Mining Significant Associations in Large Scale Text Corpora

Prabhakar Raghavan
Verity Inc.
pragh@verity.com

Panayiotis Tsaparas*
Department of Computer Science
University of Toronto
tsap@cs.toronto.edu

Abstract

Mining large-scale text corpora is an essential step in extracting the key themes in a corpus. We motivate a quantitative measure for significant associations through the distributions of pairs and triplets of co-occurring words. We consider the algorithmic problem of efficiently enumerating such significant associations and present pruning algorithms for these problems, with theoretical as well as empirical analyses. Our algorithms make use of two novel mining methods: (1) matrix mining, and (2) shortened documents. We present evidence from a diverse set of documents that our measure does in fact elicit interesting co-occurrences.

1 Overview

In this paper we (1) motivate and formulate a fundamental problem in text mining; (2) use empirical results on the statistical distributions of term associations to derive concrete measures of "interesting associations"; (3) develop fast algorithms for mining such text associations using new pruning methods; (4) analyze these algorithms, invoking the distributions we observe empirically; and (5) study the performance of these algorithms experimentally.

Motivation: A major goal of text analysis is to extract, group, and organize the concepts that recur in the corpus. Mining significant associations from the corpus is a key step in this process. In the automatic classification of text documents each document is a vector in a high-dimensional "feature space", with each axis (feature) representing a term in the lexicon. Which terms from the lexicon should be used as features in such classifiers? This "feature selection" problem is the focus of substantial research. The use of significant associations as features can improve the quality of automatic text classification [18]. Clustering significant terms and associations (as opposed to *all* terms) is shown [8, 14] to yield clusters that are purer in the concepts they yield.

Text as a domain: Large-scale text corpora are intrinsically different from structured databases. First, it is known [15, 22] that terms in text have skewed distributions. How can we exploit these distributional phenomena? Second, as shown by our experiments, *co-occurrences* of terms themselves have interesting distributions; how can one exploit these to mine the associations quickly? Third, many statistically significant text associations are intrinsically uninteresting, because they mirror well-known syntactic rules (e.g., the frequent co-occurrence of the words "of" and "the"); one of our contributions is to distill relatively significant associations.

2 Background and contributions

2.1 Related previous work

Database mining: Mining association rules in databases was studied by Agrawal et al. [1, 2]. These papers introduced the support/confidence framework as well as the *a priori* pruning paradigm that is the basis of many subsequent mining algorithms. Since then it has been applied to a number of different settings, such as mining of sequential patterns and events. Brin, Motwani and Silverstein [6] generalize the a priori framework by establishing and exploiting closure properties for the χ^2 statistic. We show in Section 3.2 that the χ^2 test does not work well for our domain. Brin et al. [5] extend the basic association paradigm in two ways: they provide performance improvements based on a new method of enumerating large itemsets and additionally propose the notion of *implication rules* as an alternative to association rules, introducing the notion of *conviction*. Bayardo et al. [4] and Webb [20] propose branch and bound algorithms for searching the space of possible associations. Their algorithms apply pruning rules that do not rely solely on support (as in the case of a priori algorithms). Cohen et al. [7] propose an algorithm for fast mining of associations with high confidence without support pruning. In the case of text data, their algorithm favors pairs of low support. Furthermore, it is not clear how to extend it to associations of more than two terms.

[*] This work was conducted while the author was visiting Verity Inc.

Extending database mining: Ahonen et al. [3] build on the paradigm of *episode mining* (see [16] and references therein) to define a text sequence mining problem. Where we develop a new measure that directly mines semantically useful associations, their approach is to first use a "generic" episode mining algorithm (from [16]) then post-filter to eliminate uninteresting associations. They do not report any performance/scaling figures (their reported experiments are on 14 documents), which is an area we emphasize. Their work is inspired by the similar work of Lent et al. [13]. Feldman *et al.* describe the KDT system [10, 12] and Document Explorer [11]. Their approach, however, requires prior labeling (through some combination of manual and automated methods) using keywords from a given ontology, and cannot directly be used on general text. DuMouchel and Predigibon [9] propose a statistically motivated metric, and apply empirical Bayes methodology for mining associations in text. Their work has similar motivation to ours. The authors do not report on efficiency and scalability issues.

Statistical natural language processing: The problem of finding associations between words (often referred to as *collocations*) has been studied extensively in the field of Statistical Natural Language Processing (SNLP) [17]. We briefly review some of this literature here, but expand in Section 3.1 on why these measures fail to address our needs.

Frequency is often used as a measure of interestingness, together with a part-of-speech filter to discard syntactic collocations like "of the". Another standard practice is to apply some statistical test that, given a pair of words, evaluates the null hypothesis that this pair is generated by picking two words independently at random. The interestingness of the pair is measured by the deviation from the null hypothesis. The t test and the χ^2 test are statistical tests frequently used in SNLP. There is a qualitative difference between collocations and the associations that we are interested in. Collocations include patterns of words that tend to appear together (e.g. phrasal verbs – "make up", or common expressions like "strong tea"), while we are mostly interested in associations that convey some latent concept (e.g. "chapters indigo" – this pertains to the recent acquisition of Chapters, then Canada's largest bookstore, by the Indigo corporation).

2.2 Main contributions and guided tour

1. We develop a notion of semantic as opposed to syntactic text associations, together with a statistical measure that mines such associations (Section 3.3). We point out that simple statistical frequency measures such as the χ^2 test and mutual information (as well as variants) will not suffice (Section 3.2).

2. Our measure for associations lacks the monotonicity and closure properties exploited by prior work in association mining. We therefore require novel pruning techniques to achieve scalable mining. To this end we propose two new techniques: (i) matrix mining (Section 4.2) and (ii) shortened documents (Section 4.3).

3. We analyze the pruning resulting from these techniques. A novel aspect of this analysis: to our knowledge, it is the first time that the Zipfian distribution of terms and pairs is used in the *analysis* of mining algorithms. We combine these pruning techniques into two algorithms (Section 4 and Theorem 1).

4. We give results of experiments on three test corpora for the pruning achieved in practice. These results suggest that the pruning is more efficient than our (conservative) analytical prediction and that our methods should scale well to larger corpora (Section 4.4).

We report results on three test corpora taken from news agencies: the CBC corpus, the CNN corpus and the Reuters corpus. More statistics on the corpora are given in Section 4.4.

3 Statistical basis for associations

In this section we develop our measure for significant associations. We begin (Section 3.1) by discussing qualitatively the desiderata for significant text associations. Next, we give a detailed study of pair occurrences in our test corpora (Section 3.2). Finally, we bring these ideas together in Section 3.3 to present our new measure for interesting associations.

3.1 Desiderata for significant text associations

We first experimented with naive support measures such as document pair frequency, sentence pair frequency and the product of the individual sentence term frequencies. We omit the detailed results here due to space constraints. As expected, the highest ranking associations are mostly *syntactic* ones, such as (of,the) and (in,the), conveying little information about the dominant concepts. Furthermore, it is clear that the document level is too granular to mine useful associations – two terms could co-occur in many documents for template (rather than semantic) reasons; for example, associations such as (business, weather), and (corporate, entertainment) in the CBC corpus.

We also experimented with well known measures from SNLP such as the χ^2 test and mutual information as well as the *conviction* measure, a variation of the well known confidence measure defined in [6]. We modified the measure slightly so that it is symmetric. Table 1 shows the top associations for the CNN corpus for these measures. The number next to each pair indicates the number of sentences in

rank	χ^2	conviction	mutual information	weighted MI
1	afghani libyan :2	afghani libyan :2	allowances child-care :1	of the :40073
2	antillian escudo :2	antillian escudo :2	alanis morissette :1	the to :41504
3	algerian angolan :2	algerian angolan :2	americanas marisa :1	in the :34750
4	allowances child-care :1	allowances child-care :1	charming long-stem :1	click here :13594
5	alanis morissette :1	alanis morissette :1	cane stalks :1	and the :30397
6	arterial vascular :2	arterial vascular :2	hk$116.50 hk$53.50 :1	a the :32088
7	americanas marisa :1	americanas marisa :1	ill.,-based pyrex :1	a to :28211
8	balboa rouble :2	balboa rouble :2	boston.it grmn :1	call market :11061
9	bolivian lesotho :2	bolivian lesotho :2	barbed inventive :1	latest news :11740
10	birr nicaraguana :2	birr nicaraguan :2	160kpns telias :1	a of :23362

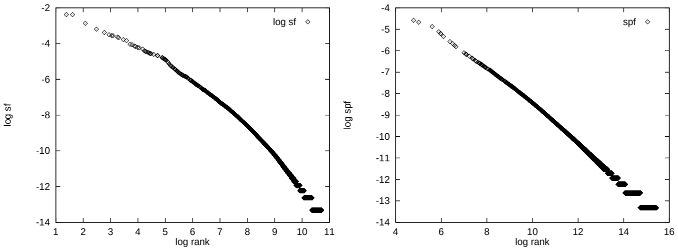
Table 1. Top associations from the CNN corpus under different measures.

which this pair appears. Although these measures avoid syntactic associations, they emphasize on pairs of words with very low sentence frequency. If two words t and q appear only a few times but they always appear in the same sentence, then the pair $\{t, q\}$ scores highly for all of these measures, since it deviates significantly from the independence assumption. This is especially true for the mutual information measure [17]. We also experimented with a weighted version of the mutual information measure [17], where we weight the mutual information of a pair by the sentence frequency of the pair. However, in this case the weight of the sentence pair frequency dominates the measure. As a result, the highly ranked associations are syntactic ones.

It appears that any statistical test that compares against the independence hypothesis (such as the χ^2 test, the t test, or mutual information) falls prey of the same problem: it favors associations of low support. One might try to address this problem by applying a pruning step before computing the various measures: eliminate all pairs that have sentence pair frequency below a predefined threshold. However, this approach just masks the problem. The support threshold directly determines the pairs that will be ranked higher.

3.2 Statistics of term and pair occurrences

We made three measurements for each of our corpora: the distributions of *corpus term frequencies* (the fraction of all words in the corpus that are term t), *sentence term frequencies* (fraction of sentences containing term t) and *document term frequencies* (fraction of documents containing term t). We also computed the distribution of the *sentence pair frequencies* (fraction of sentences that contain a pair of terms). We observed that the Zipfian distribution essentially holds, not only for corpus frequencies but also for document and sentence frequencies, as well as for sentence pair frequencies. Figure 1 presents the sentence term frequencies and the sentence pair frequencies for the CNN corpus. We use these observations for the analysis of the pruning algorithms in Section 4. The plots for the other test corpora are essentially the same as those for CNN.



(a) Sentence Term Frequencies (b) Sentence Pair Frequencies

Figure 1. Statistics for the CNN corpus

3.3 The new measure

Intuitively we seek pairs of terms that co-occur frequently in sentences, while eliminating pairs resulting from very frequent terms. This bears a strong analogy to the concept of weighting term frequencies by *inverse document frequency* (idf) in text indexing.

Notation: Given a corpus of documents C, let N_d denote the number of documents in C, let N_s denote the number of sentences in C and let N_t denote the the number of distinct terms in C. For a set of terms $T = \{t_1, t_2, \ldots, t_k\}$, for $k \geq 1$, let $n_d(t_1, t_2 \ldots, t_k)$ denote the number of documents in C that contain all terms in T and let $n_s(t_1, t_2, \ldots, t_k)$ denote the number of sentences in C that contain all terms in T. We define the *document frequency* of T as $df(t_1, t_2, \ldots, t_k) = n_d(t_1, t_2 \ldots, t_k)/N_d$, and the *sentence frequency* of the set T as $sf(t_1, t_2, \ldots, t_k) = n_s(t_1, t_2, \ldots, t_k)/N_s$. If $k = 2$, we will sometimes use dpf and spf to denote the document and sentence pair frequencies. For a single term t, we define the inverse document frequency of t, $idf(t) = \log(N_d/n_d(t))$ and the inverse sentence frequency $isf(t) = \log(N_s/n_s(t))$. In typical applications the base of the logarithm is immaterial since it is the relative values of the idf that matter. The particular formula for idf owes its intuitive justification to the underlying Zipf distribution on terms; the reader is referred to [17, 21] for details.

Based on the preceding observations, the following idea

rank	$spf \times idf \times idf$	$spf \times isf \times isf$	$dpf \times idf \times idf$	$\log(spf) \times idf \times idf$
1	deutsche telekom	click here	danmark espaol	conde nast
2	hong kong	of the	espaol svenska	mph trains
3	chevron texaco	the to	danmark svenska	allegheny lukens
4	department justice	in the	espaol travelcenter	allegheny teledyne
5	mci worldcom	and the	danmark travelcenter	newell rubbermaid
6	aol warner	a the	svenska travelcenter	hummer winblad
7	aiff wav	call market	espaol norge	hauspie lernout
8	goldman sachs	latest news	danmark norge	bethlehem lukens
9	lynch merrill	a to	norge svenska	globalstar loral
10	cents share	a of	norge travelcenter	donuts dunkin

Table 2. Top associations for variants of our measure for the CNN corpus.

suggests itself: weight the frequency of a pair by the (product of the) idf's of the constituent terms. The generalization beyond pairs to k-tuples is obvious. We state below the formal definition of our new measure for arbitrary k.

Definition 1 *For terms $t_1, t_2, \ldots, t_k$, the measure for the association $\{t_1, t_2, \ldots, t_k\}$ is*

$$m_k(t_1, t_2, \ldots, t_k) = sf(t_1, t_2, \ldots, t_k) \times \prod_{j=1}^{k} idf(t_j) \; .$$

Variants of the measure: We experimented with several variants of our measure and settled on using idf rather than isf, and spf rather than dpf. Table 2 gives a brief summary from the CNN corpus to give the reader a qualitative idea. Replacing idf with isf introduces more syntactical associations. This is due to the fact that the sentence frequency of words like "the" and "of" is lower than their document frequency, so the impact of the isf as a dampening factor is reduced. This allows the sentence frequency to take over. A similar phenomenon occurs when we replace spf with dpf. The impact of dpf is too strong, causing uninteresting associations to appear. We also experimented with $\log(spf)$, an idea that we plan to investigate further in the future.

Figure 2 shows two plots of our new measure. The first is a scatter plot of our measure (which weights the spf's by idf's) versus the underlying spf values[1]. The line $y = x$ is shown for reference. We also indicate the horizontal line at threshold 0.002 for our measure; points below this line are the ones that "succeed". Several intuitive phenomena are captured here. (1) Many frequent sentence pairs are attenuated (moved upwards in the plot) under our measure, so they fail to exceed the threshold line. (2) The pairs that do succeed are "middling" under the raw pair frequency. The plot on the right shows the distribution of our measure, in a log-log plot, suggesting that it in itself is roughly Zipfian; this requires further investigation. If this is indeed the case then we can apply the theoretical analysis of Section 4.1 to the case of higher order associations.

Non-monotonicity: A major obstacle in our new measure: weighting by idf can increase the weight of a pair with low sentence pair frequency. Thus, our new measure does not enjoy the *monotonicity property* of the support measure exploited by the *a priori* algorithms. Let I be some measure of interestingness that assigns a value $I(T)$ to every possible set of of terms T. We say that I is monotone if the following holds: if $T' \subseteq T$, then $I(T') \geq I(T)$. This property allows for pruning, since if for some $T' \subseteq T$, $I(T') \leq \theta$, then $I(T) \leq \theta$. That is, all interesting sets must be the union of interesting subsets. Our measure does not enjoy this property. For some pair of terms $\{t_1, t_2\}$, it may be the case that $m_2(t_1, t_2) > \theta$, while $m_1(t_1) \leq \theta$, or $m_1(t_2) \leq \theta$.

Formal problem statement: Given a corpus and a threshold θ, find (for $k = 2, 3, \ldots$) all k-tuples for which our measure exceeds θ.

4 Fast extraction of associations

We now present two novel techniques for efficiently mining associations deemed significant by our measure: *matrix mining* and *shortened documents*. Following this, we analyze the efficiencies yielded by these techniques and give experiments corroborating the analysis. We first describe how to find all pairs of terms $\{x, y\}$ such that the measure $m(x, y) = spf(x, y) idf(x) idf(y)$ exceeds a prescribed threshold θ. We also show how our techniques generalize for arbitrary k-tuples.

4.1 Pruning

Although our measure is not monotone we can still explore some monotonicity properties to apply pruning. We observe that

$$m(x, y) = spf(x, y) idf(x) idf(y) \leq sf(x) idf(x) idf(y) \; . \tag{1}$$

Let $q(x) = sf(x) idf(x)$ and $f(y) = idf(y)$. The value of $f(y)$ cannot exceed $\log N_d$. Therefore, $m(x, y) \leq$

[1] The axes are scaled and labeled negative logarithmically, so that the largest values are to the bottom left and the smallest to the top and right.

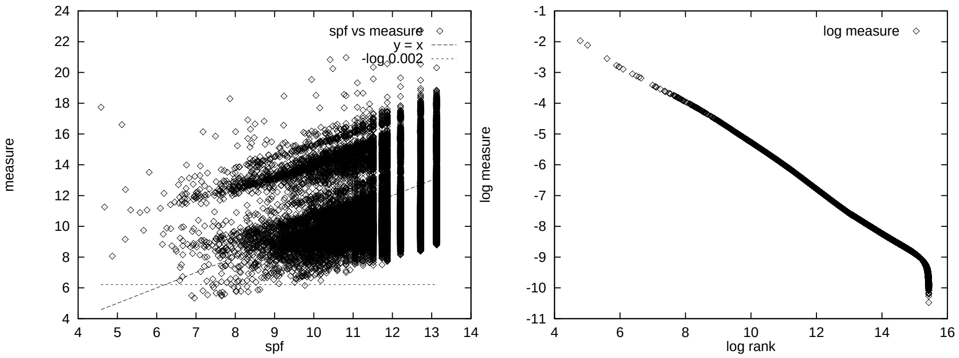


Figure 2. The new measure

$q(x)f(y) \leq q(x)\log N_d$. Thus, we can safely eliminate any term x for which $q(x) \leq \theta/\log N_d$. We observe experimentally that this results in eliminating a large number of terms that appear in just a few sentences. We will refer to this pruning step as *low end pruning* since it eliminates terms of low frequency.

Equation 1 implies that if $m(x,y) > \theta$, then $q(x)f(y) > \theta$. Therefore, we can safely eliminate all terms y such that $f(y) \leq \theta/\max_x q(x)$. We refer to this pruning step as *high end pruning* since it eliminates terms of high frequency. Although this step eliminates only a small number of terms, it eliminates a large portion of the text.

We now invoke additional information from our studies of sentence term frequency distributions in Section 3.2 to estimate the number of terms that survive low end pruning.

Theorem 1 *Low end pruning under a power law distribution for term frequencies eliminates all but $O(\log^2 N_d)$ terms.*

Proof: The sf values are distributed as a power law: the ith-largest frequency is proportional to $1/i^{\alpha}$. If t_i denotes the ith most frequent term, $sf(t_i) = A/i^{\alpha}$ for a constant A. Since no idf value exceeds $\log N_d$, we have $q(t_i) = sf(t_i)idf(t_i) \leq A\log N_d/i^{\alpha}$. If $q(t_i) > \theta/\log N_d$, then $\theta < A\log^2 N_d/i^{\alpha}$. Therefore, $i < (A/\theta)^{1/\alpha}\log^{2/\alpha} N_d$. Let $c = (A/\theta)^{1/\alpha}$ and $\beta = 2/\alpha$. If $B = c\log^{\beta} N_d$, then only $O(B)$ terms can generate candidate pairs. Since $\alpha \geq 1$, $O(B) = O(\log^2 N_d)$. ■

Pruning extends naturally to k-tuples. A k-tuple can be thought as a pair consisting of a single term and a $(k-1)$-tuple. Since $m_k(t_1,\ldots,t_k) \leq m_{k-1}(t_1,\ldots,t_{k-1})idf(t_k)$, we can safely prune all $(k-1)$-tuples such that $m_{k-1}(t_1,\ldots,t_{k-1}) \leq \theta/\log N_d$. Proceeding recursively we can compute the pruning threshold for i-tuples and apply pruning in a bottom up fashion (terms, pairs, and so on). We define $\theta_i = \theta/\log^{k-i} N_d$ to be the threshold for i-tuples for all $1 \leq i \leq k$.

4.2 Matrix mining

Given the terms that survive pruning we now want to minimize the number of pairs for which we compute the $spf(x,y)$ value. Let N_t' denote the number of (distinct) terms that survive pruning. The key observation is best visualized in terms of the matrix depicted in Figure 3(left). It has N_t' rows and N_t' columns, one for each term. The columns of the matrix are arranged left-to-right in non-increasing order of the values $q(x)$ and the rows bottom-up in non-increasing order of the values $f(x)$. Let q_i denote the ith largest value of $q(x)$ and f_j denote the jth largest value of $f(x)$. Imagine that matrix cell (i,j) is filled with the product $q_i f_j$ (we do not actually *compute* all of these values).

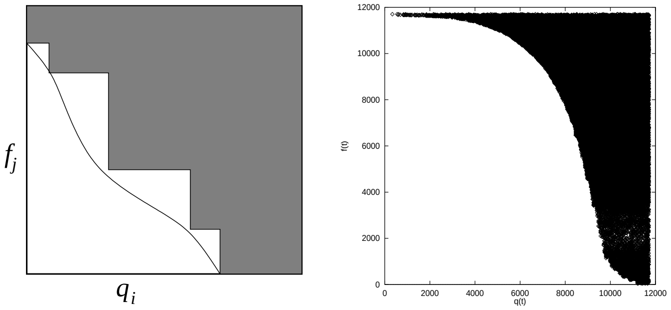


Figure 3. Matrix mining

The next crucial observation: by Equation 1 the pair (i,j) is eliminated from further consideration if the entry in cell (i,j) is less than θ. This elimination can be done especially efficiently by noting a particular structure in the matrix: entries are non-increasing along each row and up each column. This means that once we have found an entry that is below the threshold θ, we can immediately eliminate all entries above and to its right, and not bother computing those entries (Figure 3). We have such a "upper-right" rectangle in each column, giving rise to a frontier (the curved line in the left

```
MATRIX-WAM($\theta$)

(1) Collect Term Statistics
(2) $T \leftarrow$ Apply pruning; $n \leftarrow |T|$
(3) $X \leftarrow$ sort $T$ by $sf \times idf$ in decreasing order
(4) $Y \leftarrow$ sort $T$ by $idf$ in decreasing order
(5) For $y = Y[0]$ to $Y[n]$
(6)     For $x = X[0]$ to $X[n]$
(7)         if $x$ has not been considered already
(8)             if $sf(x) \times idf(x) \times idf(y) > \theta$
(9)                 Compute $spf(x, y)$
(10)                if $spf(x, y) \times idf(x) \times idf(y) > \theta$
(11)                    Add $\{x, y\}$ to answer set $A$
(12)            else discard all terms right of $x$; break
(13) return $A$
```

Figure 4. The MATRIX-WAM algorithm

figure) between the eliminated pairs and those remaining in contention. For cells remaining in contention, we proceed to the task of computing their spf values, computing $m(x, y)$, and comparing with θ. Applying Theorem 1 we observe that there are at most $O(log^4 N_d)$ candidate pairs. In practice our algorithm computes the spf values for only a fraction of the $\binom{N_t'}{2}$ candidate pairs. Figure 3 (right) illustrates the frontier line for the CNN corpus.

We now introduce the first Word Associations Mining (WAM) algorithm. The MATRIX-WAM algorithm shown in Figure 4.2 implements matrix mining. The first step makes a pass over the corpus and collects term statistics. The pruning step performs both high and low end pruning, as described in Section 4.1. For each term we store an *occurrence list* keeping all sentences the term appears in. For a pair $\{x, y\}$ we can compute the $spf(x, y)$ by going through the occurrence lists of the two terms. Lines (8)-(12) check the column frontier and determine the pairs to be stored.

For higher order associations, the algorithm performs multiple matrix mining passes. In the ith pass, one axis of the matrix holds the idf values as before, and the other axis the m_{i-1} values of the $(i-1)$-tuples that survived the previous pass. We use threshold θ_i for the ith pass

4.3 Shortened documents

While matrix mining reduces the computation significantly, there are still many pairs for which we compute the spf value. Furthermore, for most of these pairs the spf value is actually zero, so we end up examining many more pairs than the ones that actually appear in the corpus. We invoke a different approach, similar to the AprioriTID algorithm described by Agrawal and Srikant [2]. Let H_1 denote the set of terms that survive the pruning steps described in Section 4.1 – we call these the *interesting* terms. Given H_1 we make a second pass over the corpus, keeping a counter for each pair of *interesting* terms that appear together in a sentence. That is, we replaced each document by a *shortened document* consisting only of the terms deemed interesting.

```
SHORT-WAM($k$,$\theta$)

Collect Term Statistics.
$H_1 \leftarrow$ Prune Terms; $C_1 \leftarrow$ Corpus
For $i = 2$ to $k$
    For each sentence $s$ in $C_{i-1}$
        $I_s = (i-1)$-tuples in $s$ that are in $H_{i-1}$
        $s' = i$-tuples generated by joining $I_s$ with itself
        Add tuples in $s'$ to $H_i$
        if $s' \neq \emptyset$ Add $s'$ to $C_i$
    $H_i \leftarrow$ apply pruning on $H_i$.
```

Figure 5. The SHORT-WAM algorithm

The shortened documents algorithm extends naturally for higher order associations (Figure 4.3). The algorithm performs multiple passes over the data. The input to the ith pass is a corpus C_{i-1} that consists of sentences that are sets of $(i-1)$-tuples and a hash table H_{i-1} that stores all interesting $(i-1)$-tuples. An i-tuple t is interesting if $m_i(t) > \theta_i$. During the ith pass the algorithm generates candidate i-tuples by joining interesting $(i-1)$-tuples that appear together in a sentence. The join operation between $(i-1)$-tuples is performed as in the case of the a priori algorithms [2]. The candidates are stored in a hash H_i and each sentence is replaced by the candidates it generates. At the end of the pass, the algorithm outputs a corpus C_i that consists of sentences that are collections of i-tuples. Furthermore, we apply low end pruning to the hash table H_i using threshold θ_i. At the end of the pass H_i contains the interesting i-tuples.

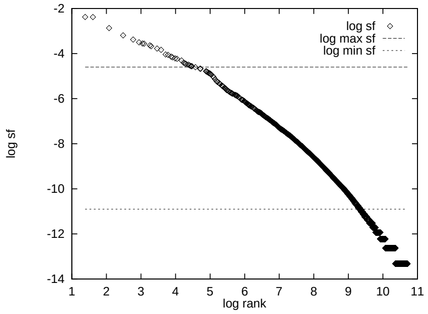


Figure 6. Pruned Terms for CNN corpus

4.4 Empirical study of WAM algorithms

We ran our two algorithms on our three corpora, applying both high and low end pruning. Figure 6 shows a plot of how the thresholds are applied. The terms that survive pruning correspond to the area between the two lines in the plot. The top line in the figure was determined by high end pruning,

		CBC	CNN	Reuters
Corpus Statistics				
1	distinct terms	16.5K	44.7K	37.1K
2	corpus terms	471K	3.6M	1.3M
3	distinct sp's	1.2M	5M	3.7M
4	corpus sp's	3.9M	28.8M	16.3M
Pruning Statistics				
5	threshold	0.002	0.001	0.015
6	pruned	9.6K (58%)	33.2K (74%)	31.4K (84%)
7	high pruned	20	57	0
8	collected	2,798	3,006	2,699
MATRIX-WAM Statistics				
9	naive pairs	23.8M	66.2M	16.2M
10	computed spf's	19.1M (80%)	47M (70%)	9.2M (57%)
11	zero spf	22.5M	60.6M	13.6M
SHORT-WAM Statistics (w/o high pruning)				
12	pruned corpus terms	45K (10%)	0.2M (5%)	0.1M (7%)
13	gen sp's	3.5M (91%)	26.6M (92%)	14.1M (86%)
14	distinct sp's	963K (77%)	3.6M (72%)	2.1M (57%)
SHORT-WAM Statistics (with high pruning)				
15	pruned corpus terms	134K (29%)	1.2M (32%)	0.1M (7%)
16	gen sp's	2.4M (60%)	16.3M (56%)	14.1M (86%)
17	distinct sp's	898K (72%)	3.3M (67%)	2.1M (57%)

Table 3. Statistics for the WAM algorithms

while the bottom line was determined by low end pruning.

Table 3 shows the statistics for the two algorithms when mining for pairs for all three corpora. In the table sp stands for sentence pair and corpus sp's is the total number of sentence pairs in the corpus. We count the appearance of a term in a sentence only once. In all cases we selected the threshold so that around 3,000 associations are collected (line 8). Pruning eliminates at least 58% of the terms and as much as 84% for the Reuters corpus (line 6). Most terms are pruned from the low end of the distribution; high end pruning removes just 20 terms for the CBC corpus, 57 for the CNN corpus and none for the Reuters corpus (line 7). The above observations indicate that our theoretical estimates for pruning may be too conservative. To study how pruning varies with corpus size we performed the following experiment. We sub-sampled the CNN and Reuters corpora, creating synthetic collections with sizes $N_d = 2^8, 2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}$. For each run, we selected the threshold so that the percentage of pairs above the threshold (over all distinct pairs in the corpus) is approximately the same for all runs. The results are shown in Figure 7. The x axis is the log of the corpus size, while the y axis is the fraction of terms that were pruned.

Matrix mining improves the performance significantly: compared to the naive algorithm that computes the spf values for all $\binom{N'_t}{2}$ pairs of the terms that survive pruning (line 9), the MATRIX-WAM algorithm computes only a fraction of these (maximum 80%, minimum 57%, line 10). Note however that most of the spf's are actually zero (line 11).

The SHORT-WAM algorithm considers only (a fraction of) pairs that actually appear in the corpus. To study the importance of high end pruning we implemented two versions of SHORT-WAM, one that applies high end pruning and one that does not. In the table, lines 12 and 15 show the percentage of the corpus terms that are pruned, with and without high end pruning. Obviously, high end pruning is responsible for most of the removed corpus. For the CNN corpus, the 57 terms removed due to high end pruning cause 28% of the corpus to be removed.

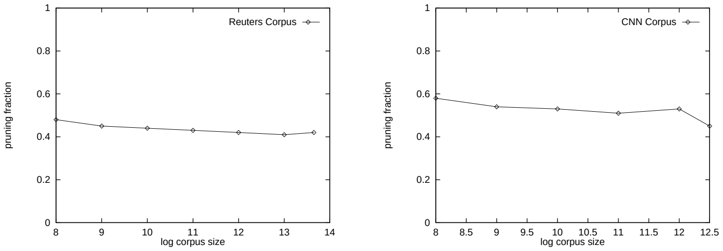


Figure 7. Pruning for Reuters and CNN corpus

The decrease is even more impressive when we consider the pairs generated by SHORT-WAM (lines 13, 16). For the CNN corpus, the algorithm generates only 56% of all possible corpus sp's (ratio of lines 4 and 16). This decrease becomes more important when we mine higher order tuples, since the generated pairs will be given as input to the next iteration. Again high end pruning is responsible for most of the pruning of the corpus sp's. Finally, our algorithm generates at most 72% of all possible *distinct* sentence pairs (line 17). These pairs are stored in the hash table and they reside in main memory while performing the data pass: it is important to keep their number low. Note that AprioriTID generates all pairwise combinations of the terms that survived pruning (line 9).

	CBC	CNN	Reuters
threshold	0.006	0.003	0.03
pruned terms	39%	53%	56%
computed spf's	50.4M	212M	129M
generated sp's	13,757	17,547	64,513
computed stf's	79.3M	203M	659M
collected	2,970	3,213	3,258

Table 4. MATRIX-WAM for triples

We also implemented the algorithms for higher order tuples. Table 4 shows the statistics for MATRIX-WAM, for triples. Clearly we still obtain significant pruning. Furthermore, the volume of sentence pairs generated is not large, keeping the computation in control.

We implemented SHORT-WAM for k-tuples, for arbitrarily large k. In Figure 8 we plot, as a function of the iteration number i, the size of the corpus C_i (figure on the left), as well as the number of candidate tuples and the number of these tuples that survived each pruning phase (figure on the right). The threshold is set to 0.07 and we mine 8,335 5-tuples. Although the sizes initially grow significantly, they fall fast at subsequent iterations. This is consistent with the observations in [2].

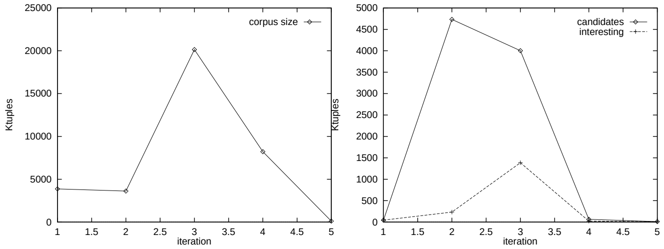


Figure 8. Statistics for SHORT-WAM

4.5 Sample associations

At http://www.cs.toronto.edu/∼tsap/TextMining/ there is a full list of the associations. Table 5 shows a sample of associations from all three corpora that attracted our interest.

Pairs
deutsche telekom, hong kong, chevron texaco, department justice, mci worldcom, aol warner, france telecom, greenspan tax, oats quaker, chapters indigo, nestle purina, oil opec, books indigo, leaf maple, states united, germany west, arabia saudi, gas oil, exxon jury, capriati hingis

Triples
chateau empress frontenac, indigo reisman schwartz, del monte sun-rype, cirque du soleil, bribery economics scandal, fuel spills tanker, escapes hijack yemen, al hall mcguire, baker james secretary, chancellor lawson nigel, community ec european, arabia opec saudi, chief executive officer, child fathering jesse, ncaa seth tournament, eurobond issuing priced, falun gong self-immolation, doughnuts kreme krispy, laser lasik vision, leaf maple schneider

Table 5. Sample associations

5 Conclusions

In this paper, we introduced a new measure of interestingness for mining word associations in text, and we proposed new algorithms for pruning and mining under this (non-monotone) measure. We provided theoretical and empirical analyses of the algorithms. The experimental evaluation demonstrates that our measure produces interesting associations, and our algorithms perform well in practice. We are currently investigating applications of our pruning techniques to other non-monotone cases. Furthermore, we are interested in examining if the analysis in Section 4.1 can be applied to other settings.

References

[1] R. Agrawal, T. Imielinski, A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. SIGMOD 1993.

[2] R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. VLDB 1994.

[3] H. Ahonen, O. Heinonen, M. Klemettinen, A. Inkeri Verkamo. Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Document Collections. ADL 1998.

[4] R. Bayardo, R. Agrawal, D. Gunopulos, Constraint-based rule mining in large, dense databases. ICDE, 1999.

[5] S. Brin, R. Motwani, J. D. Ullman, S. Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. SIGMOD 1997.

[6] S. Brin, R. Motwani, C. Silverstein. Beyond Market Baskets: Generalizing Association Rules to Correlations. SIGMOD 1997.

[7] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. Ullman, C. Yang, Finding Interesting Associations without Support Pruning, ICDE 2000.

[8] D.R. Cutting, D. Karger, J. Pedersen and J.W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. 15th ACM SIGIR, 1992.

[9] W. DuMouchel and D. Pregibon, Empirical Bayes Screening for Multi-Item Associations, KDD 2001.

[10] R. Feldman, I. Dagan and W. Klosgen. Efficient algorithms for mining and manipulating associations in texts. *13th European meeting on Cybernetics and Systems Research*, 1996.

[11] R. Feldman, W. Klosgen and A. Zilberstein. Document explorer: Discovering knowledge in document collections. *10th International Symposium on Methodologies for Intelligent Systems*, Springer-Verlag LNCS 1325, 1997.

[12] R. Feldman, I. Dagan, H. Hirsh. Mining text using keyword distributions. *Journal of Intelligent Information Systems* 10, 1998.

[13] B. Lent, R. Agrawal and R. Srikant. Discovering trends in text databases. KDD, 1997.

[14] D.D. Lewis and K. Sparck Jones. Natural language processing for information retrieval. *Communications of the ACM* 39(1), 1996, 92–101.

[15] A. J. Lotka. The frequency distribution of scientific productivity. *J. of the Washington Acad. of Sci.*, 16:317, 1926.

[16] H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. KDD, 1996.

[17] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*, 1999. The MIT Press, Cambridge, MA.

[18] E. Riloff. Little words can make a big difference for text classification. 18th ACM SIGIR, 1995.

[19] F. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics* 19(1), 1993, 143–177.

[20] G. Webb, Efficient Search for association rules, KDD, 2000.

[21] I. Witten, A.Moffat and T. Bell. *Managing Gigabytes*. Morgan Kaufman, 1999.

[22] G. K. Zipf. Human behavior and the principle of least effort. *New York: Hafner*, 1949.