Fast Computation of Lexical Affinity Models

Egidio Terra **Charles L.A. Clarke**
School of Computer Science
University of Waterloo
Canada
{elterra,claclark}@plg2.uwaterloo.ca

Abstract

We present a framework for the fast computation of lexical affinity models. The framework is composed of a novel algorithm to efficiently compute the co-occurrence distribution between pairs of terms, an independence model, and a parametric affinity model. In comparison with previous models, which either use arbitrary windows to compute similarity between words or use lexical affinity to create sequential models, in this paper we focus on models intended to capture the co-occurrence patterns of any pair of words or phrases at any distance in the corpus. The framework is flexible, allowing fast adaptation to applications and it is scalable. We apply it in combination with a terabyte corpus to answer natural language tests, achieving encouraging results.

1 Introduction

Modeling term co-occurrence is important for many natural language applications, such as topic segmentation (Ferret, 2002), query expansion (Vechtomova et al., 2003), machine translation (Tanaka, 2002), language modeling (Dagan et al., 1999; Yuret, 1998), and term weighting (Hisamitsu and Niwa, 2002). For these applications, we are interested in terms that co-occur in close proximity more often than expected by chance, for example, {"NEW","YORK"}, {"ACCURATE","EXACT"} and {"GASOLINE","CRUDE"}. These pairs of terms represent distinct lexical-semantic phenomena, and as consequence the terms have an *affinity* for each other. Examples of such affinities include synonyms (Terra and Clarke, 2003), verb similarities (Resnik and Diab, 2000) and word associations (Rapp, 2002).

Ideally, a language model would capture the patterns of co-occurrences representing the affinity between terms. Unfortunately, statistical models used to capture language characteristics often do not take contextual information into account. Many models incorporating contextual information use only a select group of content words and the end product is a model for sequences of adjacent words (Rosenfeld, 1996; Beeferman et al., 1997; Niesler and Woodland, 1997).

Practical problems exist when modeling text statistically, since we require a reasonably sized corpus in order to overcome sparseness problems, but at the same time we face the difficulty of scaling our algorithms to larger corpora (Rosenfeld, 2000). Attempts to scale language models to large corpora, in particular to the Web, have often used general-purpose search engines to generate term statistics (Berger and Miller, 1998; Zhu and Rosenfeld, 2001). However, many researchers are recognizing the limitations of relying on the statistics provided by commercial search engines (Zhu and Rosenfeld, 2001; Keller and Lapata, 2003). ACL 2004 features a workshop devoted to the problem of scaling human language technologies to terabyte-scale corpora.

Another approach to capturing lexical affinity is through the use of similarity measures (Lee, 2001; Terra and Clarke, 2003). Turney (2001) used statistics supplied by the Altavista search engine to compute word similarity measures, solving a set of synonym questions taken from a series of practice exams for TOEFL (Test of English as a Foreign Language). While demonstrating the value of Web data for this application, that work was limited by the types of queries that the search engine supported.

Terra and Clarke (2003) extended Turney's work, computing different similarity measures over a local collection of Web data using a custom search system. By gaining better control over search semantics, they were able to vary the techniques used to estimate term co-occurrence frequencies and achieved improved performance on the same question set in a smaller corpus. The choice of the term co-occurrence frequency estimates had a bigger impact on the results than the actual choice of similarity measure. For example, in the case of the pointwise mutual information measure (PMI), values for $p(b|c)$ are best estimated by counting the number of times the terms b and c appear together within 10-30 words. This experience suggests that the empirical *distribution* of distances between adjacent terms may represent a valuable tool for assessing term affinity. In this paper, we present an novel algorithm for computing these distributions over large corpora and compare them with the expected distribution under an independence assumption.

In section 2, we present an independence model and a parametric affinity model, used to capture term co-occurrence with support for distance information. In section 3 we describe our algorithm for computing lexical affinity over large corpora. Using this algorithm, affinity may be computed between terms consisting of individual words or phrases. Experiments and examples in the paper were generated by applying this algorithm to a terabyte of Web data. We discuss practical applications of our framework in section 4, which also provides validation of the approach.

2 Models for Word Co-occurrence

There are two types of models for the co-occurrence of word pairs: *functional models* and *distance models*. Distance models use only positional information to measure co-occurrence frequency (Beeferman et al., 1997; Yuret, 1998; Rosenfeld, 1996). A special case of the distance model is the *n-gram model*, where the only distance allowed between pairs of words in the model is one. Any pair of word represents a parameter in distance models. Therefore, these models have to deal with combinatorial explosion problems, especially when longer sequences are considered. Functional models use the underlying syntactic function of words to measure co-occurrence frequency (Weeds and Weir, 2003; Niesler and Woodland, 1997; Grefenstette, 1993). The need for parsing affects the scalability of these models.

Note that both distance and functional models rely only on pairs of terms comprised of a single word. Consider the pair of terms "NEW YORK" and "TERRORISM", or any pair where one of the two items is itself a collocation. To best of our knowledge, no model tries to estimate composite terms of form $P(a,b|c)$ or $P(a,b|c,d)$ where *a,b,c,d* are words in the vocabulary, without regard to the distribution function of P.

In this work, we use models based on distance information. The first is an independence model that is used as baseline to determine the strength of the affinity between a pair of terms. The second is intended to fit the empirical term distribution, reflecting the actual affinity between the terms.

Notation. Let G be a random variable with range comprising of all the words in the vocabulary. Also, let us assume that G has multinomial probability distribution function P_g. For any pair of terms b and d, let $\Delta_{b,d}$ be a random variable with the distance distribution for the co-occurrence of terms b and d. Let the probability distribution function of the random variable $\Delta_{b,d}$ be $P_\Delta(b,d)$ and the corresponding cumulative be $C_\Delta(b,d)$.

2.1 Independence Model

Let b and d be two terms, with occurrence probabilities $P_g(b)$ and $P_g(d)$. The chances, under independence, of the pair b and d co-occurring within a specific distance δ, $P_\Delta(b,d|\delta)$ is given by a geometric distribution with parameter p, $\Delta \sim Geo(\delta;p)$. This is straightforward since if b and d are independent then $P_g(b|d) = P_g(b)$ and similarly $P_g(d|b) = P_g(d)$. If we fix a position for a b, then if independent, the next d will occur with probability $P_g(d)\cdot(1-P_g(d))^{\delta-1}$ at distance δ of b. The expected distance is the mean of the geometric distribution with parameter p.

The estimation of p is obtained using the Maximum Likelihood Estimator for the geometric distribution. Let f_δ be the number of co-occurrences with distance δ, and n be the sample size:

$$p = \frac{1}{\mu} = \frac{1}{\frac{1}{n}\sum\limits_{\delta=1}^{\infty} f_\delta} \qquad (1)$$

We make the assumption that multiple occurrences of b do not increase the chances of seeing d and vice-versa. This assumption implies a different estimation procedure, since we explicitly discard what Befeerman et al. and Niesler call *self-triggers* (Beeferman et al., 1997; Niesler and Woodland, 1997). We consider only those pairs in which the terms are adjacent, with no intervening occurrences of b or d, although other terms may appear between them

Figure 1 shows that the geometric distribution fits well the observed distance of independent words DEMOCRACY and WATERMELON. When a dependency exists, the geometric model does not fit the data well, as can be seen in Figure 2. Since the geometric and exponential distributions represent related idea in discrete/continuous spaces it is expected that both have similar results, especially when $p \ll 1$.

2.2 Affinity Model

The model of affinity follows a exponential-like distribution, as in the independence model. Other researchers also used exponential models for affin-

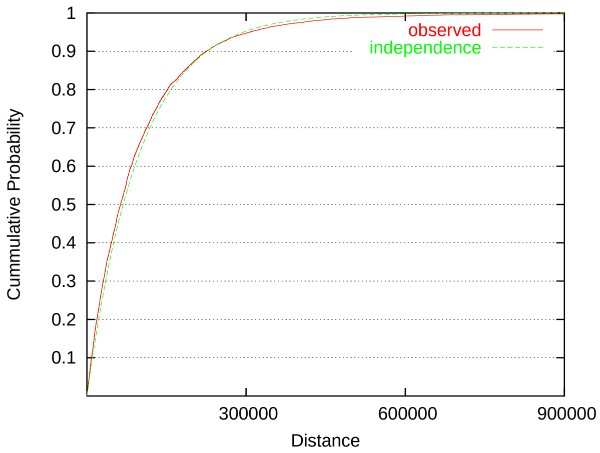

Figure 1: $C_\Delta(\mathit{watermelon}, \mathit{democracy})$

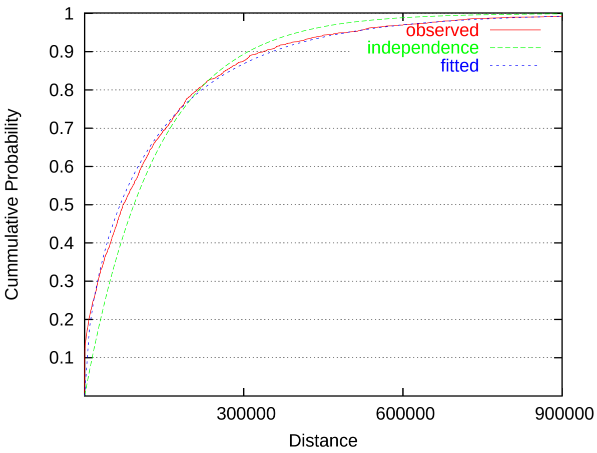

Figure 2: $C_\Delta(\mathit{watermelon}, \mathit{fruits})$

ity (Beeferman et al., 1997; Niesler and Woodland, 1997). We use the gamma distribution, the generalized version of the exponential distribution to fit the observed data. Pairs of terms have a skewed distribution, especially when they have affinity for one another, and the gamma distribution is a good choice to model this phenomenon.

$$Gamma(\Delta = \delta; \alpha, \beta) = \frac{\delta^{\alpha-1} e^{-\delta/\beta}}{\beta^\alpha \Gamma(\alpha)} \quad (2)$$

where $\Gamma(\alpha)$ is the complete gamma function. The exponential distribution is a special case with $\alpha = 1$. Given a set of co-occurrence pairs, estimates for α and β can be calculated using the Maximum Likelihood Estimators given by:

$$\alpha\beta = \frac{1}{n} \sum_{\delta=1}^{\infty} f_\delta \quad (3)$$

and by:

$$\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} - \log \alpha = \frac{1}{n}(\sum_{\delta=1}^{\infty} f_\delta \log \delta) - \log(\frac{1}{n} \sum_{\delta=1}^{\infty} f_\delta) \quad (4)$$

Figure 2 shows the fit of the gamma distribution to the word pair FRUITS and WATERMELON ($\alpha = 0.559947$).

3 Computing the Empirical Distribution

The independence and affinity models depend on a good approximation to μ. We try to reduce the bias of the estimator by using a large corpus. Therefore, we want to scan the whole corpus efficiently in order to make this framework usable.

3.1 Corpus

The corpus used in our experiments comprises a terabyte of Web data crawled from the general web in 2001 (Clarke et al., 2002; Terra and Clarke, 2003). The crawl was conducted using a breadth-first search from a initial seed set of URLs representing the home page of 2392 universities and other educational organizations. Pages with duplicate content were eliminated. Overall, the collection contains 53 billion words and 77 million documents.

3.2 Computing Affinity

Given two terms, b and d, we wish to determine the affinity between them by efficiently examining all the locations in a large corpus where they co-occur. We treat the corpus as a sequence of terms $\mathcal{C} = t_1, t_2, ..., t_N$ where N is the size of the corpus. This sequence is generated by concatenating together all the documents in the collection. Document boundaries are then ignored.

While we are primarily interested in within-document term affinity, ignoring the boundaries simplifies both the algorithm and the model. Document information need not be maintained and manipulated by the algorithm, and document length normalization need not be considered. The order of the documents within the sequence is not of major importance. If the order is random, then our independence assumption holds when a document boundary is crossed and only the within-document affinity can be measured. If the order is determined by other factors, for example if Web pages from a single site are grouped together in the sequence, then affinity can be measured across these groups of pages.

We are specifically interested in identifying all the locations where b and d co-occur. Consider a particular occurrence of b at position k in the sequence ($t_k = b$). Assume that the next occurrence of b in the sequence is t_w and that the next occurrence of d is t_v (ignoring for now the exceptional case where t_k is close to the end of the sequence and is not followed by another b and d). If $w > v$, then no b or d occurs between t_k and t_v, and the interval can be counted for this pair. Otherwise, if $w < v$ let t_u be the last occurrence of b before t_v. No b or d occurs between t_u and t_v, and once again the interval containing the terms can be considered.

Our algorithm efficiently computes all locations in a large term sequence where b and d co-occur with no intervening occurrences of either b or d. Two versions of the algorithm are given, an asymmetric version that treats terms in a specific order, and a symmetric version that allows either term to appear before the other.

The algorithm depends on two *access functions* r and l that return positions in the term sequence $t_1, ..., t_N$. Both take a term t and a position in the term sequence k as arguments and return results as follows:

$$r(t,k) = \begin{cases} v & \text{if } \exists\, t_v = t \text{ s.t. } k \leq v \\ & \quad \text{and } \nexists\, t_{v'} = t \text{ s.t. } k \leq v' < v \\ N+1 & \text{otherwise} \end{cases}$$

and

$$l(t,k) = \begin{cases} u & \text{if } \exists\, t_u = t \text{ s.t. } k \geq u \\ & \quad \text{and } \nexists\, t_{u'} = t \text{ s.t. } k \geq u' > u \\ 0 & \text{otherwise} \end{cases}$$

Informally, the access function $r(t,k)$ returns the position of the first occurrence of the term t located at or after position k in the term sequence. If there is no occurrence of t at or after position k, then $r(t,k)$ returns $N+1$. Similarly, the access function $l(t,k)$ returns the position of the last occurrence of the term t located at or before position k in the term sequence. If there is no occurrence of t at or before position k, then $l(t,k)$ returns 0.

These access functions may be efficiently implemented using variants of the standard inverted list data structure. A very simple approach, suitable for a small corpus, stores all index information in memory. For a term t, a binary search over a sorted list of the positions where t occurs computes the result of a call to $r(t,k)$ or $l(t,k)$ in $O(\log f_t) \leq O(\log N)$ time. Our own implementation uses a two-level index, split between memory and disk, and implements different strategies depending on the relative frequency of a term in the corpus, minimizing disk traffic and skipping portions of the index where no co-occurrence will be found. A cache and other data structures maintain information from call to call.

The asymmetric version of the algorithm is given below. Each iteration of the while loop makes three calls to access functions to generate a co-occurrence pair (u,v), representing the interval in the corpus from t_u to t_v where b and d are the start and end of the interval. The first call ($w \leftarrow r(b,k)$) finds the first occurrence of b after k, and the second ($v \leftarrow r(d,w+1)$) finds the first occurrence of d after that, skipping any occurrences of d between k and w. The third call ($u \leftarrow l(b,v-1)$) essentially indexes "backwards" in the corpus to locate last occurrence of b before v, skipping occurrences of b between w and u. Since each iteration generates a co-occurrence pair, the time complexity of the algorithm depends on M, the number of such pairs, rather than than number of times b and d appear individually in the corpus. Including the time required by calls to access functions, the algorithm generates all co-occurrence pairs in $O(M \log N)$ time.

```
$k \leftarrow 1$;
**while** $k \leq N$ **do**
  $w \leftarrow r(b,k)$;
  $v \leftarrow r(d,w+1)$;
  $u \leftarrow l(b,v-1)$;
  **if** $v \leq N$ **then**
    *Generate* $(u,v)$;
  **end if**;
  $k \leftarrow u+1$;
**end while**;
```

The symmetric version of the algorithm is given next. It generates all locations in the term sequence where b and d co-occur with no intervening occurrences of either b or d, regardless of order. Its operation is similar to that of the asymmetric version.

```
$k \leftarrow 1$;
**while** $k \leq N$ **do**
  $v \leftarrow max(r(b,k), r(d,k))$;
  $u \leftarrow min(l(b,v), l(d,v))$;
  **if** $v \leq N$ **then**
    *Generate* $(u,v)$;
  **end if**;
  $k \leftarrow u+1$;
**end while**;
```

To demonstrate the performance of the algorithm, we apply it to the 99 word pairs described in Section 4.2 on the corpus described in Section 3.1, distributed over a 17-node cluster-of-workstations. The terms in the corpus were indexed without stemming. Table 1 presents the time required to scan all co-occurrences of given pairs of terms. We report the time for all hosts to return their results.

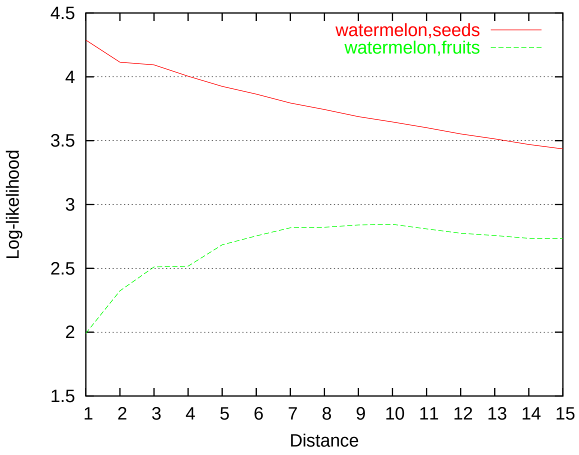

Figure 3: Log-likelihood – WATERMELON

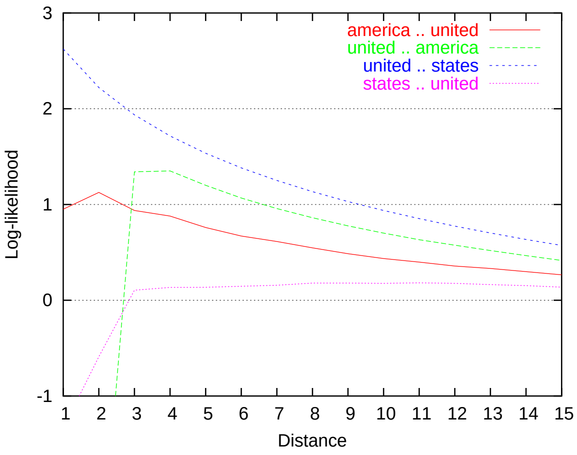

Figure 4: Log-likelihood – UNITED

	Time
Fastest	1ms
Average	310.32 ms
Slowest	744.1ms

Table 1: Scanning performance on 99 word pairs of the Minnesota Word Association Norms

4 Evaluation

We use the empirical and the parametric affinity distributions in two applications. In both, the independence model is used as a baseline.

4.1 Log-Likelihood Ratio

The co-occurrence distributions assign probabilities for each pair at every distance. We can compare point estimations from distributions and how unlikely they are by means of log-likelihood ratio test:

$$log\lambda = log\frac{L(P_{\triangle}(b,d);p_O)}{L(P_{\triangle}(b,d);p_I)} \tag{5}$$

where p_O and p_I are the parameters for $P_{\triangle}(b,d)$ under the empirical distribution and independence models, respectively. It is also possible to use the cumulative $C_{\triangle}$ instead of $P_{\triangle}$. Figure 3 show log-likelihood ratios using the asymmetric empirical distribution and Figure 4 depicts log-likelihood ratio using the symmetric distribution.

A set of fill-in-the-blanks questions taken from GRE general tests were answered using the log-likelihood ratio. For each question a sentence with one or two blanks along with a set of options $\mathcal{A}$ was given, as shown in Figure 5.

The correct alternative maximizes the likelihood of the complete sentence S:

$$log\lambda = log\frac{\prod_{b\in S}\prod_{d\in S,b\neq d} L(P_{\triangle}(b,d|\delta_{b,d});p_O)}{\prod_{b\in S}\prod_{d\in S,b\neq d} L(P_{\triangle}(b,d|\delta_{b,d});p_I)} \tag{6}$$

where $\delta_{b,d}$ is distance of b and d in the sentence. Since only the blanks change from one alternative to another, the remaining pairs are treated as constants and can be ignored for the purpose of ranking:

$$log\lambda_b = log\frac{\prod_{d\in S,b\neq d} L(P_{\triangle}(b,d|\delta_{b,d});p_O)}{\prod_{d\in S,b\neq d} L(P_{\triangle}(b,d|\delta_{b,d});p_I)} \tag{7}$$

for every $b \in \mathcal{A}$.

It is not necessary to compute the likelihood for all pairs in the whole sentence, instead a cut-off for the maximum distance can be specified. If the cut-off is two, then the resulting behavior will be similar to a word bigram language model (with different estimates). An increase in the cut-off has two immediate implications. First, it will incorporate the surroundings of the word as context. Second, it causes an undirect effect of smoothing, since we use cumulative probabilities to compute the likelihood. As with any distance model, this approach has the drawback of allowing constructions that are not syntactically valid.

The tests used are from GRE practice tests extracted from the websites: `gre.org` (9 questions), `PrincetonReview.com`(11 questions), `Syvum.com` (15 questions) and `Microedu.com` (28 questions). Table 2 shows the results for a cutoff of seven words. Every questions has five options, and thus selecting the answer at random gives an expected score of 20%. Our framework answers 55% of the questions.

The _____ *science of seismology has grown just enough so that the first overly bold theories have been* _____.

a) *magnetic. . . accepted*
b) *predictive . . . protected*
c) *fledgling. . . refuted*
d) *exploratory . . . recalled*
e) *tentative. . . analyzed*

Figure 5: Example of fill-in-the-blanks question

Source	Correct Answers
ETS.org	67%
Princeton Review	54%
Syvum.com	67%
Microedu.com	46%
Overall	55%

Table 2: Fill-in-the-blanks results

4.2 Skew

Our second evaluation uses the parametric affinity model. We use the skew of the fitted model to evaluate the degree of affinity of two terms. We validated our hypothesis that a greater positive skew corresponds to more affinity. A list of pairs from word association norms and a list of randomly picked pairs are used. Word association is a common test in psychology (Nelson et al., 2000), and it consists of a person providing an answer to a stimulus word by giving an associated one in response. The set of words used in the test are called "norms". Many word association norms are available in psychology literature, we chose the Minnesota word association norms for our experiments (Jenkings, 1970). It is composed of 100 stimulus words and the most frequent answer given by 1000 individuals who took the test. We also use 100 word pairs generated by randomly choosing words from a small dictionary. The skew in the gamma distribution is $\gamma = 2/\sqrt{\alpha}$ and table 3 shows the normalized skew for the association and the random pair sets. Note that the set of 100 random pairs include some non-independent ones.

The value of the skew was then tested on a set of TOEFL synonym questions. Each question in this synonym test set is composed of one target word and a set of four alternatives. This TOEFL synonym test set has been used by several other researchers. It was first used in the context of Latent Semantic Analisys(LSA) (Landauer and Dumais, 1997), where 64.4% of the questions were answered correctly. Turney (Turney, 2001) and Terra et al. (Terra and Clarke, 2003) used different similarity measures and statistical estimates to answer the questions, achieving 73.75% and 81.25% correct answers respectively. Jarmasz (Jarmasz and Szpakowicz, 2003) used a thesaurus to compute the distance between the alternatives and the target word, answering 78.75% correctly. Turney (Turney et al., 2003) trained a system to answer the questions with an approach based on combined components, including a module for LSA, PMI, thesaurus and some heuristics based on the patterns of synonyms. This combined approach answered 97.50% of the questions correctly after being trained over 351 examples. With the exception of (Turney et al., 2003), all previous approaches were not exclusively designed for the task of answering TOEFL synonym questions.

Pair Sets	γ
Minnesota association norm	3.1425
Random set	2.1630

Table 3: Skewness, $\gamma = 2.0$ indicates independence

In order to estimate α and β we compute the empirical distribution. This distribution provides us with the right hand side of the equation 4 and we can solve for α numerically. The calculation of β is then straightforward. Using only skew, we were able to answer 78.75% of the TOEFL questions correctly. Since skew represents the degree of asymmetry of the affinity model, this result suggests that skew and synonymy are strongly related.

We also used log-likelihood to solve the TOEFL synonym questions. For each target-alternative pair, we calculated the log-likelihood for every distance in the range four to 750. The initial cut-off discarded the affinity caused by phrases containing both target and alternative words. The upper cut-off of 750 represents the average document size in the collection. The cumulative log-likelihood was then used as the score for each alternative, and we considered the best alternative the one with higher accumulated log-likelihood. With this approach, we are able to answer 86.25% of questions correctly, which is a substantial improvement over similar methods, which do not require training data.

5 Conclusion

We presented a framework for the fast and effective computation of lexical affinity models. Instead of using arbitrary windows to compute word similarity measures, we model lexical affinity using the complete observed distance distribution along with independence and parametric models for this distribution. Our results shows that, with minimal effort to adapt the models, we achieve good results by applying this framework to simple natural language tasks, such as TOEFL synonym questions and GRE fill-in-the-blanks tests. This framework allows the use of terabyte-scale corpora by providing a fast algorithm to extract pairs of co-occurrence for the models, thus enabling the use of more precise estimators.

Acknowledgments

This work was made possible also in part by PUC/RS and Ministry of Education of Brazil through CAPES agency.

References

D. Beeferman, A. Berger, and J. Lafferty. 1997. A model of lexical attraction and repulsion. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL*, pages 373–380.

A. Berger and R. Miller. 1998. Just-in-time language modelling. In *Proceedings of IEEE ICASSP*, volume 2, pages 705–708, Seatle, Washington.

C.L.A. Clarke, G.V. Cormack, M. Laszlo, T.R. Lynam, and E.L. Terra. 2002. The impact of corpus size on question answering performance. In *Proceedings of 2002 SIGIR conference*, Tampere, Finland.

I. Dagan, L. Lee, and F. C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69.

O. Ferret. 2002. Using collocations for topic segmentation and link detection. In *Proceedings of the 19th COLING*.

G. Grefenstette. 1993. Automatic theasurus generation from raw text using knowledge-poor techniques. In *Making sense of Words. 9th Annual Conference of the UW Centre for the New OED and text Research*.

T. Hisamitsu and Y. Niwa. 2002. A measure of term representativeness based on the number of co-occurring salient words. In *Proceedings of the 19th COLING*.

M. Jarmasz and S. Szpakowicz. 2003. Roget's thesaurus and semantic similarity. In *Proceedings of RANLP-03*, Borovets, Bulgaria.

J.J. Jenkings. 1970. The 1952 minnesota word association norms. In G. Keppel L. Postman, editor, *Norms of word association*, pages 1–38. Academic Press, New York.

F. Keller and M. Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

T. K. Landauer and S. T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

L. Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics 2001*, pages 65–72.

D. Nelson, C. McEvoy, and S. Dennis. 2000. What is and what does free association measure? *Memory & Cognition*, 28(6):887–899.

T. Niesler and P. Woodland. 1997. Modelling word-pair relations in a category-based language model. In *Proc. ICASSP '97*, pages 795–798, Munich, Germany.

R. Rapp. 2002. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th COLING*.

P. Resnik and M. Diab. 2000. Measuring verb similarity. In *22nd Annual Meeting of the Cognitive Science Society (COGSCI2000)*, Philadelphia, August.

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. computer speech and language. *Computer Speech and Language*, 10:187–228.

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, volume 88.

T. Tanaka. 2002. Measuring the similarity between compound nouns in different languages using non-parallel corpora. In *Proceedings of the 19th COLING*.

E. Terra and C. L. A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of HLT–NAACL 2003*, pages 244–251, Edmonton, Alberta.

P.D. Turney, Littman M.L., J. Bigham, and V. Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of RANLP-03*, Borovets, Bulgaria.

P. D. Turney. 2001. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. In *Proceedings of ECML-2001*, pages 491–502.

O. Vechtomova, S. Robertson, and S. Jones. 2003. Query expansion with long-span collocates. *Information Retrieval*, 6(2):251–273.

J. Weeds and D. Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

D. Yuret. 1998. *Discovery of linguistic relations using lexical attraction*. Ph.D. thesis, Department of Computer Science and Electrical Engineering, MIT, May.

X. Zhu and R. Rosenfeld. 2001. Improving trigram language modeling with the world wide web. In *Proceedings of IEEE ICASSP*, volume 1, pages 533–536.