Text Analysis via Binomial Tails

Omid Madani, omadani@cisco.com

ABSTRACT

We show that several tasks in text processing, in particular co-occurrence analysis, term weighting in documents, and document similarity, can be modeled by the binomial tail. The tail yields easy to interpret significance scores, and can make finding a good cut-off threshold simpler, or improve ranking tasks and similarity spaces. Because the tail can be efficiently approximated, it is a basic tool that should find applications in text analysis.

KEYWORDS

Co-Occurrence Analysis, Term Weighting, Document Representation, Document Similarity, Binomial Tails, Statistical Significance

1 INTRODUCTION

The binomial is a simple model that arises in the statistical analysis of basic phenomena in numerous domains. In particular, the tail of the binomial is a natural fit to bounding the probability of an extreme event in many settings ("extreme" relative to the binomial modeling). However, the tail involves a sum, of a count to a maximum possible, which makes it inefficient to use. Fortunately, there is an efficient analytic approximation [2, 3], both an upper and lower bound, that is particularly suited for the range of tiny probabilities that is often the case in various problems. This fact may have been unknown in the datamining and information retrieval and natural language communities, and we seek to highlight the potential applications here.

We explore the use of the binomial tail to derive statistical confidences for several text analysis tasks. As will be seen, the tail can be applied to a number of problems, such as term association via co-occurrence, weighting terms in documents, and term or document similarity, and because it can be efficiently computed (approximated), and offers well understood statistical confidence semantics, this points to its potential for simplifying thresholding and improving ranking, vector representations, and similarity spaces for various downstream tasks such as clustering and classification. Our prior work explored the tail for community analysis in social networks [11]. As we will see, there are a number of ways to model a task via the tail, and what we present is a brief exploration of the possibilities. We view the tail as a basic principled tool that offers another lever and dimension of control for text processing.

The next section defines the binomial tail and presents its approximation, and describes a few properties of the tail and the approximation. The following sections develop a few applications and report on preliminary experiments: Section 3 explores the co-occurrence application, and Section 4 and the appendix explore document representation (term weighting) and similarity.

2 THE BINOMIAL TAIL

The binomial tail, $\texttt{Tail}(p, n, k)$, captures the probability that, in tossing a weighted two-sided coin (heads or tails), with probability p of heads, out of n trials or tosses (independent, identical), k or more heads is observed. It is given by the following sum:

$$\texttt{Tail}(p, n, k) = \sum_{k \leq i \leq n} \binom{n}{i} p^i (1-p)^{n-i} \tag{1}$$

In the applications of the tail, we are interested in how low the tail is, thus how far in a probabilistic sense the observed event E of interest ($E = (p, n, k)$) is from a simple random model. In particular, the observed k as a proportion of n, or $q = \frac{k}{n}$, can be substantially higher than the probability p would imply (p is the 'expected' proportion according to the model). The higher $q = \frac{k}{n}$, the lower the tail. The count-based event of interest can be the number of times a term occurs in a document or the number of times two terms co-occur nearby, and so on. Candidate events whose corresponding tail probabilities are not sufficiently low can be ignored or dropped, and the remaining could be ranked by (statistical) *confidence*,[1] defined as $1 - \texttt{Tail}(p, n, k)$.

While Equation 1 is a simple computation, it is inefficient to compute for tens of thousands of events and beyond (term co-occurrences, etc.) specially when[2] $n \gg k \geq 1$. The following upper and lower bounds on the tail work well, and furthermore, shed light on the properties of the tail:

$$\frac{1}{\sqrt{2n}} U \leq \texttt{Tail}(p, n, k) \leq U, \tag{2}$$

where U (the upper bound) is:

$$U = \exp(-n\mathrm{KL}(q||p)), \text{where } q = \frac{k}{n}, \tag{3}$$

and KL(), is the (asymmetric) relative entropy function (or Kullback Leibler divergence) [16]: $\mathrm{KL}(q||p) = q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$, where ln() denotes the natural logarithm function (note: $\frac{U}{\sqrt{2n}}$ in Eq. 2 is the lower bound). The (log) ratio of the observed to expected proportion, the *intensity* of the event, $\log \frac{q}{p}$, is the same as pointwise mutual information (pmi) in the co-occurrence analysis of the next section. We often use the negative log base 10 of the above upper and lower tail probabilities,[3] and take the average of the lower and upper bounds for scoring:

$$\begin{aligned} l(E) &= d\mathrm{KL}(\frac{k}{n}||p)/\ln(10) \qquad \text{(lower score (of event } E\text{))} \\ u(E) &= l(E) + 0.5 \log_{10}(2n) \qquad \text{(upper score)} \end{aligned} \tag{4}$$

Since we use log base 10, the scores 1, 2, 3, $\cdots$ correspond respectively to chance (tail) probabilities of 0.1, 0.01, 0.001, $\cdots$ thus a score of 1 corresponds to a fairly weak confidence (0.1 p-value, confidence 90%), and each successive increment implies 10 times more significance, and scores around 2 (99% confidence) and higher indicate increasingly good confidence.

[1] Confidence that the event is not governed by the random model (chance).
[2] In several special cases, *e.g.* when n is small, exact or nearly exact tail can be computed. De Moivre's (Stirling's) approximation can be used for the factorial terms in $\binom{n}{i}$ [5, 13].
[3] Akin to Richter scale for measuring the power of earthquakes

We make use of two properties from the recent work applying the tail to derive significance of communities in social graphs [11]:

(1) The score is increasing in n (evidence/support) and in $\log \frac{q}{p}$.
(2) The approximation quality, such as relative error, improves with increasing score.

Point (1) above, *i.e.* the dependence on both the number of trials (support) and the intensity (logarithmic) can be deduced from the upper and lower bound formulae [11]. It can be verified that the relative error of approximation is less than 10% when $n \geq 50$ and $KL() \geq 0.5$ (and rapidly shrinks as n increases). For small values of n, one could compute the score exactly. All our co-occurrence experiments of next section (*e.g.* on over 100k abstracts), written in Python, finished around or under a minute.

3 CO-OCCURRENCE ANALYSIS

Term co-occurrence analysis finds a number of applications including phrase discovery (meaningful bigrams), term expansion during searching, sparse sense representations, and topic discovery [4, 6, 12, 14]. Here, we investigate processing of immediate co-locations which has applications to meaningful phrase discovery and query expansion. We also introduce notation next (with examples) as we explain this application of the binomial tail.

Notation and an example. Let the word w_1 occur $\mathrm{F}(w_1)$ times (frequency of w_1) in the corpus, and w_2 to immediately follow w_1 $\mathrm{F}(w_2, w_1)$ times. Let the prior of w_2 be $\mathrm{P}(w_2)$, where the (empirical) prior of a word w is defined as $\mathrm{P}(w) = \frac{\mathrm{F}(w)}{N}$, where $N = \sum_{w_i} \mathrm{F}(w_i)$.[4] As a simple example, if the corpus has two documents, with document d_1, being the sequence $d_1 = (w_1, w_3, w_1)$ (thus $|d_1| = 3$ and note $N = \sum_{w_i} \mathrm{F}(w_i) = \sum_{d_i} |d_i|$), and document d_2 being $d_2 = (w_1, w_2, w_1, w_2)$, then $\mathrm{F}(w_1) = 4$, $\mathrm{P}(w_1) = \frac{4}{7}$, and $\mathrm{F}(w_2, w_1) = 2$ (w_2 follows w_1 twice, in document d_2).

Our binomial tail formulation is simply $\mathtt{Tail}(\mathrm{P}(w_2), \mathrm{F}(w_1), \mathrm{F}(w_2, w_1))$ (or $n = \mathrm{F}(w_1)$, and $k = \mathrm{F}(w_2, w_1)$). The lower it is, we have more confidence that w_2 follows w_1 with probability *exceeding* the background prior $\mathrm{P}(w_2)$.

A popular statistical approach for term co-occurrence is the pointwise mutual information technique (pmi) [6], defined as the log of the ratio $\frac{P(w_2|w_1)}{P(w_2)}$, *i.e.* probability that the word w_2 follows w_1 (the conditional probability) over the unconditional probability or prior of w_2.

There is some similarity between the two scores as the binomial incorporates the intensity (here the pmi), as a component. We report the agreement of the top 5 ranked co-occurring terms for each term w_1, distinguished as a function of a few ranges for frequency of w_1. The experiments are on the NSF abstracts dataset (120k documents) [8] and [10] (test partition, 7.5k documents) (similar patterns were observed on a few other datasets). Table 1 provides statistics on the agreement rates between pmi and binomial, and Fig. 1 shows a few scores and co-occur counts of top ranked co-occurring terms given a few initial example terms. We observe that specially for higher frequency terms w_1 the top term picked also have higher co-occurrence counts via the binomial tail.

[4] P(w) is the (empirical) probability that word w is observed when a random position in the corpus (viewed as concatenation of documents) is examined.

dataset	Newsgroups	NSF Abstracts
Jaccard on top 5 term-sets returned by each method		
$1000 < \mathrm{F}(w_1)$	0.04	0.05
$500 < \mathrm{F}(w_1) \leq 1000$	0.14	0.13
$100 < \mathrm{F}(w_1) \leq 500$	0.46	0.34
$50 < \mathrm{F}(w_1) \leq 100$	0.80	0.65
co-occurrence count of top term picked (on average)		
binomial, $1000 < \mathrm{F}(w_1)$	433	794
pmi, $1000 < \mathrm{F}(w_1)$	5	15
binomial, $500 < \mathrm{F}(w_1) \leq 1000$	94	127
pmi, $500 < \mathrm{F}(w_1) \leq 1000$	8	9
binomial, $100 < \mathrm{F}(w_1) \leq 500$	30	40
pmi, $100 < \mathrm{F}(w_1) \leq 500$	7	7
binomial, $50 < \mathrm{F}(w_1) \leq 100$	13	13
pmi, $50 < \mathrm{F}(w_1) \leq 100$	7	5

Table 1: Runs on the Newsgroups (on test partition, 7.5k docs, N =1.6mil term occurrences) and NSF abstracts datasets (120k docs, 24mil occurrences). The first few rows display the average agreement (Jaccard: $\frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$) among the top 5 ranked by each method, and the next few report on co-occurrence count of the top word returned by each method (broken by frequency range). PMI picks low frequency terms (yielding low co-occurrence counts) and the difference in top 5 grows with frequency of w_1.

It is well known that pmi is sensitive to low counts (of w_2) as it does not include evidence or support, and a number of variants have been proposed, such as raising the numerator of pmi, when written in alternative form $\frac{\mathrm{P}(w_1, w_2)^j}{\mathrm{P}(w_1)\mathrm{P}(w_2)}$, to higher powers $j = 2$ (pmi2) or $j = 3$ (pmi3)[7]. This makes it harder to interpret and can yield negative scores but does favor more common co-occurrences [14]. The binomial approach naturally incorporates the support, and as we observe here, tends to pick phrases with significantly higher co-occurrence counts. With $j = 2$ or 3, the top-ranked get closer to binomial, but still significantly different: average Jaccard similarities on most frequent (> 1000) to less frequent terms (100 to 500) range go from 0.3, to 0.5 between binomial and pmi2, and from 0.7 to 0.6 between binomial and pmi3, and the pmi variants continue to rank somewhat smaller co-occurrences on top.

Discussion and Some Extensions.. As seen in the examples, the tail can pick very frequent w_2 with high priors, since they provide the most evidence (see below for alternatives). Although the ratio $P(w_2|w_1)/P(w_2)$ is already a component of the binomial tail, only obtaining the confidence that the condition probability is higher than the (tiny) prior may not be sufficiently informative, especially once score exceeds certain high values (*e.g.* say 3 corresponding to 99.9%). Therefore, we could require a further constraint on the prior or on the pmi ratio (intensity).

A direct way to constrain the pmi using the tail is as follows: in the invocation for the tail, we can query with a p higher than the prior. For instance, we can ask for the confidence that the conditional probability is at least twice the prior, $2\mathrm{P}(w_2)$, or 10x or 100x the prior, and/or we can constrain the conditional with absolute bounds as well, such as $p = \max(0.01, 10\mathrm{P}(w_2))$ (*i.e.* , the

```
Ranked by tail significance, for "role"
1. 5303.6 1.0 "role of" co=7455 c1=15655 c2=1155547
2. 3518.5 1.1 "role in" co=4518 c1=15655 c2=527997
3. 397.4 2.5 "role played" co=195 c1=15655 c2=1035
4. 211.1 1.2 "role models" co=273 c1=15655 c2=28343
5. 135.6 0.6 "role that" co=454 c1=15655 c2=166174

Ranked by pmi, for "role", pmi (pow=1.0)
1. 6.6 3.2 "role misocyclone" co=2 c1=15655 c2=2
2. 6.6 3.2 "role Dike" co=2 c1=15655 c2=2
3. 6.3 3.0 "role Tough" co=2 c1=15655 c2=3
4. 6.0 2.9 "role MDN" co=2 c1=15655 c2=4
5. 9.7 2.6 "role inthe" co=4 c1=15655 c2=16

Ranked by pmi2, for "role", pmi2 (pow=2.0)
1. 0.0 -2.5 "role of" co=7455 c1=15655 c2=1155547
2. 90.4 -2.6 "role in" co=4518 c1=15655 c2=527997
3. 205.0 -2.6 "role played" co=195 c1=15655 c2=1035
4. 9.3 -3.8 "role models" co=273 c1=15655 c2=28343
5. 4.6 -3.9 "role misocyclone" co=2 c1=15655 c2=2

pmi3 on "role": "of"(-6),"in"(-6.3),"played","models","that"

"sperm" (F=702) (co-occurrence counts shown)
binomial: "egg"(co=41),"competition"(22),"whale"(7),
          "release"(11),"axoneme"(6)
pmi: "Chemoattractant"(2),"Bos"(2),"fertilizes"(4),
     "offensive"(2),"chemoattractant"(2)
pmi2: "fertilizes"(4),"egg"(co=41),"Chemoattractant"(2),
      "Bos"(2),"offensive"(2)
pmi3: "egg"(co=41),"fertilizes"(4),"competition"(22),
      "axoneme"(6),"whale"(7)

"develop" (F=27093) (co-occurrence counts shown)
binomial: "a"(co=7621),"new"(1381),"an"(1580),
          "methods"(348),"efficient"(160)
pmi:      "tetraamidomacrocyclic"(2),"spaceport"(2),
          "superconvergent"(2),"Cyborgs"(2),"hi2h"(2)
pmi2:     "a"(7621),"new"(1381),"an"(1580),
          "methods"(348),"and"(2020)
pmi3:     "a"(7621),"new"(1381),"an"(1580),
          "and"(2020),"the"(1366)
```

Figure 1: Top 5 bigrams for $w_1 \in$ {"role", "sperm", and "develop"}, using binomial and pmi variants (NSF abstracts data): in each line for "role", the tail score and pmi (both $\log_{10}$) as the 1st two columns, followed by the co-occurrence count and term counts ($\mathrm{F}(w)$). Plain pmi is attracted to very low frequencies. Binomial scores have an interpretation (as confidence, or a distance from random chance).

conditional probability should be 10x the prior or 0.01 whichever is higher). The prior multiplier can be a function of how high the prior is (*e.g.* , 2x for most frequent terms, 100x for least). Finally, we could compute the highest pmi ratio at which the confidence remains at fairly high threshold, such as 99.9% (see next section). Furthermore, we can do sensitivity analysis or render the confidence more conservative, for example by subtracting say a 1 from the observed co-count $k = \mathrm{F}(w_2, w_1)$.

4 DOCUMENT REPRESENTATION

Document representation is a fundamental problem that affects the quality and efficiency of down-stream tasks, including search and retrieval, clustering, and classification. We briefly explore binomial significance for term weighting, beginning with two (symmetric) formulations of the tail score, plus use of intensity:

- ***Document centric tails,* or weights as importance of terms to a document:** For a document with $|d|$ term occurrences, we imagine sampling $n = |d|$ times (trials), where term w is picked with probability $p = \mathrm{F}(w)/N$, and we observe $k = \mathrm{F}(w, d)$ occurrences of w in the document.[5] The weight of a term w in d is determined by $\texttt{Tail}(\frac{\mathrm{F}(w)}{N}, |d|, \mathrm{F}(w, d))$.
- ***Term centric tails,* or weights as importance of documents to a term:** Imagine a word w in every trial of $n = \mathrm{F}(w)$ many, picks a document to occur in, where it picks document d with probability $p = |d|/N$ in each trial, and we observe $\mathrm{F}(w, d)$, occurrences of w, in document d. The tail is then: $\texttt{Tail}(\frac{|d|}{N}, \mathrm{F}(w), \mathrm{F}(w, d))$.
- **Intensity.** Use intensity after possibly thresholding by a confidence derived from above (*e.g.* require 99% confidence), as intensity alone does not account for low evidence (see Table 3, and previous section). Thus, the weight of a (remaining) term is $\log_2 \frac{q}{p}$, where $q = k/n$.

Note that in the above tails, akin to tfidf [9, 15], both the corpus frequency of terms (the more frequent terms are often less significant) and the number of term occurrences in the document are taken into account, as well as the entire document length (akin to length normalization). The tfidf measure is somewhat heuristic and attempts have been made to connect it to information theoretic principles [1], and weighting (in-part) by the tail provides a more principled alternative.

Figure 2 shows a portion of an example abstract and a few top words ranked based on term-centric, tfidf weights, and intensity weightings, to get a sense of the range of weights and the top rankings. For tfidf weighting, we used tf=$\log_2(\mathrm{F}(w, d) + 1)$ multiplied by idf, idf=$1 + \log(\frac{|D|}{df(w)})$, where $|D|$ is the number of documents in the corpus), and $df(w)$ is the number of documents that w appears in. When we rank terms of a given document by various weighting schemes, we find high correlation between the different weighting schemes, *e.g.* , on NSF abstracts, we get an average Spearman rank correlation of 0.99 between the two tail variants above, 0.97 between the document-view weighting and tfidf, and 0.94 document-view weighting and intensity.

A natural question is the fraction of terms that reach a significance level. In the NSF abstracts data, about half of the on average 70 unique terms in an abstract are deemed insignificant at 99%, and about 2/3 do not reach the 99.9% threshold. Thus the reduction can be substantial. We observe similar numbers on the newsgroups dataset, shown in Table 2.

We assess term weights derived as a function of the binomial scores for document similarity. The newsgroups dataset has 20 class labels (20 newsgroups, roughly equal size partitioning), and we label a document pair positive iff both pair pairs are from the same class.

[5] We are using same notation of Sec. 3. $|d|$ is the number of non-unique term occurrences, and $\mathrm{F}(w, d) \geq 0$ is number of occurrences of w in document d.

```
The Reunion hot spot has been active over at least the past
70 M.y. and has left its mark on the Indian and African
tectonic plates of the Indian Ocean. On the African plate, the
Mascarene ridge is coincident with a significant positive geoid
anomaly, similar to the Hawaiian swell, indicating that the
apparent compensation of the topographic load ...

By tfidf: (score=9.0,F(w,d)=2,F(w)=13,Hawaiian),(8.7,2,15,spot),
(8.5,2,22,anomaly),(8.4,2,28,compensation),(8.3,3,164,Indian),
(7.5,2,49,hot),(7.4,2,71,African),(6.9,1,1,geoid),
(6.6,1,2,Mascarene),(6.4,1,3,Reunion),(6.3,1,5,asthenosphere),
(6.2,2,186,Ocean),(6.1,1,5,swell),(6.0,2,247,deep),
(5.8,1,9,coincident),(5.7,2,386,heat)...

By binomial:(score=6.3,F(w,d)=3,Indian),(6.2,2,Hawaiian),
(6.1,2,spot),(5.8,2,anomaly),(5.6,2,compensation),
(5.1,2,hot),(4.7,2,African),(4.1,1,geoid),(4.0,2,Ocean),
... (1.4,1,determine),(1.4,1,based),(1.4,2,as),(1.4,3,is),
(1.4,1,its),(1.4,1,methods),(1.3,2,with),(1.3,1,support),
(1.3,1,it),(1.3,1,other),...,(1.1,1,that),(0.0,3,to),
(0.0,3,of),(0.0,2,will),(0.0,3,and),(0.0,2,in)

By intensity:[(intensity=9.4,F(w,d)=1,geoid),(8.7,1,Mascarene),
(8.3,1,Reunion),(7.8,1,asthenosphere),(7.8,1,swell),
(7.5,2,Hawaiian),(7.3,2,spot),(7.2,1,coincident),
(7.0,2,anomaly),(6.8,1,Utilizing),(6.7,2,compensation),...
```

Figure 2: An example NSF abstract and the top terms as ranked by tfidf, binomial and intensity. A few bottom terms shown for binomial as well. Binomial and tfidf rankings tend to highly correlate at top as both are sensitive to terms with 2 or higher frequency in the doc (both in-document count F(w,d) and corpus frequency F(w) of each term is shown in the tfidf list). Observe that terms such as 'determine', and 'support' get relatively low confidence (binomial score) in this corpus.

weight (score) threshold→	0	1	2	3
NSF Abstracts	77	73	43	16
Newsgroups	144	128	54	22.5

Table 2: Average number of unique terms left per document after thresholding the binomial significance scores (0=no threshold, 1 = 90% confidence, 2 = 99%, 3 = 99.9%).

We conducted experiments on roughly 100k randomly picked[6] document pairs, ranking the pairs by a few similarity methods, shown in Table 3. A similarity score that scores all the positive pairs higher than the rest yields an ideal ranking. We report maxf1 [17].[7] Note that evaluating the ranking this way is an (imperfect) measure of how good a term representation and a corresponding similarity function is, as there can be both false positives and false negative pairs (*e.g.* two documents inside a group may have no relevance or semantic similarity to one another, and vice versa).

[6]We repeated this experiment more than 5 times (different 100k subsets) and the deviation in maxf1 and average precision was small (less than 0.01) and the relative performances of the methods do not change.

[7]Maxf1 is the score on the precision-recall curve where harmonic mean of precision and recall is maximized.

The similarity score based on tfidf is the dot product of l2-normed tfidf vectors (cosine), where tf=$\log_2(\mathrm{F}(w,d)+1)$ multiplied by idf, idf=$1+\log(\frac{|D|}{df(w)})$, where $|D|$ is the number of documents in the corpus, and $df(w)$ is the number of documents that w appears in. All other techniques first threshold a document (a vector) based on the above term-view binomial score (threshold of 1, 2, or 3). The 'bool' (boolean) similarity technique simply counts the terms in common in the document pair (after thresholding) (ignores weights or scores), while the 'score' technique sums the minimum of the pair of binomial scores for each term in common. The intensity (or pmi) technique sums the minimum of the pair of intensity ratios for each term in common.

We observe that the techniques improve over tfidf. The improved performance of the bool technique along with others, with increasing threshold, implies that a threshold of 1 (90% confidence) is too low, as many noisy terms remain, but around 2 to 3 works relatively better. The superior performance of intensity (after thresholding by significance) is not without plausible explanation: intensity is a measure of how much a term is special to a document and the more such 'special' terms two document have in common, the higher their similarity should be. However, the binomial significance score alone does not reflect this intensity directly.

It is interesting that with almost half the terms removed (per document), the rankings appear to perform better than tfidf-cosine, with significance-based techniques. We also note that we did not attempt to normalize (adjust) for document size in computing similarity, as the binomial modeling already takes that into account. We have also observed that performing a product of the intensities (rather than taking the minimum) performs better according to above maxf1 criterion. More systematic experiments and evaluations on additional datasets are needed to assess, for example, whether similar confidence thresholds work well on different datasets. It is also important to explore the representation for other tasks, such as dimensionality reduction, clustering and supervised learning. Finally, document similarity can be more directly modeled by the binomial tail (Appendix A). We leave further exploration to future work.

	tfidf	bool	score	intensity
maxf1, sig. threshold=1	0.184	0.126	0.151	0.185
maxf1, sig. threshold=2	0.184	0.187	0.194	**0.205**
maxf1, sig. threshold=3	0.184	0.180	0.180	0.186

Table 3: Ranking performance on a random sample of 100k pairs of newsgroups documents, where a pair is positive iff both documents are from the same newsgroup. We use several document representations and similarity functions. All the techniques except tfidf first drop terms below the binomial significance threshold of 1, 2, or 3.

5 CONCLUSIONS

The binomial tail is a versatile tool for deriving significance efficiently, and we explored a few applications. We hope that future work furthers the applications and extends our understanding of the tail's properties and its relation to other techniques.

REFERENCES

[1] A. Aizawa. 2003. An information-theoretic perspective of tfidf measures. *Information Processing and Management* (2003).
[2] R. Arratia and L. Gordon. 1989. Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology* 51 (1989), 125–131.
[3] R. B. Ash. 1990. *Information Theory.* Dover Publications. 115 pages.
[4] S. Mittal A.t A. Mahabal, D. Roth. 2018. Robust Handling of Polysemy via Sparse Representations. In *Seventh Joint Conference on Lexical and Computational Semantics.*
[5] L. Le Cam. 1986. The central limit theorem around 1935. *Statist. Sci.* (1986), 81.
[6] K. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.* (1990).
[7] B. Daille. 1994. *Approche mixte pour extraction automatique de terminologie : statistiques lexicales et filtres linguistiques.* Ph.D. Dissertation. Universit Paris.
[8] D. Dua and C. Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[9] K. S. Jones. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* (1972).
[10] K. Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning.*
[11] O. Madani, T. Ngo, W. Zeng, S. Averin, S. Evuru, V. Malhotra, S. Gandham, and N. Yadav. 2020. Binomial Tails for Community Analysis.
[12] C. D. Manning and H. Schutze. 1999. *Foundations of statistical natural language processing.* Cambridge University Press.
[13] R. Motwani and P. Raghavan. 1995. *Randomized Algorithms.* Cambridge University Press.
[14] F. Role and M. Nadif. 2011. Handling the Impact of Low Frequency Events on Co-occurrence based Measures of Word Similarity - A Case Study of Pointwise Mutual Information. In *International Conference on Knowledge Discovery and Information Retrieval (KDIR).*
[15] G. Salton and M. J. McGill. 1986. *Introduction to modern information retrieval.* McGraw-Hill.
[16] J. A. Thomas T. M. Cover. 1991. *Elements of Information Theory.* John Wiley & Sons.
[17] C. J. van Rijsbergen. 1979. *Information Retrieval.* Butterworth-Heinemann.

A SIMILARITY VIA THE BINOMIAL TAIL

Here we describe a way of modeling the similarity between a pair of documents as a binomial tail. Just as in Sec. 4, a symmetric approach can be formulated for term similarity. We will next describe the approach and our simplifying assumptions to fit the binomial to the task. On the ranking task on newsgroups of Sec. 4, this similarity scores an improved maxf1 of above 0.22 (see Table 3), and preliminary experiments on clustering (via community discovery algorithms) provide some evidence that the similarity space (the graph) induced can be better than what is achieved via cosine similarity.

Description. One document, say d_1 is given (fixed, or conditioned on) and we ask if the other document is sampled at random (*i.e.* , for each of its $|d_2|$ positions, a term is drawn with probability equal to its prior), what is the probability that we observe such pattern of matches, ie the magnitude of the priors of matching terms (how low they are) and the number of such matches? The challenge is reducing the pattern of different matching priors to a one dimensional similarity score (to a one-dimensional binomial tail). We make a few simplifying assumptions: the algorithm sets up a few (matching) events and takes the lowest probability (highest scoring) one.

As shown in Algorithm 1, we use a range of probabilities around a matching term's prior $\mathrm{P}(w)$, and within this range, we do not distinguish whether the same term is matched multiple times or different terms match. The range is determined by a ceiling probability, controlled by a parameter, the *granularity multiple*, $\alpha \geq 1$, where we have found $\alpha = 2$ to work well, the ceiling being $\alpha \mathrm{P}(w)$ (see discussion below).

As an example, say d_1 has 6 terms, $w_1, w_2, \cdots$, with priors 0.03, 0.06, 0.1, 0.11, 0.13, 0.15 respectively, and for simplicity assume all occur once in d_1. Assume $d_1 \cap d_2 = \{w_2, w_4, w_6\}$ (3 terms in common), and $|d_2| = 8$. Then the main loop of Algorithm 1 first finds word w_2 in common, determines the upper ceiling to be $\alpha \times 0.06 = 0.12$, with default $\alpha = 2$, and there are two terms matches in this range (given probability ceiling is set to 0.12), with success probability set at $p = 0.03+0.06+0.1+0.11 = 0.30$, and it is a success if d_2 picks any of these terms. There are two term matches (assuming each matching term occurs once in d_2, therefore $k = 2$. From the first time through the loop, we get binomial_score($p = 0.3, n = 8, k = 2$). The 2nd time through loop also adds w_6 to the matches, k goes to 3, and success becomes more likely with $p = 0.58$. nstays at 8 throughout. We take whichever binomial score is higher (the match is more surprising).

Discussion. There are a number of design decisions to explore, and we briefly discuss and motivate a few. As we decrease the multiple α (minimum of 1.0), we increase *granularity*, but we may get only one matching event, and may ignore words with near priors (that may or may not match). Use of $\alpha > 1$ has a similar conservative nature to using the tail (a tail uses a one-sided interval *vs.* using a point or a smaller interval). On the other hand, if α is made too large (coarse), the score loses its power, as words with a wide range of probability are bundled together, and success probability can reach or exceed 1.0. For simplicity of presentation, we did not set a floor on the priors (in defining the set C), although that could be done, and instead of taking the maximum, it may be sound or beneficial, to add up the scores. Term correlations are not taken into account (by binomial definition), however we have found that removing terms that are (near) duplicates (which can be achieved efficiently) can improve the similarity/ranking results. In addition to exploring the effect of α and algorithmic variations, it would be good to explore the extent to which the scores (the confidences) are calibrated (*i.e.* whether two items, scoring over say 95%, are deemed similar, at roughly 95%, by human raters?). Finally, there may exist more sophisticated statistical models better suited to modeling document similarity.

We note that Algorithm 1 can be implemented efficiently (similar run-time cost to cosine), via the use of hash sets and maps, and involves sorting (terms in a document), and linear scans of both documents.

Algorithm 1: Similarity score, sim(), of two documents, d_1 and d_2, via a reduction to the binomial tail. The required parameter $\alpha \geq 1$, is the granularity (or coarseness) multiple. Assume wlog $|d_1| \leq |d_2|$ or take min(sim(d_1,d_2), sim(d_2,d_1)).

$O \leftarrow d_1 \cap d_2$, mark all 'unexamined', sort ascending by prior.
score $\leftarrow 0$ # The similarity score (binomial tail).
Take next unexamined word w in sorted O:
  $C \leftarrow \{t \in d_1 | \mathrm{P}(t) \leq \alpha * \mathrm{P}(w)\}$ # Terms (in d_1), close prior.
  $p \leftarrow \sum_{t \in C} \mathrm{P}(t)$ # prob. of 'hitting' (any of) these terms.
  $k \leftarrow \sum_{t \in C} \min(\mathrm{F}(t, d_1), \mathrm{F}(t, d_2))$ # Count of matches of C.
  score $\leftarrow \max(score, binomial\_score(p, |d_2|, k))$
  Mark terms in C as examined.
return score