei., and Han, Jiawei. 2008.Training Linear Discriminant Analysis in Linear Time Proc. 2008 Int. Conf. on Data Engineering (ICDE'08)

[5] Genzel, D., Charniak, E., 2002. Entropy Rate Constancy in Text. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02) pages 199-206

[6] Kontoyiannis. 1996. The complexity and entropy of literary styles. NSF Technical Report No. 97, Department of Statistics, Stanford University,

[7] Plotkin, J. B. , and Nowak, M. A.. 2000. Language evolution and information theory. Journal of Theoretical Biology, pages 147-159.

[8] Brown, Nik. 2013. BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text. In submission

[9] PubMed/Medline. 2013. PubMed XML Data Retrieved from http://www.nlm.nih.gov/databases/journal.html

[10] Gray, R. M. (1990). Entropy and Information Theory: Springer-Verlag.

[11] The Apache Software Foundation. 2013. Apache License Version 2.0 Data Retrieved from http://www.apache.org/licenses/LICENSE-2.0.html

[12] Creative Commons. 2013. Creative Commons Attribution license 3.0. Data Retrieved from http://creativecommons.org/licenses/by/3.0/us/

[13] Twitter Search API Retrieved from https://dev.twitter.com/docs/api/1/get/search

[14] Twitter Streaming APIs. Retrieved from https://dev.twitter.com/docs/streaming-apis

[15] Theil. 1972. Statistical Decomposition Analysis. Studies in Mathematical and Managerial Economics, 14.

A Calculus of Word Entropy

Nik Bear Brown

Computer Science Department, University of California Los Angeles

nik@ucla.edu

ABSTRACT

The massive amount of text produced by social networks imposes great challenges to search information quickly and accurately. The diversity of data on the Internet make is increasingly difficult for one search engine to address a wide range of needs. Twitter is particularly interesting in that users often create neologisms in the form of hash-tags that may not match known keywords. We present a dynamic iterative tag-based search which allows users to create a search kernel as a list of tags. This paper reports an information theoretic calculus that allows users to create “wiki-style” search kernels from tag sets. This approach uses the Rényi entropies, plogs, and a simple calculus to convert these tag sets in to search kernels. The property of these kernels that they can always be shown as tag sets which allows users easily create, share and re-mix them. We use real-world examples to illustrate the pros and cons of using re-mixable, tag-based, information theoretic search kernels for an iterative persistent Twitter search/filter.

Keywords - Twitter, Social Search, Social Tagging, Rényi Entropy, Information Theory

INTRODUCTION

Choosing keywords for search can be a difficult task. This is particularly true of Twitter where users often create neologisms in the form of hash-tags that may not match known keywords. The 140 character limit for Twitter also increases the likelihood that relevant tweets won't include exact keyword matches. An alternative to exact keyword matching is to create a list of keywords of interest and calculating a score for how related the tweet is to the list. In this work, I calculate the "quasi-entropy of a tweet" to generate an entropic rank that estimates how related a tweet is a list of tags. The basic algorithm is to find tags in text, look up quasi-entropies, sum and generate an entropic rank and set of tags for the tweet. The calculations are simple and parallelizable; they take the form of – log (p) where p is typically a frequency count. Most of the computation involves pre-computing the counts for the tag set. Once that is done, determining the entropy involves retrieving a pre-computed value for each tag in a tweet and adding them up.

This tag based search kernel search/filtering is well suited to the dynamic nature of text on the social web. The social web has changed the nature of text made available for public consumption. From the time of the Gutenberg printing press until the advent of Web 2.0, nearly all text presented in public was written by professionals. [1,2] Whether was a book, a business or government record, a sermon, a news or opinion article, a scientific paper, or an advertisement, it was written by a professional with the intent to communicate information and/or ideas. Not until the social web and Twitter did musings about what a non-celebrity ate for breakfast or whether someone likes naps was widely available for public consumption. This text is very different from highly edited, often jargon-filled text intended to communicate ideas or information to peers, like technical papers or newspaper articles.

To develop a calculus for ranking tag set matches in text, I focus on two ideas from information theory: ergodicity and entropy. [3] In statistics, ergodicity describes a random process wherein the average time for one sufficiently long realization of events is the same as the ensemble average. That is, the ensemble's statistical properties (such as its mean or entropy) can be deduced from a single, sufficiently long sample of the process. In other words, there are long-term invariant measures that describe the asymptotic properties of the underlying probability distribution, and they can be measured by following any single reprehensive portion if followed long enough. The notion of using entropy as a measure of system state and dynamics comes both from statistical physics and from information theory. In information theory, entropy is also a measure of the uncertainty in a random variable. In this context, however, the term usually refers to the Shannon entropy (Equation 5), which quantifies the expected value of the information contained in a message (or the expected value of the information of the probability distribution). The concept was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication." [4] Shannon entropy establishes the limits to possible data compression and channel capacity. That is, the entropy gives a lower bound for the efficiency of an encoding scheme (in other words, a lower bound on the possible compression of a data stream). Typically this is expressed in the number of ‘bits’ or ‘nats’ that are required to encode a given message.

The ergodic notion that a sufficiently long sample of a single process represents the process as a whole allows us to use invariant measures to estimate not only the amount of information in a given tweet but the kind of information as well For example, let’s say we want to classify a tweet to by language. The entropy of many languages has been determined. English has 1.65 bits per word, French has 3.02 bits per word, German has 1.08 bits per word, and Spanish has 1.97 bits per word. Given the probability density function of word entropies and the average bits per word of a single tweet we could then assign probabilities that it is English, French, German, or Spanish. Additionally, there is a lot of evidence that the use of words in natural language is an ergodic process. Researchers have shown entropy rate constancy in text [5], ergodic signatures in tagging distributions [6], and even a complexity and entropy of literary styles [7]. These ergodic natural signatures provide an additional dimension that help detect language features such as keyword spamming even when the spammers are not using “spam words.” Information theoretic approaches have the simplicity of keyword counting; the keyword weight adjustment and can detect not only the words used but the style in which they are used through complexity and entropy of natural language.

In this work we use the equations for the Rényi entropies and plogs to create what we call a “quasi-entropy” for each tag. The quasi-entropy is an estimate of the “information” conveyed by the tag with respect to a user. For example, the tag 'functional magnetic resonance imaging' coveys more information to a user than the tag ‘with’ and as such should have a higher quasi-entropy. We differentiates a “quasi-entropy” from is that the initial estimate is generating using one of the Rényi entropy equations but that estimate can be adjusted manually should a user not agree with the estimate. We perform no manual adjustment in this work; but will explore extending the calculus in future work. By using quasi-entropy for text classification we would like the ability to combine several tags into a search kernel to estimate a quasi-entropic rank for that tag set. In this approach a search kernel is generated from a list of tags, to "add entropy" we simply combine lists. If one wants to weigh individual tag entropy or a list of tags this scaling is easily done by the properties of logs. The sum the log n times is the same as n times the log, so we can use the properties of logs to scale. This process of generating a list if tags and calculating quasi-entropic rank allows for the iterative development of tag lists (search kernels). The tags with the highest entropic rank that co-occur with the search kernel tags are presented to the users possible additional tags to add to the search. We illustrate the approach by developing several search kernels to search Twitter for bio-medically related tweets.

Text can be classified with n tag-based Entropic Ranks

Figure 1. In a single pass through the data, text can be classified with a set of n quasi-entropic ranks.

METHODS

A. BioTags

We used "BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text." [8] http://biotags.org/ as our base tag set.

B. Entropy

The p_i in all of our entropy equations refers to 1/(frequency of tag i).

Our min entropy is 0 (Equation 1):

$$E_{\min} = 1 \cdot \ln\left(\frac{1}{1}\right) = 0 \quad (1)$$

When all states are equally probable ($p_i = \frac{1}{n}$), the entropy value is maximum (Equation 2):

$$E_{\max} = \sum_{i=1}^{n} \frac{1}{n} \ln(n) = n \frac{1}{n} \ln(n) = \ln(n) \quad (2)$$

A proof the minimum and maximum values for entropy is given by Theil in *Statistical Decomposition Analysis* 1972. [9] Our analysis is focused of entropy thresholds generated using the plog and the Shannon entropy (Equations 3,5). [3]

We calculated a plog using equation 3:

$$E = -\sum_{i} \ln p_i \qquad (3)$$

We calculated the Hartley entropy using equation 4:

$$E = -\ln |X| \qquad (4)$$

We calculated the Shannon entropy using equation 5:

$$E = -\sum_{i} p_i \ln p_i \qquad (5)$$

C. Counting Tags in Text

Tags are counted by loading a lexical dictionary in to a hash table. The lexical dictionary has separate entries for case, spelling variations and mis-spelling of tags. For example, Color, color, coluor and colour would all have separate entries but would belong to the same synset.

Individual entities of text (e.g. tweets, PubMed abstracts, webpages, etc.) use a local hash for counting. Some punctuation is converted to white space using three regular expressions. Once punctuation is converted to white space the text is tokenized by white space and all n-grams of to five grams are generated. A local dictionary is used to count the ngrams that match tags in the lexical dictionary. Those local tag counts are added to the overall tag counts after the text is processed.

D. Twitter

We wrote python scripts to access both the Using the Twitter Search API [10] and the Twitter Streaming APIs. We used these API's to extract Tweet data, Twitter user profiles and friends & follower relationships between Twitter users. We parsed the tweets for retweets, urls and hastags using regular expressions. We saved all of the data provided by Twitter for tweets and users and as specified Twitter REST API v1.1. [11] We extracted millions of records of tweet and user profile data and stored them in a MySQL database. Tag frequencies for Twitter were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. The raw tag counts were then used to calculate information theoretic measures.

E. Entropy

The Rényi entropies [7] (Equation 6) generalize the Shannon entropy, the plog, the min-entropy, and the collision entropy. As such, these entropies as an ensemble are often called the Rényi entropies (or the Rényi entropy, even though this usually refers to a class of entropies). The difference between these entropies is in the respective value for each of an order parameter called alpha: the values of alpha are greater than or equal to zero but cannot equal one. The Renyi entropy ordering is related to the underlying probability distributions and allows more probable events to be weighted more heavily. As alpha approaches zero, the Rényi entropy increasingly weighs all possible events more equally, regardless of their probabilities. A higher alpha weighs more probable events more heavily. The base used to calculate entropies is usually base 2 or Euler's number base e. If the base of the logarithm is 2, then the uncertainty is measured in ‘bits’. If it is the natural logarithm, then the unit is ‘nats’.

$$E(Text) = \frac{-1}{(\alpha\,\text{-}1)} \sum_i \ln p_i^{\alpha}$$

Equation 6. Rényi entropies (p_i refers to 1/(frequency of tag i))

We also used the negative log of the frequency of a term or plog as an estimate of entropy for the purpose of calculating entropic rank. The normalized entropy is determined by dividing an entropy estimate by the logarithm of the number of possible observations.

F. Quasi-Entropic Rank

Quasi-Entropic Rank Algorithm

1. Filter the tags set by an entropy threshold to create an entropic tag list.

2. For each article/tweet/web page find tags matching the entropic tags and their counts.

3. For each tag with at least one count divide the entropic tag count by number of tags in text.

4. Keep those tags above a threshold. These called the explicit tags.

5. Sum the entropy of the explicit tags and divide by the total number of tags found in the text.

6.(Optional) Divide the entropic rank by a theoretical or empirical maximum entropic rank to create a normalized entropic rank in the range 0.0 to 1.0.

G. PubMed Data

We download over 11 million PubMed [13] records for the National Library of Medicine (NLM) as XML over ftp. This represented Medline data through December 2012. We wrote a python script that converted the XML files to tab delimited text files. We keep the PubMed id, title, abstract, MeSH terms, and author list. All subsequent counting and text processing was done using the tab delimited text files. Tag frequencies for PubMed were generated by using BioTags as a lexical dictionary to generate raw tags counts and total tag counts. .The raw tag counts were then used to calculate the Rényi entropies (plog, Shannon entropy, collision entropy, min-entropy) and plog.

RESULTS

We first used the BioTags tag set to filter one million tweets (Figure 2) and found that 60% of the tweets were given an quasi-entropic rank of zero. Looking through the zero rank tweets by hand found only tweets irrelevant to biomedicine. We then searched twitter by hand looking for a set of tweets related to “brain mapping” and found 788 over a course of a few weeks. We found that our entropic rank filter would have kept these tweets for which 99% had a non-normalized entropic rank between 2 and 6 (Figure 3).

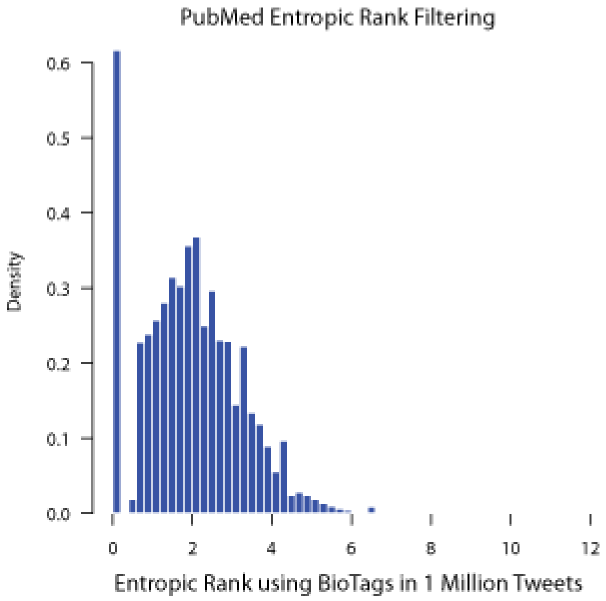


Figure 2. The BioTags “tag-bag” set 60% of one million tweets to an Entropic Rank of zero.

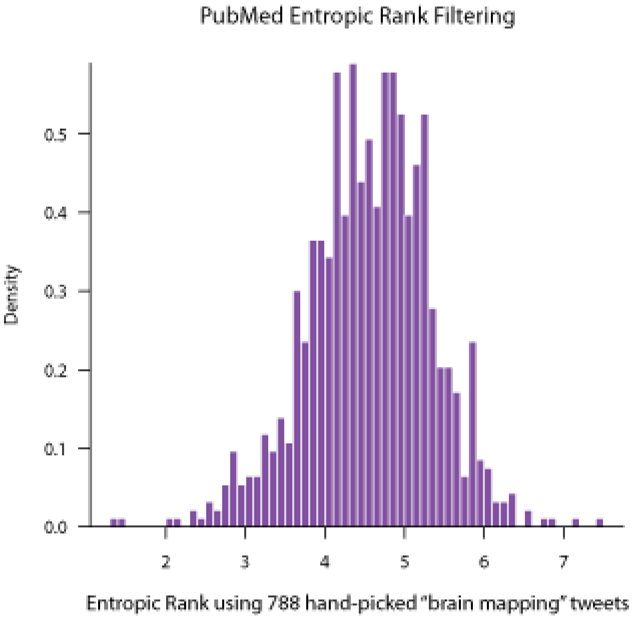


Figure 3. The BioTags "tag-bag" Entropic Rank did not filter a set of hand-picked brain mapping tweets.

Our entropic tagging should not filter out PubMed articles as the represent "real" biomedical text. We ran the BioTags entropic rank filtering on several sets of PubMed XML files. Each PubMed XML file represents about 30,000 abstracts. Every XML entropic rank distribution that we analyzed were very similar to the distributions shown for medline 13n0498 (Figure 4) and medline 13n0499 (Figure 5).

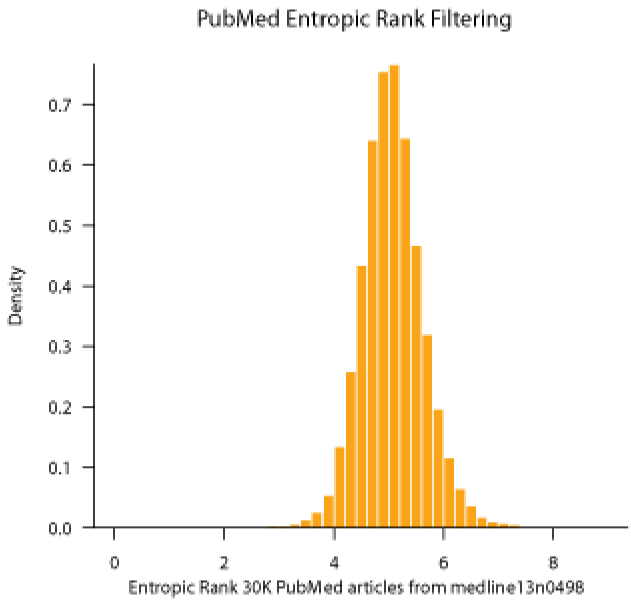


Figure 4. BioTags Entropic Rank Filtering keeps PubMed articles in medline 13n0498.

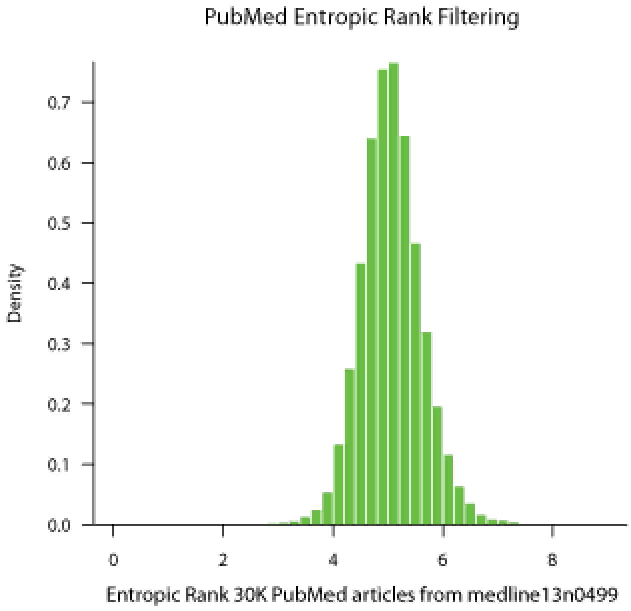


Figure 5. BioTags Entropic Rank Filtering keeps PubMed articles in 13n0499.

Table 1 shows examples of tweets at various quasi-entropic rank. These results indicate that quasi-entropic rank can act as an effective filtering approach.

Tweet	Rank
Carpal Tunnel Syndrome And Tarsal Tunnel Syndrome: Title: Carpal Tunnel Syndrome And Tarsal Tunnel SyndromeCateg... http://t.co/AsQj3fRW	0.79
Nervous Tissue. Muscular Tissue. Connective Tissue. Epithelial Tissue.	0.558
Fine needle aspiration biopsy	0.518
Homo sapiens | Homo habilis | Homo erectus | Homo antecessor | Homo ergaster | Homo neanderthalensis | RT if you see your species!	0.487
Manic Depressive.	0.47
Glandular epithelia	0.468
Sensorineural Hearing Loss is caused by damage or malfunction of the cochlea/the auditory nerve http://t.co/cHIoLdNb	0.395

Featured Article: The Projection of Nerve Roots on the Posterior Aspect of Spine From T11 to L5: A Cadaver and R... http://t.co/1B8Esdy1	0.235
Safely Manage Joint #Inflammation with #Curcumin - http://t.co/etU9UlcN @LifeExtension	0.235
BioMed Central's open access journal Genome Biology : Specific proteins, trigger hygienic behavior of the adults honey bees and promote the...	0.235
Magnesium Stearate: Make Sure It's Not On Your Vitamin Bottle's Ingredient List :: http://t.co/1EpO6ens	0.235
I hate gas stoves with a deep burning passion. No pun intended.	0.235
@hanicattack when I first started this I was getting weak like that. So I went out and bought me some special k protein bars and drink mix.	0.036
wtf my TL wont load	0
Crumpet o'clock	0
@magicman08 Yes but I like the art and character design it has. Gameplay just	0

sucks	
@_ImJustE: Booty Hopskotch"" how does that game go?? Lmao	0
want the killer body? Its all in the plan http://t.co/835Huj3n	0
Evidently, I'm a far ways from being great. Lol	0

Table 1. Examples of Tweets vs Entropic Rank

Table 2 shows examples of tweets at various quasi-entropic rank 0.25. These results indicate that quasi-entropic rank can act as an effective filtering approach.

Related Entropic Tags	**keyword**
cingulate cortex,functional mri,dorsal anterior,aviv,anterior cingulate,cortex,dorsal anterior cingulate,anterior cingulate cortex,cingulate,eeg fmri,tel aviv university,mri,eeg,mind,france,israeli,robot,bir,#sciamblogs,mountain,thursday,twins,kiwi,meiosis,blogging,fmri	fmri
brain imaging,psychoanalysis,blog	brain

	imaging
brain mapping	brain mapping
fmri,blog,congenital heart disease,congenital heart,aviv,read,heart disease,congenital,eeg fmri,tel aviv university,eeg,cancer patients,beneficiaries,volumetric,chemotherapy,liver cancer,liver,mri scan,functional mri,mri	functional mri
nuclear magnetic resonance spectroscopy,magnetic resonance,nuclear magnetic resonance,magnetic resonance spectroscopy,spectroscopy	nuclear magnetic resonance spectroscopy
pet scan	pet scan
clear cell,positron emission,tomography,tomography computed,#gucancer,fluorodeoxyglucose,fluorodeoxyglucose	positron emission

positron,computed tomography,clinical value,positron emission tomography,positron,atherosclerosis,renal,renal cell	tomography
voxel,sculpt,meshes	voxel

Table 2. Tweets Entropic Rank 0.25

DISCUSSION

In this research, we show that information-theoretic measures such as the Rényi entropies (Hartley entropy, Shannon entropy, collision entropy, min-entropy) as a basis for creating a simple calculus to generate search filters for messy, noisy social network text. This approach allows us to create unsupervised efficient, scalable and parallelizable algorithm to classify text in tweets. The use of word tag quasi-entropy allows users to create search kernels that they can share, tweak and re-mix as they can always be shown as tag lists.

These properties of computational efficiency and understandable tag lists make them well suited for filtering social network text. By using word quasi-entropy, like probabilistic approaches, the weight the keywords are adjusted more appropriately than by simply counting tag matches. Unlike probabilistic approaches there is evidence that the use of words in natural language is an ergodic process. Researchers have shown entropy rate constancy in text, ergodic signatures in tagging distributions, and even a complexity and entropy of literary styles. These ergodic natural signatures provide an additional dimension that help detect language features such as keyword spamming even when the spammers are not using “spam words.” Information theoretic approaches have the simplicity of keyword counting; the keyword weight adjustment and can detect not only the words used but the style in which they are used through complexity and entropy of natural language. A main limitation of this approach is that it applies to cases when there is already a base tag set or users can think of a relevant keyword list.

The use of entropic rank filtering can potentially help supplement social network analysis for understanding the dynamics of networks. The number of possible connections in a network grows exponentially with the number of nodes in it; hence, social network analysis is typically limited to small networks. The use the word calculus for pruning graphs will be a central aspect of future work. This should allow computationally expensive approaches to expand the size of the networks they can handle by focusing on the essential structure in a network.

REFERENCES

[1] Shirky, C. 2010. “Cognitive Surplus: Creativity and Generosity in a Connected Age.” *Penguin Press.*

[2] Shirky, C. 2008. “Here comes everybody: the power of organizing without organizations.” *Penguin Press.*

[3] Gray, R. M. 1990. “Entropy and Information Theory.” *Springer-Verlag.*

[4] Shannon, C.E. 1948. “A Mathematical Theory of Communication.” *Bell System Technical Journal,* 27(3), 379–423.

[5] Genzel, D., Charniak, E., 2002. “Entropy Rate Constancy in Text.” *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL-02) pages 199-206

[6] Kontoyiannis. 1996. “The complexity and entropy of literary styles.” *NSF Technical Report No. 97, Department of Statistics, Stanford University*

[7] Plotkin, J. B. , and Nowak, M. A.. 2000. “Language evolution and information theory.” *Journal of Theoretical Biology,* pp 47-159.

[8] Brown, Nik. 2013. “BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text.” In submission

[9] Theil. 1972. “Statistical Decomposition Analysis.” *Studies in Mathematical and Managerial Economics,* 14.

[10] Twitter. 2013. “Twitter Search API.” Retrieved from https://dev.twitter.com/docs/api/1/get/search

[11] Twitter (2013). REST API v1.1 Resources. Retrieved from https://dev.twitter.com/docs/api/1.1

[12] Twitter. 2013. “Twitter Streaming APIs.” Retrieved from https://dev.twitter.com/docs/streaming-apis

[13] PubMed/Medline (2013) PubMed XML Data Retrieved from

http://www.nlm.nih.gov/databases/journal.html

References

AG's news. 2013. “AG's corpus of news articles.” 2013. Retrieved from http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

Albert, R. and Barabási, A. 2002. “Statistical mechanics of complex networks.” *Rev. Mod. Phys*., 74:47–97.

Alchin, M. 2008. “Pro Django” *Apress*.

Alexander, I., and Maiden, N. 2004. “Scenarios, Stories, Use Cases” *Wiley*.

arXiv. 2013. “arXiv Bulk Data Access” Retrieved from http://arxiv.org/help/bulk_data

Autodesk. 2013 “Maya.” Retrieved from http://www.autodesk.com/products/autodesk-maya/overview

Albert-László, B. 2003. “Linked: how everything is connected to everything else and what it means for business, science, and everyday life.” *Plum.* New York, NY:

Barrat, A., Barth´elemy M., Pastor-Satorras, R.,Vespignani. A., 2004, “The architecture of complex weighted networks” *PNAS*11, 3747-3752.

Beck, K; et al. 2001. “Manifesto for Agile Software Development “*Agile Alliance*.

Beck, K; et al. 2001. “Principles behind the Agile Manifesto.” *Agile Alliance*

Behara, E., Krickeberg, K., & Wolfowitz, J. 1973. “Probability and information theory II” *Springer-Verlag*

Bell, G. 2009. “Building Social Web Applications” *O'Reilly Media, Inc*

Bilder RM, Parker DS, Brown N, Kalar D, Sabb F, Glahn D, Bearden C, Poldrack R, Shattuck D, Cannon TD, London E, Freimer N, Toga AW. 2007. “Cognitive Phenomics: Informatics Strategies for Schizophrenia Research.” *Biennial Meeting of the International Congress on Schizophrenia Research*, Colorado Springs CO.

Bilder RM, Parker DS, Brown N, Kalar D, Sabb F, Glahn D, Bearden C, Poldrack R, Shattuck D, Cannon TD, London E, Freimer N, Toga AW. 2007. “Informatics Strategies for Neuropsychiatric Phenomics.” *Annual Meeting of the Society of Biological Psychiatry*, San Diego CA.

Bilder, R., Poldrack, R., Parker, DS., Reise, SP., Jentsch, DJ., Cannon, T., London, E., Sabb, FW., Foland-Ross, L., Rizk-Jackson, A., Kalar, D., Brown, N., Carstensen, A. and Freimer, N. 2009. “The Neuropsychology of Mental Illness (C. 18 Ed.).”

Bilder RM, Parker DS, Poldrack RA, Kalar D, Brown N, Toga AW. 2007. “Mapping Cognition: Development of Cognitive Ontologies for Visualization and Modeling of Brain-Behavior Relationships. “*Annual meeting of the International Neuropsychological Society*, Portland OR.

BioLexicon 2013. Retrieved from http://www.ebi.ac.uk/Rebholz-srv/BioLexicon/biolexicon.html

Bird, S., Klein, E., & Loper, E. 2009. “Natural Language Processing with Python” *O'Reilly Media, Inc.*

Blei, D. M. 2012. “Introduction to Probabilistic Topic Models.” *Comm. ACM*, 55(4), 77–.

Blei, D. M. N., Andrew Y.; Jordan, Michael I; Lafferty, John. 2003. “Latent Dirichlet allocation.” *Journal of Machine Learning Research*, 3, 993–10 22

Boguraev, B., Pustejovsky, J. 1996. “Corpus Processing for Lexical Acquisition “ *The MIT Press*

Borgatti, S., Ajay, B, Daniel J.; Labianca, G. 2009. “Network Analysis in the Social Sciences.” *Science*, 323(5916), 892–.

Brandes, U. 2001. “A Faster Algorithm for Betweenness Centrality.” *Journal of Mathematical Sociology*, 25, 163

Bressert, E. 2012. “SciPy and NumPy” *O'Reilly Media, Inc.*

Brown, N. 2013a. “A Calculus of Word Entropy. “ In submission *ACM Conference on Online Social Networks.* Retrieved from http://nikbearbrown.com/AoP/

Brown, N. 2013b. “BioTags: A Semantic Lexical Database for the Social Tagging of Biomedical Text. “ In submission *BIBM 2013 : IEEE International Conference on Bioinformatics and Biomedicine*. Retrieved from http://nikbearbrown.com/AoP/

Brown, N. 2013c. “The Unsupervised Classification and Tagging of Free Text.” In submission *The 7th ACM International Conference on Web Search and Data Mining Conference (WSDM2014)*. Retrieved from http://nikbearbrown.com/AoP/

Chapelle, O., Joachims, T., Radlinski, F., & Yue, Y. 2012. “Large-scale validation and analysis of interleaved search evaluation.” *ACM Transactions on Information Systems*, 30(1), 1-41.

Collins L, Holmes C, Peters TM, Evans, AC. 1995. “Automatic 3-D model-based neuroanatomical segmentation.” *Human Brain Mapping*; 3 (3): 190-208.

Common Crawl. 2013. “Five Billion Webpage Data Dump” Retrieved from http://commoncrawl.org/

Conway, D., and White, J. M. 2011. “Machine Learning for Email” *O'Reilly Media, Inc.*

Conway, D., and White, J. M. 2012. “Machine Learning for Hackers” *O'Reilly Media, Inc.*

Cover, T. M., and Thomas, J. A. 1991. “Elements of Information Theory” *Wiley.*

Crawley, MJ. 2012. “The R Book” *John Wiley and Sons.*

Creative Commons. 2013. Creative Commons Attribution license 3.0. Retrieved from http://creativecommons.org/licenses/by/3.0/us/

Crucitti, P. V. L., and Porta, S. 2006. “Centrality measures in spatial networks of urban streets.” *Physical Review* E, 73.

D3. 2013. “Javascript Visualization” Retrieved from http://d3js.org/

DARPA. 2013. “DARPA Cognitive hierarchy” Retrieved from http://www.darpa.mil/ipto/solicitations/open/05-18_PIP.htm

Crotty, D. 2008. "Why Web 2.0 is failing in Biology?" http://cshbenchmarks.wordpress.com/2008/02/14/why-web-20-is-failing-in-biology/

Dehmer, M., and Basak, S. C. 2012. "Statistical and Machine Learning Approaches for Network Analysis" *John Wiley and Sons.*

Durlauf, S., Young, P. 2004. "Social Dynamics." The MIT Press. Cambridge, MA

Easley, D., Kleinberg, J. 2010. "Networks, Crowds, and Markets: Reasoning about a Highly Connected World" *Cambridge University Press.*

Economist. 2012. "Economist Intelligence Unit" Retrieved from http://www.eiu.com/

Emmert-Streib, F., and Dehmer, M. 2009. "Information Theory and Statistical Learning" *Springer-Verlag.*

Ernstson, H. 2010. "Reading list: Using social network analysis (SNA) in social-ecological studies." *Resilience Science* Retrieved from http://rs.resalliance.org/2010/11/03/reading-list-using-social-network-analysis-sna-in-social-ecological-studies/

Eve, R. A., Horsfall, S., and Lee, M. E. 1997. "Sociology and Complexity Science" *Sage Publications.*

Evert, S. 2004. "The Statistics of Word Cooccurrences: Word Pairs and Collocations." PhD dissertation, *University of Stuttgart*. (www.collocations.de)

Evert, S. 2013. UCS toolkit. Retrieved from http://www.collocations.de/software.html

Fellbaum, C . 1998. "WordNet: An Electronic Lexical Database." *The MIT Press.* Cambridge,

Fox, J., and Brown, N. 2007. "Automatically Extracting Acronyms from Biomedical Text." *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering*. Boston, Massachusetts, USA; 14–17 October 1245-1248.

Fox, J., and Brown, N. 2007. “Data Dependencies in the Quantitation of Affymetrix Gene Expression Data.” *Proceedings of the WCECS International Conference on Computational Biology.* San Francisco, California, USA; 24-26 October,29-33.

Fox, J., and Brown, N. 2007. “Sensitivity and Consistency of Affymetrix GeneChip Normalization Methods. “ *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering.* Boston, Massachusetts, USA; 14–17 October, 1081-1086.

Freeman, L. 2004. “The Development of Social Network Analysis: A Study in the Sociology of Science” *Empirical Press*.

Ganguly, N., Deutsch, A., and Mukherjee, A. 2009. “Dynamics On and Of Complex Networks” *Applications to Biology, Computer Science, and the Social Sciences* Birkhäuser Boston.

Garnaat, M. 2011. “Python and AWS Cookbook” *O'Reilly Media, Inc.*

Gene Ontology 2013. “GO Database.” Retrieved from http://www.geneontology.org/

Genzel, D., Charniak, E., 2002. “Entropy Rate Constancy in Text.” *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)* pages 199-206

Gould, R.V. 1993. “Collective action and network structure. “ *American Sociological Review* 58 (2), 182–1

Granovetter, M. 1973. “The strength of weak ties.” *American Journal of Sociology* 78 (6). pp. 1360–1380.

Granovetter, M. 1976. “Network sampling: Some first steps.” American Journal of Sociology 81 (6). pp. 1287–1303.

Gray, J., Chambers, L., and Bounegru, L. 2012. “The Data Journalism Handbook” *O'Reilly Media, Inc*

Gray, R. M. 1990. “Entropy and Information Theory” *Springer-Verlag.*

Greene, D. and Cunningham, P. 2006. “Practical solutions to the problem of diagonal dominance in kernel document clustering.” *Proc. 23rd International Conference on Machine learning (ICML 2006).* Dataset: BBC

Han, J., Kamber, M., and Pei, J. 2011. “Data Mining: Concepts and Techniques” *Morgan Kaufmann.*

Hastie, T., Tibshirani, R., and Friedman, J. 2001. “The Elements of Statistical Learning” *Springer-Verlag*.

Hofmann, T. 1999. “Probabilistic Latent Semantic Indexing.” *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval.*

Hristea, F. T. 2012. “The Naïve Bayes Model for Unsupervised Word Sense Disambiguation.” *Springer-Verlag.*

IBM. 2008. “Dow Jones and the Healthcare business of Thomson Reuters plan to collaborate with IBM.” Retrieved from http://www-03.ibm.com/press/us/en/pressrelease/25130.wss

IBM. 2012. “What is big data?” Retrieved from http://www-01.ibm.com/software/data/bigdata/

IBM. 2013. “IBM’s Center for Social Software.” Retrieved from http://www.research.ibm.com/social/

Idris, I. 2012. “NumPy Cookbook” *Packt Publishing.*

Irizarry K, K. V., Li C, Brown N, Nelson S, Wong W, Lee CJ. 2000. “Genome-wide analysis of single-nucleotide polymorphisms in human expressed sequences.” *Nat Genet*, 26(2), 233-236.

Pitt-Francis, J. and Whiteley, J. 2012. “Guide to Scientific Computing in C++.” *Springer-Verlag*.

Jones. K.S. 1972. “A statistical interpretation of term specificity and its application in retrieval.” *Journal of Documentation* 28 (1): 11–21. doi:10.1108/eb026526.

Jost, L. 2006. “Entropy and diversity.” *Oikos*, 113, 363–.

Kadushin, C. 2012.”Understanding social networks: Theories, concepts, and findings.” *Oxford University Press.*

Kilduff, M., Tsai, W. 2003. “Social networks and organisations.” *Sage Publications.*

Kirk, A. 2012. “Data Visualization: a successful design process.” *Packt Publishing.*

Kong, J. S., Sarshar, N., and Roychowdhury, V. P. 2008. “Experience versus talent shapes the structure of the Web.” *PNAS*, 105(3 7), 13724-13729.

Kontoyiannis. 1996. “The complexity and entropy of literary styles. NSF Technical Report No. 97”, *Department of Statistics, Stanford University.*

Kruschke, J. 2010. “Doing Bayesian Data Analysis” *Academic Press.*

Ladha, K. 1992. “The Condorcet Jury Theorem, Free Speech, and Correlated Votes.” *American Journal of Political Science,* 36(3), 617-634.

Laird AR, Lancaster JL, Fox PT. 2005. “BrainMap: The social evolution of a functional neuroimaging database.” *Neuroinformatics* 3, 65-78.

Last.fm. 2013. “Last.fm music tags.” Retrieved from http://www.last.fm/charts/toptags

Leipzig, J., and Li, X.-Y. 2011. “Data Mashups in R” *O'Reilly Media, Inc.*

Lin,J., Dyer,C., Hirst,G. 2010. “Data-Intensive Text Processing with MapReduce (Synthesis Lectures on Human Language Technologies)” *Morgan and Claypool Publishers*

Waldrop, MM. 2008. “Science 2.0: Great New Tool, or Great Risk? Wikis, blogs and other collaborative web technologies could usher in a new era of science. Or not.” *Scientific American* Retrieved from http://www.scientificamerican.com/article.cfm?id=science-2-point-0-great-new-tool-or-great-risk

Mai JK, Assheuer J, and Paxinos G. 1997. “Atlas of the Human Brain.” *Academic Press*

Denker, M., Grillenberger, C. and Sigmund, K. 1976. "Ergodic Theory on Compact Spaces (Lecture Notes in Mathematics)" *Springer*

Maglott, D., Ostell,L., Pruitt, K. and Tatusova, T. 2005. "Entrez Gene: gene-centered information at NCBI" *Nucleic Acids Res.* 33(Database Issue): D54–D58.

Manning, C. , Raghavan, P., and Schuetze, H. 2008. "Introduction to Information Retrieval." *The MIT Press*

Manning, C. and Schuetze, H. 1999. "Foundations of Statistical Natural Language Processing." *The MIT Press*.

matplotlib 2013. Retrieved from http://matplotlib.org/

McCallum, Q. E. 2012. "Bad Data Handbook." *O'Reilly Media, Inc.*

McKinney, W. 2012. "Python for Data Analysis." *O'Reilly Media, Inc.*

McKusick VA. 2007. "Mendelian Inheritance in Man and Its Online Version, OMIM." *Am J Hum Genet.* 80(4):588-604.

Miller, G. 1995. "WordNet: A Lexical Database for English." *Communications of the ACM* Vol. 38, No. 11: 39-41.

Minqing H. and Bing L. 2004. "Mining and summarizing customer reviews.*" Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Seattle, Washington, USA

Mosteller, F., and Wallace, D. L. 1984. "Applied Bayesian and Classical Inference." *Springer-Verlag.*

Murphy, K. P. 2012. "Machine Learning: A Probabilistic Perspective." *The MIT Press.*

Murray, S. 2013. Interactive Data Visualization for the Web." *O'Reilly Media, Inc.*

NetworkX. 2013. "NetworkX software." Retrieved from http://networkx.github.io/

Newman, M. E. J., 2001. “Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.” *Physical Review* E 64, 016132.

NIH Office of Rare Diseases Research (ORDR). 2013. Retrieved from http://rarediseases.info.nih.gov/

NINDS. 2013. “The National Institute of Neurological Disorders and Stroke (NINDS) Disorder Index.” Retrieved from http://www.ninds.nih.gov/disorders/disorder_index.htm

Opsahl, T., Agneessens, F., Skvoretz, J. 2010. “Node centrality in weighted networks: Generalizing degree and shortest paths.” *Social Networks*, 3 2(3), 245-.

O'Reilly, T. 2005. “What Is Web 2.0?” Retrieved from http://oreilly.com/pub/a/web2/archive/what-is-web-20.html

Pace, L. 2012. “Beginning R: An Introduction to Statistical Programming.” *Apress.*

Papadimitriou, C., Raghavan, P., Tamaki, H., Vempala, S. 1998. Latent Semantic Indexing: A probabilistic analysis. *Proceedings of ACM PODS*. http://dx.doi.org/10.1006/jcss.2000.1711

Percival, H. 2013. “Test-Driven Web Development with Python.” O’Reilly Media, Inc.

Perera, S., and Gunarathne, T. 2013. “Hadoop MapReduce Cookbook.” *Packt Publishing.*

Phillips, L. 2012. “gnuplot Cookbook.” *Packt Publishing.*

Plotkin, J. B. , and Nowak, M. A.. 2000. “Language evolution and information theory.” *Journal of Theoretical Biology*, pages 147-159.

Bramer, M. 2013. “Principles of Data Mining (Undergraduate Topics in Computer Science)”

PubMed/Medline. 2013. “PubMed XML Data.” Retrieved from http://www.nlm.nih.gov/databases/journal.html

Quiñonero-Candela, J. 2008. “Data Set Shift.” *The MIT Press*

R graphics 2013. Retrieved from http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html

R Package. 2013. ‘tnet’ Retrieved from http://cran.r-project.org/web/packages/tnet/

R Package. 2013. ‘lda’. Retrieved from http://cran.r-project.org/web/packages/lda/lda.pdf

R Package. 2013. ‘topicmodel’. Retrieved from http://cran.r-project.org/web/packages/topicmodels/vignettes/topicmodels.pdf

R Package. 2013. ’igraph’. 2013. Retrieved from http://igraph.sourceforge.net/

R Package. 2013. ‘pROC’. Retrieved from http://web.expasy.org/pROC/

R Package. 2013. ‘tm’. Retrieved from http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

Rosse C, Mejino JVL. 2003. “A reference ontology for biomedical informatics: the Foundational Model of Anatomy.” *J Biomed Inform.* 36:478-500.

Russell, MA. 2011. “Mining the Social Web.” *O’Reilly Media, Inc.*

Salton G., Buckley C. 1988. “Term-weighting approaches in automatic text retrieval. “ *Information Processing and Management* 24 (5): 513–523. doi:100.1016/0306-4573(88)90021-0.

Salton G, Fox EA, Wu H. 1983. “Extended Boolean information retrieval.” *Communications of the ACM* 26 (11): 1022–1036. doi:10.1145/182.

Salton G., McGill MJ. 1986. “Introduction to modern information retrieval.” *McGraw-Hill.*

Samara, T. 2010. The Designer’s Graphic Stew: Visual Ingredients, Techniques, and Layout Recipes for Graphic Designers: Rockport Publishers.

Sanjeev A, Ge, R., and Moitra, A. 2012. “Learning Topic Models—Going beyond SVD.” Retrieved from arXiv:1204.1956v2

Scott, JP. 2000. “Social Network Analysis: A Handbook.” *Sage Publications.*

Segaran, T., and Hammerbacher, J. 2009. “Beautiful Data.” *O’Reilly Media, Inc.*

Shannon, C.E. 1948. “A Mathematical Theory of Communication.” *Bell System Technical Journal*, 27(3), 379–423.

Shirky, C. 2008. “Here comes everybody: the power of organizing without organizations.” *Penguin Press*

Shirky, C. 2010. “Cognitive Surplus: Creativity and Generosity in a Connected Age.” *Penguin Press*

Shneiderman B. 2008 “Computer science. Science 2.0.” *Science.* Mar 7;319(5868):1349-50.

SideFX. 2013. “SideFX Houdini.” Retrieved from http://www.sidefx.com/

Smith, G. 2008. “Tagging: People-powered Metadata for the Social Web.”

statnet: Software tools for the Statistical Modeling of Network Data. 2013. Retrieved from http://statnetproject.or

Steele, J., and Iliinsky, N. 2010. “Beautiful Visualization.” *O’Reilly Media, Inc.*

StopBadware. 2013. “StopBadware not for profit organization.” Retrieved from https://www.stopbadware.org

Strogatz, S. 2001. “Exploring complex networks.” *Nature*, 410: 268-276.

Sunstein, C. 2009. “Infotopia: How Many Minds Produce Knowledge.” *Oxford University Press.*

Sunstein, C. 2006. “Deliberating Groups versus Prediction Markets (or Hayek's Challenge to Habermas).” *Episteme: A Journal of Social Epistemology* pp 192-213.

Surowiecki, J.2005. “The Wisdom of Crowds.” *Anchor Press.*

Talairach J, Tournoux P. 1988. “Co-planar stereotaxic atlas of the human brain.” *Thieme*, New York.

Teetor, P. 2011. “25 Recipes for Getting Started with R.” *O’Reilly Media, Inc.*

The Apache Software Foundation. 2013. “Apache License Version 2.0.” Retrieved from http://www.apache.org/licenses/LICENSE-2.0.html

The Register. 2007. “Scientists shun Web 2.0.” Retrieved from http://www.theregister.co.uk/2007/03/11/sxsw_science_web_2/

Theil. 1972. “Statistical Decomposition Analysis.” *Studies in Mathematical and Managerial Economics*, 14.

Turkington, G. 2013. “Hadoop Beginner's Guide.” *Packt Publishing.*

Twitter. 2013. “REST API v1.1 Resources.” Retrieved from https://dev.twitter.com/docs/api/1.1

Twitter. 2013. “Twitter Search API.” Retrieved from https://dev.twitter.com/docs/api/1/get/search

Twitter. 2013. “Twitter Streaming APIs.” Retrieved from https://dev.twitter.com/docs/streaming-apis

UC Irvine Machine Learning Repository. 2013. “Spambase.” Retrieved from http://archive.ics.uci.edu/ml/datasets/Spambase

UniProt. 2013. Retrieved from http://www.uniprot.org/downloads

Vega-Redondo, F. 2007. “Complex Social Networks.” *Cambridge University Press*, New York

Venner, J. 2009. “Pro Hadoop.” *Apress.*

Verzani, J. 2011. “Getting Started with RStudio.” *O’Reilly Media, Inc.*

Waldrop, M. 2008. “Big data: Wikiomics.” *Nature* 455:22-25

Walters, P. 1975. “Ergodic Theory.” *Springer-Verlag.*

Warden, P. 2011. “Big Data Glossary.” *O’Reilly Media, Inc.*

Ware, C. 2008. “Visual Thinking.” *Morgan Kaufmann.*

Wasserman, SF., and Faust, K. 1994. “Social Network Analysis in the Social and Behavioral Sciences.” *Cambridge University Press.*

Watts, D. 2003. “Small Worlds: The Dynamics of Networks between Order and Randomness: Princeton Studies in Complexity.”

Watts, D. 2004. “Six Degrees: The Science of a Connected Age.“ *W. W. Norton and Company Inc*

Wellman, B. 2008. “Review: The development of social network analysis: A study in the sociology of science.” *Contemporary Sociology*, 37, 221-222.

White, T. 2012. “Hadoop: The Definitive Guide.” *O’Reilly Media, Inc*.

Wikipedia. 2013. 2013. Retrieved from http://en.wikipedia.org/

Wikipedia. 2013. “Data Dump.” Retrieved from http://en.wikipedia.org/wiki/Wikipedia:Database_download

Williams, G. 2011. “Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery (Use R!)”

Witten, I., Frank, E., and Holmes, G. 2011. “Data Mining: Practical Machine Learning Tools and Techniques.” *Morgan Kaufmann.*

Wu HC, Luk RWP, Wong KF, Kwok KL 2008. “Interpreting tf–idf term weights as making relevance decisions. “ *ACM Transactions on Information Systems* 26(3): 1–37. doi:10.1145/1361684.1361686.

Xu, G., and Li, L. 2013. “Social Media Mining and Social Network Analysis.” *IGI Global.*

Yeung, R. W. 2002. “A first course in information theory.” *Kluwer Academic/Plenum Publishers.*

Youngjoong K. 2012. “A study of term weighting schemes using class information for text classification. “ *SIGIR'12. ACM.*