# Space Efficiencies in Discourse Modeling via Conditional Random Sampling

**Brian Kjersten**
CLSP
Johns Hopkins University

**Benjamin Van Durme**
HLTCOE
Johns Hopkins University

## Abstract

Recent exploratory efforts in discourse-level language modeling have relied heavily on calculating Pointwise Mutual Information (PMI), which involves significant computation when done over large collections. Prior work has required aggressive pruning or independence assumptions to compute scores on large collections. We show the method of Conditional Random Sampling, thus far an underutilized technique, to be a space-efficient means of representing the sufficient statistics in discourse that underly recent PMI-based work. This is demonstrated in the context of inducing Shankian *script*-like structures over news articles.

## 1 Introduction

It has become common to model the distributional affinity between some word or phrase pair, $(w_i, w_j)$, as a function of co-occurance within some *context* boundary. Church and Hanks (1990) suggested pointwise mutual information: $\text{PMI}(w_i, w_j) = \log \frac{\Pr(w_i, w_j)}{\Pr(w_i)\Pr(w_j)}$, showing linguistically appealing results using contexts defined by fixed width n-gram windows, and syntactic dependencies derived from automatically parsed corpora. Later work such as by Lin (1999) continued this tradition. Here we consider document, or *discourse*-level contexts, such as explored by Rosenfeld (1994) or Church (2000), and more recently by those such as Chambers and Jurafsky (2008) or Van Durme and Lall (2009b).

In the spirit of recent work in randomized algorithms for large-scale HLT (such as by Ravichandran et al. (2005), Talbot and Osborne (2007), Goyal et al. (2010), Talbot and Brants (2008),Van Durme and Lall (2009a), Levenberg and Osborne (2009), Goyal et al. (2010), Petrovic et al. (2010), Van Durme and Lall (2010), or Goyal and Daumé (2011)), we propose the method of Conditional Random Sampling (CRS) by Li and Church (2007) as an efficient way to store *approximations* of the statistics used to calculate PMI for applications in inducing rudimentary *script*-like structures.

Efficiently storing such structures is an important step in integrating document-level statistics into downstream tasks, such as characterizing complex scenarios (Chambers and Jurafsky, 2011), or story understanding (Gordon et al., 2011).

## 2 Background

**Conditional Random Sampling (CRS)** Li and Church (2007) proposed CRS to approximate the contingency table between elements in a query, to be used in distributional similarity measures such as cosine similarity, correlation, and PMI. Central is the idea of the *postings list*, which is made up of the identifiers of each document that contains a given word or phrase. A set of such lists, one per type in the underlying vocabulary, is known as an *inverted index*. To reduce storage costs, a CRS truncates these lists, now called *sketches*, such that each sketch is no larger than some length parameter $k$.

Formally, assume an ordered list of document identifiers, $\Omega = (1, 2, ...)$, where each referenced document is a bag of words drawn from a vocabulary of size $V$. Let $P_i \subseteq \Omega$ be the postings list for some element $w_i \in V$. The function $\pi$ represents a random permutation on the space of identifiers in $\Omega$. The sketch, $S_i$, is defined as the first $k$ elements of the permuted list: $S_i = \min_k(\pi(P_i))$. [1]

Let $q$ be a two-element *query*, $(w_i, w_j)$. Given the postings lists for $w_i, w_j$, we can construct a four-cell *contingency table* containing the frequency of documents that contained only $w_i$, only $w_j$, both together, or neither. A CRS allows for approximating this table in $O(k)$ time by relying on a sample of $\Omega$, specific to $q$: $\pi(\Omega)_q = (1, 2, 3, ..., min(max(S_i), max(S_j)))$.

The PMI of $q$, given $\Omega$, can be estimated from $\pi(\Omega)_q$ using the approximate word occurrence, $\Pr(w_i) = |S_i \cap \pi(\Omega)_q|/|\pi(\Omega)_q|$, and co-occurrence, $\Pr(w_i \cap w_j) = |S_i \cap S_j \cap \pi(\Omega)_q|/|\pi(\Omega)_q|$.

This scheme generalizes to longer queries of length $m$, where storage costs remain $O(Vk)$, and query time scales at $O(mk)$. Li and Church (2007) proved that CRS produces an unbiased estimate of the probabilities, and showed empirically that variance is a function of $k$ and $m$.

Despite its simplicity and promise for large-scale data mining in NLP, CRS has thus-far seen minimal application in the community.

**Trigger Language Models** As here, Rosenfeld (1994)'s work on trigger language models was concerned with document level context. He identified trigger pairs: pairs of word sequences where the presence of the first word sequence affects the probability of the other, possibly at long distances. He recommended selecting a small list of trigger pairs based on the highest *average* mutual information (often simply called mutual information), although intuitively PMI could also be used. Computational constraints forced him to apply heavy pruning to the bigrams in his model.

**Scripts** A *script*, proposed by Schank (1975), is a form of Minsky-style *frame* that captures commonsense knowledge regarding typical events. For example, if a machine were to reason about *eating at a restaurant*, it should associate to this event: the existence of a *customer* or *patron* that usually *pays* for the *meal* that is *ordered* by the patron, then *served* by the *waiter*, etc.

Chambers and Jurafsky (2008) suggested inducing a similar structure called a *narrative chain*: focus on the situational descriptions explicitly pertaining to a single *protagonist*, a series of references within a document that are automatically labeled as coreferent. With a large corpus, one can then find those sets of verbs (as anchors of basic situational descriptions) which tend to co-occur, and share a protagonist, leading to an approximate subset of Schank's original conception.[2]

Underlying the co-occurrence framework of Chambers and Jurafsky was finding those verbs with high PMI. Starting with some initial element, chains were built greedily by adding the term, $x$, that maximized the average of the pairwise PMI between $x$ and every term already in the chain:

$$W_{n+1} = \arg\max_{W} \frac{1}{n} \sum_{j=1}^{n} \text{pmi}(W, W_j)$$

By relying on the average pairwise PMI, they are making independence assumptions that are not always valid. In order to consider more nuanced joint effects between more than two terms, more efficient methods would need to be considered.

## 3 Experiments

**Setup** Following Chambers and Jurafsky (2008), we extracted and lemmatized the verbs from the New York Times section of the Gigaword Corpus using the Stanford POS tagger (Toutanova et al., 2004) and the Morpha lemmatizer (Minnen et al., 2000). After filtering various POS tagger errors and setting a minimum document frequency (df) of 50, we went from a vocabulary of 94,803 words to 8,051.[3] For various values of $k$ we built sketches over 1,655,193 documents, for each resulting word type.

[1]For example, assume some word $w_i$ that appears in documents $d_1, d_4, d_{10}$ and $d_{12}$. The identifiers are then randomly permuted via $\pi$ such that: $d'_3 = d_1$, $d'_2 = d_4$, $d'_7 = d_{10}$ and $d'_1 = d_{12}$. Following permutation, the postings list for $w_i$ is made up of identifiers that map to the same underlying documents as before, but now in a different order. If we let $k = 3$, then $S_i = (1, 2, 3)$, corresponding to documents: $(d_{12}, d_4, d_1)$.

[2]Given a large collection of news articles, some on the topic of local crime, one might see a story such as: *"... searched for Michael$_i$ ... he$_i$ was arrested ... Mike$_i$ plead guilty ... convicted him$_i$ ..."*, helping to support an induced chain: (*search, arrest, plead, acquit, convict, sentence*).

[3]Types containing punctuation other than hyphens and underscores were discarded as tagger-error.

Table 1: Top-$n$ by approximate PMI, for varying $k$. Subscripts denote rank under true PMI, when less than 50.

| | plead | | | plead, admit | | plead, admit, convict | |
|---|---|---|---|---|---|---|---|
| 1 | sentence$_4$ | sentence$_4$ | sentence$_4$ | abuse$_-$ | sentence$_5$ | owe$_-$ | sentence$_2$ |
| 2 | commit$_-$ | defraud$_5$ | misbrand$_2$ | convict$_{22}$ | prosecute$_{15}$ | admitt$_{11}$ | prosecute$_3$ |
| 3 | indict$_{10}$ | indict$_{10}$ | defraud$_5$ | owe$_-$ | testify$_{20}$ | engage$_-$ | arrest$_8$ |
| 4 | prosecute$_{33}$ | arraign$_6$ | arraign$_6$ | investigate$_-$ | indict$_{10}$ | investigate$_{28}$ | testify$_5$ |
| 5 | abuse$_-$ | conspire$_{11}$ | manslaughter$_1$ | understand$_-$ | defraud$_7$ | prey$_-$ | acquit$_1$ |
| 6 | convict$_{24}$ | convict$_{24}$ | bilk$_8$ | defraud$_7$ | convict$_{22}$ | defraud$_-$ | indict$_4$ |
| $k =$ | 100 | 1,000 | 10,000 | 1,000 | 10,000 | 1,000 | 10,000 |

We use a generalized definition of PMI for three or more items as the logarithm of the joint probability divided by the product of the marginals.

**Subjective Quality** We first consider the lemmatized version of the motivating example by Chambers and Jurafsky (2008): [*plead, admit, convict*], breaking it into 1-, 2-, and 3-element seeds. They reported the top 6 elements that maximize average pairwise PMI as: *sentence*, *parole*, *fire*, *indict*, *fine*, *deny*. We see similar results in Table 1, while noting again the distinction in underlying statistics: we did not restrict ourselves to cooccurrence based on shared coreferring arguments.

These results show intuitive discourse-level relationships with a sketch size as small as $k = 100$ for the unary seed. In addition, when examining the true PMI rank of each of these terms (reflected as subscripts), we see that highly ranked items in the approximate lists come from the set of items highly ranked in the non-approximate version.[4] A major benefit of the approach is that it allows for approximate scoring of larger sets of elements *jointly*, without the traditionally assumed storage penalty.[5]

**Accuracy 1** We measured the trade-off between PMI approximation accuracy and sketch size. Triples of verb tokens were sampled at random from the *narrative cloze* test set of Chambers and Jurafsky (2008). Seed terms were limited to verbs with df between 1,000 and 100,000 to extract lists of the top-25 candidate verbs by joint, approximate PMI. For a given rank $r$, we measured the overlap of the *true* top-3 PMI and the approximate list, rank $r$ or higher (see Figure 1(a)). If query size is 2, $k = 10,000$, the true top-3 true PMI items tend to rank well in the approximate PMI list. We observe that these randomly assembled queries tax the sketch-based approximation, motivating the next experiment on non-uniformly sampled queries.

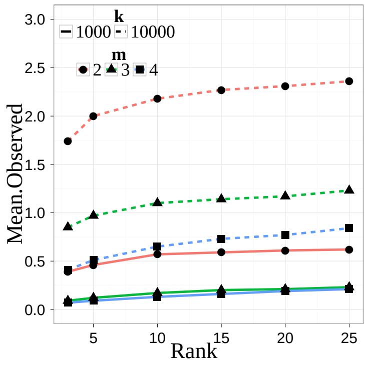
**Accuracy 2** In a more realistic scenario, we might have more discretion in selecting terms of interest. Here we chose the first word of each seed uniformly at random from each document, and selected subsequent seed words to maximize the true PMI with the established words in the seed. We constrained the seed terms to have df between 1,000 and 100,000. Then, for each seed of length 1, 2, and 3 words, we found the 25-best list of terms using approximate PMI, considering only terms that occur in more than 50 documents. Figure 1(b) shows the results of this PMI approximation tradeoff. With a sketch size of 10,000, a rank of 5 is enough to contain two out of the top three items, and the number gradually continues to grow as rank size increases.

**Memory Analysis** Accuracy in a CRS is a function of the aggressiveness in space savings: as $k$ approaches the true length of the posting list for $w_i$, the resulting approximations are closer to truth, at the cost of increased storage. When $k = \infty$, CRS is the same as using an inverted index: Fig. 2 shows the percent memory required for our data, compared to a standard index, as the sketch size increases. For our data, a full index involves storing 95 million document numbers. For the $k = 10,000$ results, we see that 23% of a full index was needed.

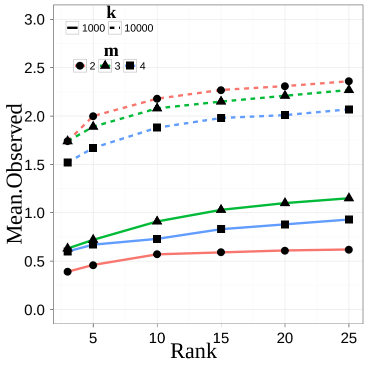
Figure 1(c) shows the quality of approximate best PMI lists as memory usage is varied. A 2-word query needs about 20% of the memory for 2.5 of the

[4]The word "sentence" is consistently higher ranked in the approximate PMI list than it is in the true PMI list: results stem from a given *shared* permutation across the queries, and thus approximation errors are more likely to be correlated.

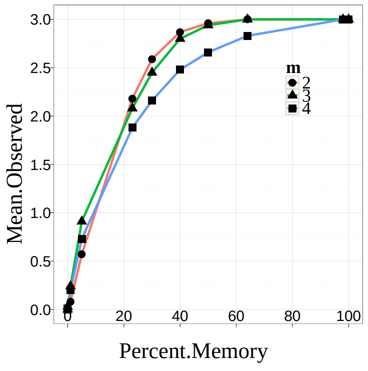
[5]For example, we report that PMI(*plead, admit, convict*) $>$ PMI(*plead, admit, owe*), when $k = 1,000$, as compared to: *avg*(PMI(*plead, convict*), PMI(*admit, convict*)) $>$ *avg*(PMI(*plead, owe*), PMI(*admit, owe*)).



(a)



(b)



(c)

Figure 1: (a) Average number of true top-3 PMI items when seed terms have 1,000 $\leq$ df $\leq$ 100,000 and are chosen uniformly at random from documents. (b) Average number of true top-3 PMI items when seeds are moderate-frequency high-PMI tuples. (c) Average number of true top-3 PMI items in the top ten approximate PMI list, as a function of memory usage, when seeds are moderate-frequency high-PMI tuples.

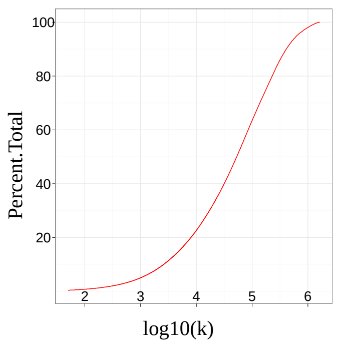


Figure 2: $\%$ of inverted index stored, as function of $k$.

top 3 true PMI items to appear in the top 10. Over 40$\%$ memory is needed for a 4-word query. 2.5 of the top 3 true PMI items appear in the top 50 when the memory is about 35$\%$. This suggests that CRS allows us to use a fraction of the memory of storing a full inverted index, but that memory requirements grow with query size.

**Discussion** Storing exact PMIs of three or four words would be expensive to store in memory for any moderately sized vocabulary, because it would involve storing on the order of $V^m$ count statistics. If we are approximating this with a CRS, we store sketches of length $k$ or less for every word in the vocabulary, which is $O(kV)$. Table 1 and Fig. 1(b) show that the two-word queries start to get good performance when $k$ is near 10,000. This requires 22.7$\%$ of the memory of a complete inverted index, or 21.5 million postings. The three and four word queries get good performance near $k = 100,000$. With this sketch size, 60.5 million postings are stored.

## 4 Conclusion

We have proposed using Conditional Random Sampling for approximating PMI in the discourse understanding community. We have shown that the approximate PMI rank list produces results that are intuitive and consistent with the exact PMI even with significant memory savings. This enables us to approximate PMI for tuples longer than pairs without undue independence assumptions. One future avenue is to explore the use of this structure in applications such as machine translation, as potentially enabling greater use of long distance dependencies than in prior work, such as by Hasan et al. (2008).

## 5 Acknowledgements

We acknowledge support from the National Science Foundation PIRE Grant No. OISE-0530118. We acknowledge the Army Research Laboratory for its support to the first author under SCEP (the Student Career Experience Program). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the supporting agencies.

## References

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL*.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of ACL*.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29.

Kenneth W. Church. 2000. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than $p^2$. In *Proceedings of COLING*.

Andrew Gordon, Cosmin Bejan, and Kenji Sagae. 2011. Commonsense causal reasoning using millions of personal stories. In *Proceedings of AAAI*.

Amit Goyal and Hal Daumé. 2011. Lossy conservative update (lcu) sketch: Succinct approximate count storage. In *AAAI*.

Amit Goyal, Jagadeesh Jagarlamundi, Hal Daumé, and Suresh Venkatasubramanian. 2010. Sketch techniques for scaling distributional similarity to the web. In *6th WAC Workshop at NAACL-HLT*.

Sasa Hasan, Juri Ganitkevitch, Hermann Ney, and J. Andrés-Ferrer. 2008. Triplet lexicon models for statistical machine translation. In *Proceedings of EMNLP*.

Abby Levenberg and Miles Osborne. 2009. Stream-based randomised language models for smt. In *Proceedings of EMNLP*.

Ping Li and Kenneth Church. 2007. A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics*, 33(2):305–354.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of ACL*.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference*.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Proceedings of NAACL*.

Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. In *Proceedings of ACL*.

Ronald Rosenfeld. 1994. *Adaptive statistical language modeling: A maximum entropy approach*. Ph.D. thesis, Carnegie Mellon University.

Roger C. Schank. 1975. Using knowledge to understand. In *Theoretical Issues in Natural Language Processing*.

David Talbot and Thorsten Brants. 2008. Randomized language models via perfect hash functions. In *Proceeedings of ACL*.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of ACL*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2004. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL*.

Benjamin Van Durme and Ashwin Lall. 2009a. Probabilistic Counting with Randomized Storage. In *Proceedings of IJCAI*.

Benjamin Van Durme and Ashwin Lall. 2009b. Streaming pointwise mutual information. In *NIPS*.

Benjamin Van Durme and Ashwin Lall. 2010. Online Generation of Locality Sensitive Hash Signatures. In *Proceedings of ACL*.