Lexical Affinities and Language Applications

by

Egidio Terra

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Understanding interactions among words is fundamental for natural language applications. However, many statistical NLP methods still ignore this important characteristic of language. For example, information retrieval models still assume word independence.

This work focuses on the creation of lexical affinity models and their applications to natural language problems. The thesis develops two approaches for computing lexical affinity. In the first, the co-occurrence frequency is the calculated by point estimation. The second uses parametric models for co-occurrence distances.

For the point estimation approach, we study several alternative methods for computing the degree of affinity by making use of point estimates for co-occurrence frequency. We propose two new point estimators for co-occurrence and evaluate the measures and the estimation procedures with synonym questions. In our evaluation, synonyms are checked directly by their co-occurrence and also by comparing them indirectly, using other lexical units as supporting evidence.

For the parametric approach, we address the creation of lexical affinity models by using two parametric models for distance co-occurrence: an independence model and an affinity model. The independence model is based on the geometric distribution; the affinity model is based on the gamma distribution. Both fit the data by maximizing likelihood. Two measures of affinity are derived from these parametric models and applied to the synonym questions, resulting in the best absolute performance on these questions by a method not trained to the task.

We also explore the use of lexical affinity in information retrieval tasks. A new method to score missing terms by using lexical affinities is proposed. In particular, we adapt two probabilistic scoring functions for information retrieval to allow all query terms to be scored. One is a document retrieval method and the other is a passage retrieval method. Our new method, using replacement terms, shows significant improvement over the original methods.

Acknowledgements

Thanks to my wife, Adriana, for her support, patience and dedication during this time and to our son, Gabriel, for all joy he has brought to our family. Thanks to my father.

Thanks to my supervisor, Charlie Clarke, for his endless support and assistance. Thanks to Gordon Cormack for his comments and discussions, and for being part of my committee. Thanks to Olga Vechtomova for discussions on the affinity measures and their applications in IR and for being part of my committee. Thanks Peter Turney for being in my committee and for his paper on the synonym questions that inspired the early stages of my work. I also thank Peter Turney and the National Resource Council for letting me use their servers for some experiments. Thanks Chrysanne diMarco for being in my committee. Thanks Frank Tompa for the comments and ideas in the second-stage exam.

Thanks to Bradley Lushman for late night discussions on my models and in the thesis in general. To Stefan Buettcher for helping on the thesis. Thanks Rodolfo Esteves, Philip Tilker, Thomas Lynam, Ashif Harji, Roy Krischer, John Johansen, Jason Skomorowski, Robert Warren, David Yeung and Daisy Guo for their friendship and good moments in the lab. Thanks to all Brazilians in Waterloo, their friendship provided me with good moments. I was able to do my work because of the financial support of CAPES, a brazilian agency from Ministry of Education, and by PUC/RS which allowed me to leave my duties as lecturer to pursue my degree.

To Adriana and Gabriel

Contents

1 Introduction **1**

1.1 Contributions 4

1.2 Thesis Organization 6

2 Background and Related Work **8**

2.1 Lexical Affinities 8

2.2 Collocation 11

2.3 Semantic Similarity and Lexicons 14

2.4 Large Corpora 16

2.4.1 Large Corpora in Practice 19

3 Affinity based on Point Estimation **23**

3.1 Co-occurrence Frequency Estimation 23

3.2 Affinity Measures 29

3.2.1 Direct Comparison 30

3.2.2 Comparison using Supporting Evidence 35

3.3 Summary 38

4 Affinity Models **40**

4.1 Empirical Distribution and Models for Lexical Affinities 40

4.2 Efficient Retrieval 46

4.3 Summary . . . 49

5 Human-oriented Language Tests **50**

5.1 Synonym questions . . . 50

5.1.1 Point Estimation . . . 52

5.1.2 Skew . . . 65

5.1.3 Log-likelihood Ratio over Intervals . . . 67

5.1.4 Knowledge-based approach for Semantic Similarity . . . 68

5.2 GRE fill-in-the-blanks . . . 69

5.2.1 Log-Likelihood Ratio . . . 69

5.3 Summary . . . 72

6 Scoring Missing Terms in IR **74**

6.1 Missing Term problem . . . 74

6.2 Related Work . . . 76

6.3 Modified Retrieval Methods . . . 77

6.3.1 Passage Retrieval . . . 77

6.3.2 Document Retrieval . . . 80

6.4 Finding Term Replacements . . . 82

6.5 Empirical Evaluation . . . 84

6.5.1 Methodology . . . 84

6.5.2 Passage Retrieval . . . 86

6.5.3 Document Retrieval . . . 89

6.6 Replacement Method and Query Formulation Strategies . . . 94

6.6.1 Evaluation . . . 96

6.7 Summary . . . 99

7 Conclusions and Future Work **100**

Bibliography **104**

List of Tables

3.1 Simple estimator smoothing effect (Terabyte Corpus) 26

3.2 Weighted-window estimator smoothing effect (Terabyte Corpus) 26

4.1 Scanning performance on 99 word pairs 49

4.2 Examples of scanning performance 49

5.1 % correct answers on the three test sets with Simple estimator 59

5.2 % correct answers on the three test sets with Weighted-window estimator 59

5.3 % correct answers on the three test sets with document estimator 59

5.4 Statistical comparison of the three estimators 60

5.5 PMI vs. DICE 60

5.6 Full document vs. document with window constraints. "+" indicates full doc is statistically better, "−" indicates that full doc is statistically worse. "0" indicates that the difference is not significant. 61

5.7 Association norms examples 65

5.8 Skewness, $\gamma = 2.0$ indicates independence 66

5.9 Skew results on synonym questions 66

5.10 Results of log-likelihood ratio over intervals in the synonym questions 67

5.11 % correct answers using similarity based on WordNet 68

5.12 Fill-in-the-blanks results 72

6.1 Replacements examples 75

6.2 Corpus Individual Frequencies 84
6.3 Corpus Frequencies of Pairs at specific distance intervals 84
6.4 Replacement Method results 87
6.5 Top five passage retrieval in Tellex *et al.* 87
6.6 Replacements in TREC topic 51 92
6.7 Replacements in TREC topic 53 92
6.8 Replacements in TREC topic 62 93
6.9 Replacements in TREC topic 68 93
6.10 Replacements in TREC topic 71 94
6.11 Replacements in TREC topic 94 94
6.12 Effectiveness of the document retrieval in the initial set 96
6.13 Passage Retrieval from top 150 Okapi documents in the AQUAINT Corpus 96
6.14 Wilcoxon p-values for p@20 in documents from the AQUAINT corpus 97
6.15 Passage Retrieval from top 150 Okapi documents in the Terabyte Corpus 97
6.16 Wilcoxon p-values for p@20 in documents from the Terabyte corpus 98

List of Figures

2.1 The Impact of Corpus Size on Passage Retrieval performance 21
2.2 The Impact of Corpus Size on Question Answering Performance 22

3.1 Sliding windows from the simple estimator 25
3.2 Sliding weighted-window estimator ($K = 2$) 25
3.3 Effect of marginals on PMI 39
3.4 Effect of marginals on χ^2 score 39
3.5 Effect of marginals on Dice 39
3.6 Effect of marginals on Jaccard 39
3.7 Effect of marginals on Cosine 39
3.8 Effect of marginals on Z-score 39

4.1 $C_\Delta(watermelon, democracy)$ 42
4.2 $C_\Delta(watermelon, fruits)$ 43
4.3 $C_\Delta(watermelon, watermelon)$ 44

5.1 Examples of synonym questions 51
5.2 Results for TOEFL test set with Simple Estimator 53
5.3 Results for TOEFL test set with Weighted-window Estimator 53
5.4 Results for TOEFL test set with Document Estimator 53
5.5 Results for TS1 test set with Simple Estimator 54
5.6 Results for TS1 test set with Weighted-window Estimator 54

5.7 Results for TS1 test set with Document Estimator 54
5.8 Results for TS1 using context and Simple Estimator 55
5.9 Results for TS1 using context and Document Estimator 55
5.10 Influence from the context on TS1 55
5.11 Results for TS2 test set with Simple Estimator 57
5.12 Results for TS2 test set with Weighted-window Estimator 57
5.13 Results for TS2 test set with Document Estimator 57
5.14 Results for TS2 using context and Simple Estimator 58
5.15 Results for TS2 using context and Document Estimator 58
5.16 Influence from the context on TS2 58
5.17 Impact of corpus size on TOEFL 64
5.18 Impact of corpus size on TS1 64
5.19 Impact of corpus size on TS2 64
5.20 Examples of fill-in-the-blanks questions 69
5.21 Log-likelihood – WATERMELON pairs 70
5.22 Log-likelihood – UNITED pairs 71

6.1 *ad hoc* topic 85
6.2 Question answering topics 85
6.3 QA query terms histogram 86
6.4 Passages correct 86
6.5 Interpolated Precision-Recall for topics 51-100 on SJMN 89
6.6 Difference in average precision per topic 90
6.7 Rank by # missing terms - original 91
6.8 Rank by # missing terms - replacement 91

Chapter 1

Introduction

The increase of computing resources in the last decades, both in terms of data availability and in processing power, has had a great impact on natural language processing (NLP). It has created possibilities for empirical research using statistical methods and posed new challenges for existing methods. From Shannon's idea of language as a stochastic process [93], the area of statistical natural language processing has flourished as a promising approach to solving natural language problems, particularly due to the use of these abundant resources. Many state-of-the-art methods in NLP are statistical in nature, including the successful language translation models; statistical parsing and part-of-speech tagging; speech recognition; and many other tasks, including machine learning applied to language problems.

There are other alternatives to NLP, notably the knowledge-based or rationalist approach, mostly expressed in linguistic terms through the ideas laid out by Chomsky [17]. According to Chomsky, the statistical method is not enough to address syntactic problems and facets since, as he claims, humans have a predisposition to language and possess mental structures suitable for language acquisition and use, a feature that machines do not have. While Chomsky mainly focuses on the syntactic level, the lexical-semantic aspect of the language is addressed in the knowledge-based approach by the use of dictionaries and thesauri. The aim of this thesis is not to discuss the advantages or disadvantages of either approach, as advocating texts exist for both [1, 17, 68].

Rather, the ideas in this work are statistical in nature and rely on the presence of large quantities of linguistic data. Moreover, we do not make statements concerning grammatical correctness of sentences, the main criticism of Chomsky with regard to the statistical approach.

As language is manifested in different modes, such as sounds, text and images, there exists a layer of acquisition, perception and synthesis that goes along with its processing. In this work we do not address these issues. Instead, we approach language by using written text, in digital format. Further, we do not try to generate text, although models presented in this work could possibly be adapted to do so, but use existing text to enrich the interaction between human and computers by improving the processes that run on the machine side.

In natural language texts there are many repetitions of word sequences, particularly when these sequences are limited to few words. This idea is one of the main observations behind the work of Shannon [93]. While studying the theoretical limits of communication over channel, Shannon proposed that the message content could be viewed as a statistical distribution of either letter or word sequences, i.e. a language model, and that by observing the empirical distribution of a message one can estimate the bounds for data transfer based on the redundancy within the message. Thus, statistical models provide approximations to language.

One challenging problem derived from Shannon's idea is the fact that it relies on analyzing the data to estimate a model before using it. While in some cases the message is known ahead of time, and thus available for pre-processing, in many others this is not the case. The fact that the data may not be available for model creation prior to its use can be viewed as a shortcoming, but the problem can be minimized by using training data which may or may not reflect the actual data. However, the larger the training data, the more accurate the model and possibly the more general as well. Since large corpora are now widely available, we can make better estimations to create these statistical models more precisely.

Another challenge resulting from the application of Shannon's statistical approach as a model of natural language is that the promising models, the so called higher-order models, pose combinatorial problems. In a first-order word approximation, the model solely addresses words individually. The statistical distribution is multinomial and the number of possible alternatives is

equal to the size of the vocabulary. In gigabyte-sized collections, there may be more than million unique words. For a second-order word approximation, any of these million unique words can be sequentially paired with any other, including itself. The upper bound for the model moves from a million to a trillion alternatives. As noted by Shannon, the higher the order of the model the closer is the approximation to language; thus it is desirable to pursue these models.

Moving to higher-order approximation models has an immediate side-effect: a huge number of pairs will not be seen in practice, even in large corpora, simply because certain words do not occur close to each other in the vocabulary. Thus, it is necessary to find alternatives to maximize the information provided by the training data and to create mechanisms to handle unseen pairs. A common solution uses smoothing and discounting techniques to assign a non-zero probability to unseen pairs [15].

This thesis takes a different approach for higher-order models, in particular for second-order models. Instead of using only adjacency, we use co-occurrences at farther distances, which also provide more efficient way to use the training data. As Shannon's higher-order models are sequential, syntactical constraints tend not to be violated, as long as the training data used to build the model does not contain ungrammatical sentences. This approach does not allow deeper semantic relationships to be captured, since related words, such as synonyms, are not adjacent in many cases. One can relax the sequential constraints, allowing pairs of words to be modeled in positions other than adjacency. This relaxation will allow these other types of semantic relationships to be included in the model but may violate syntax constraints. It is a compromise between structuralism and semantics.

It has been noted that flexing the model beyond limited sequences of adjacent words, to incorporate distance, can be beneficial to sequential language models [3, 89]. Furthermore, the existence of models with semantic relationships would benefit many natural language applications since in practice these relationships have been used in an *ad hoc* way in many applications, including topic and text segmentation [41, 52], query expansion [103], machine translation [96], language modeling [31, 114], and term weighting [47]. For these applications, researchers are interested in capturing language patterns in general but those that co-occur in close proximity

more often than expected by chance are of special interest, for example, "NEW" and "YORK", "ACCURATE" and "EXACT", and "GASOLINE" and "CRUDE". These pairs of words represent distinct lexical-semantic phenomena, and their components have *mutual expectancy* [42]. We call *lexical affinity* the tendency of any group of *lexical units* (words or phrases) to occur together frequently.

Lexical units with high affinity tend to co-occur frequently. As consequence, for particular pairs the repeated co-occurrence gives form to patterns. These patterns vary depending on the type of affinity. We consider the following lexical affinity types: grammatical constructs, e.g. "DUE TO"; semantic relations, e.g. "NURSE" and "DOCTOR"; compounds, e.g "NEW YORK"; and idioms and metaphors, e.g. "DEAD SERIOUS". All of these different types of affinities share high co-occurrence frequency of their constituents. The patterns among these lexical affinity types are not uniform, for instance, idioms are fixed expressions or templates where few words can be included or replaced, e.g. "KICK THE BUCKET" or "WALK A MILE IN (my/her/someone's/our/etc..) SHOES". The pattern of compounds is simpler, variable-sized sequences of adjacent lexical units, e.g. "UNITED STATES" and "UNITED STATES OF AMERICA". The patterns of grammatical constructs may take different forms, such as those in compounds or interspaced sequences, e.g. "The lawyers LOOKED OVER the papers" and "They LOOKED them OVER carefully". Semantic relation patterns are much more flexible, they can occur in syntactic constructs, e.g. "Our nation-wide team of CAR TYRE specialists", or as an idiom, e.g., "The BREAD and BUTTER Theater Company" or even have both lexical units co-occurring together with a reasonable number of words between them, such as "LEXICAL" and "GRAMMATICAL" in this paragraph.

1.1 Contributions

In this work we address the development of lexical affinity models and their application to NLP. We aim to build models to capture single words and phrases as units in these lexical affinity relationships. Since longer phrases tend to be relatively infrequent, it is desirable to draw statistics from large corpora, and thus efficient algorithms that scale well are needed to estimate co-occurrence

frequencies at variable distances. We show that, by benefiting from the vast amount of text now available, these models perform well in practice. In particular, the main contributions are:

1. New frequency estimates for lexical units' co-occurrence (point estimation)

   Two new estimators for co-occurrence are presented. In the first, close co-occurrences are considered to be of greater importance and the frequencies are adjusted to reflect that hypothesis. In the second one, the frequencies are taken from documents, but to avoid bias for long documents, the co-occurrences are discarded if the lexical units occur far from each other. The normalization procedures for these new estimators are also presented.

2. A new framework for computation of lexical affinity models

   We present a framework for the fast computation of lexical affinity models. It is composed of an algorithm to efficiently compute the co-occurrence distribution between pairs of lexical units, 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, these new models are 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 to terabyte-sized collections.

3. New methods for answering multiple-choice synonym questions and fill-in-the-blank questions

   We apply lexical affinity models to answer natural language tests. In particular, sets of synonym questions are answered using existing lexical affinity models and the two new methods. The first new method uses the skew of the gamma distribution, which is used to fit empirical data. The gamma distribution is well suited for skewed data and degree of skew can be used successfully to determine synonymy. The second method compares through log-likelihood the empirical distance distribution of lexical units against the independence model. The log-likelihood is used to answer both multiple-choice synonym and fill-in-the-blank questions. The statistics used come from a terabyte corpus, and our results are

encouraging.

4. A new method to use lexical affinity in document and passage retrieval

   We propose a new method to address the mismatching vocabulary problem, expanding original query terms only when necessary, by complementing the user query for missing terms while scoring documents. This method allows related semantic aspects, calculated through lexical affinity, to be included in a conservative and selective way, thus reducing the possibility of query drift. Our results using replacements for the missing query terms in modified document and passage retrieval methods show significant effectiveness improvement over the original ones.

1.2 Thesis Organization

The remainder of this work is organized as follows. Chapter 2 presents related work and demonstrates the successful use of large corpora in natural language applications. In Chapter 3, we investigate alternatives to measure co-occurrence frequencies of lexical units in close proximity; we refer to the models based on these frequencies as the point estimation models. As co-occurrence frequency is affected by how common its components are, it is necessary to take individual frequencies into account. We also investigate many different functions to compute affinity in Chapter 3. A new approach to lexical affinity models is proposed in Chapter 4, where we compute the whole distance distribution of co-occurrences to build parametric models. We propose a parametric independence model and a parametric model for lexical affinity. The accuracy of these models is related to the estimation of their parameters; we use a large corpus for the estimations and, for efficient computation, we provide a fast algorithm to compute the distribution. We apply the new affinity models to language tests in a comprehensive evaluation of estimation procedures and measures which we present in Chapter 5. In Chapter 6 we present our new method to score missing terms in information retrieval tasks. We modify a passage retrieval method and a document retrieval method to allow replacement of missing terms using lexical affinities.

Bibliographic Notes

Portions of this thesis have appeared elsewhere. Chapter 3 and parts of Chapter 5 are extended versions of the paper "Frequency Estimates for Word Similarity Measures", which appeared in the proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003). Chapter 4 and parts of Chapter 5 are based on the paper "Fast Computation of Lexical Affinity Models", which will appear in the proceedings of the 20th International Conference on Computational Linguistics (COLING-2004). Chapter 6 is based on the paper "Scoring Missing Terms in Information Retrieval Tasks" which will appear in the proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM-2004). The experiment of the impact of corpus size in Chapter 2 is based on the paper "The Impact of Corpus Size on Question Answering Performance", which appeared in the proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2002).

Chapter 2

Background and Related Work

2.1 Lexical Affinities

Lexical affinities are characterized by co-occurrence patterns that can be measured in various forms. We summarize the existing work into three models according to the type of frequency estimates used: *distance models*, *functional models* and *document models*.

Distance models measure the co-occurrence of lexical units both at adjacent and interspaced positions. A particular distance model used to capture patterns in language is the *n-gram* model. These models correspond to Shannon's n-order approximations to language and allow us capture patterns comprising sequences of adjacent lexical units. This provides a model of lexical affinity in the form of compounds, such as noun phrases, or specific grammatical constructs, such as "THE BUG" (determiner followed by noun) . In *n-gram* models, the strength of affinity is given by the conditional $P(w|H)$ (i.e., the probability of seeing w after a sequence of one or more words H), and it is sensitive to the sequential order of the lexical units. Other non-*n-gram* models also explore sequential information, such as those proposed by Kita *et al.* [57] and Frantzi *et al.* [43], who use a cost criterion to evaluate affinity of lexical unit compounds, such as proper names and noun phrases. Dunning [36] selects word bigram pairs with high affinity using log-likelihood instead of conditional probability. Kiss *et al.* [56] also use the log-likelihood ratio to identify abbreviations

in word sequences.

These sequential models are rather limited and fail to capture a broader set of relationships, since many of those are characterized by interspaced lexical units. Even though distance information has been incorporated into language models [3, 77, 89], allowing interspaced lexical units to be taken into account, the end product is a model for sequences of adjacent words, i.e. *n-gram* models. The same interspaced lexical units used to create the model cannot be directly inferred from it.

One alternative to pure sequential models has been extensively used in practice: the co-occurrence frequency between two words is measured by the number of windows of a given size that contain both words. The co-occurrence frequency and marginals are used to compute the degree of affinity. We refer to this model as the *window model* [20, 12, 100, 98], a special case of distance model. This approach to estimate the co-occurrence frequency is justified on the basis that high affinity causes lexical units to occur close together. However, the choice of the maximum distance is somewhat arbitrary, for instance in the previous paragraph exactly eight words separate LANGUAGE from LEXICAL UNITS, while in the preceding paragraph the same lexical units are separated by fifteen words[1]. Examples of distance models include the work of Church and Hanks [20], who use windows of five words to count co-occurrence and later apply mutual information to measure affinity between pairs of words. Turney [100] uses windows of ten words to count pairs of lexical units to find best synonym alternatives in TOEFL tests.

Another approach to modeling patterns of lexical affinities are the *functional models*. The syntactic information of the lexical units is recovered and the lexical-syntactic information is used to compute co-occurrence frequencies [46, 45, 85, 63, 96, 109]. For this approach the syntactic categories must be specified in advance and the co-occurrence frequency is the number of times the lexical units co-occur in those syntactic functions, e.g. DRINK as *verb* and WATER as *object*. As in the window model, the co-occurrence frequency is used along with the marginals to compute the degree of affinity. Unfortunately, there are many lexical units that are syntactically ill-formed, e.g. "BY AND LARGE" and "OF COURSE" [74]. Also, lexical units composed by phrases are

[1] and by one word in this paragraph

hard to specify for a syntactic category. Another shortcoming of this approach is that syntactic information is recovered by parsing, which affects its scalability, although shallow parsing is used as a compromise between speed and parsing information delivered. However, even shallow parsing can be expensive. In a recent work, Pantel et al. [79] estimated that a terabyte corpus would require a part-of-speech tagger to run for 125 days and a deep syntatic parser to take 388.4 years to complete its task. Some examples of functional methods include Grefenstette [46], who uses a dictionary-based shallow parsing to identify pairs of nouns and adjectives, subject and objects, and nouns modified via preposition. Lee [63] uses a part-of-speech tagger to identify nouns as heads of direct object of verbs, and later applies different affinity measures. Evert [37] proposes a new significance test to analyze association measures applied to adjective-noun pairs and prepositional-phrases as verb attachments.

A third approach, the *document model*, is commonly used in information retrieval. The co-occurrence is measured by the number of contexts in which the words appear together. The context is a linguistic unit such as a sentence, a paragraph or, as usual in information retrieval, a document [64, 82, 110, 98]. It often happens that a document is used to measure co-occurrence, and when that is the case, there are several factors that need to be addressed. For instance, document size and nature play an important role [87] and each document is seen as a context unit in which both lexical items occur. A document with a larger vocabulary will contain more pairs. For the measure of strength, the individual marginals are not estimated directly from the corpus, but in the number of documents that contain each individual lexical item in consideration. Examples are found in the Information Retrieval literature, such as the early work of Lesk [64], where associated words, i.e., words co-occurring in documents, are used for query expansion. Peat and Willett [82] investigate the usefulness of intra-document word co-occurrence and its limitations. Xu and Croft [110] extract terms for pseudo-relevance feedback from passages previously retrieved.

Related terminology, with its origins in linguistics, for lexical affinities is given in [34, 48, 84]. A two-level model is presented, composed of syntagmatic and paradigmatic levels. The syntagmatic level is comprised of relationship types between lexical units selected by their syntactic roles (e.g. "WASH" as verb and "HANDS" as noun). The paradigmatic level includes relationships other than

syntagmatic, such as synonyms and antonyms. The syntagmatic level is similar to the *functional model* defined earlier. The paradigmatic is different from the models defined above, for instance the distance model allows syntactical relationships to be captured (e.g. at adjacent positions). However, the main difference is that the two-level model, syntagmatic and paradigmatic, is used to categorize the relationships, whereas the functional, distance and document models are used to measure co-occurrence, with no regard to the relationship between its components.

Along with frequency estimates, the strength of the lexical affinities is calculated by some function of divergence between marginals and expected joint, and the actual observed joint frequency. There are many such functions or measures of affinity and in general their choice is *ad hoc*. Examples include the use of log-likelihood [36, 96], cosine and dice coefficients [82, 64], L_1-norm [84], Z-score [103], pointwise mutual information [20], and χ^2 [11] among others. To address this issue, comparative evaluations have been proposed in the literature for specific phenomena. Evert and Krenn [39] evaluated log-likelihood, t-test, χ^2 and mutual information for syntagmatic relationships between adjective-noun pairs and preposition-noun-verb triples. Thanopoulos et al. [99] evaluated t-test, χ^2, log-likelihood ratio and pointwise mutual information to compare lexically associated bigrams. Pearce [81] evaluated Z-score, pointwise mutual information, cost criteria, log odds ratio by applying collocations formed from bigrams. Lee [63] evaluates Kullback-Leibler divergence, Jensen-Shannon, skew divergence, Euclidean distance, cosine measure, L_1-norm, confusion probabilities and τ-coefficient as alternatives to distribute probability mass in a back-off language model.

2.2 Collocation

Collocation is an alternative term to describe the lexical phenomena that interest us in this work. Unfortunately, this term is overloaded by many distinct definitions in the literature [42, 18, 5, 57, 74, 68]. These definitions handle the types and characteristics of lexical affinity from a linguistic point of view.

Firth [42] states that the collocation of a word or a "piece" is not to be regarded as "mere

juxtaposition, it is an order of mutual expectancy. The words are mutually expectant and mutually prehended." He considers colligations as a separate phenomena, driven by syntactic function of the lexical units. Thus, Firth follows the syntagmatic and paradigmatic approaches to explain lexical relationships.

Choueka [18] defines collocation as "a sequence of two or more consecutive words, that has characteristics of a syntactic and semantic unit, and whose exact and unambiguous meaning or connotation cannot be derived directly from the meaning or connotation of its components". In Choueka's work there is a concern for syntactic structure since he assumes that collocations are made of consecutive words. In practice, Choueka uses sequences of 2-6 words; the types of lexical affinities captured by these patterns include compounds, some sequential idiomatic expressions and foreign-language phrases.

Benson [4, 5] studies the use of collocations in dictionaries from a lexicographer's point-of-view. His definition of collocation is somewhat vague: "arbitrary and recurrent word combination". As pointed out in his work, there are no clear rules on how collocations are created; it is one of the results of an arbitrary process of repetitive usage of a set of words. Another problem, he points out, when dealing with collocations, is that there are "many instances when the dividing line between collocations and free combinations is not clear".

Kita [57] defines collocation as "a cohesive word cluster, including idioms, frozen expressions and compound words". As with Choueka, Kita is particularly interested in sequential expressions.

Moon [74] thoroughly examines different types of natural language expressions, using the term collocation to describe "simple co-occurrence of items", and anomalous collocations to designate some special types of expressions. As pointed out by Moon, the nature of anomalous collocations is "syntagmatically and paradigmatically aberrant". She further classifies collocations into three kinds, according to the phenomena they describes: 1) semantic fields, e.g. "JAM" and other food-related words; 2) association with a member of certain class or category, e.g. "RANCID" and "BUTTER" and 3) syntactic (colligations), e.g. "TO BE" and "ONE OF".

These definitions have one or more of the following characteristics:

- Non-compositionality

The meaning of the co-occurring lexical units can not be derived by taking the meaning of its constituents in isolation, e.g. "RAINING CATS AND DOGS".

- Non-substitutability

  The expression is fixed and the substitution of one of its constituents would generate a meaningless expression. For instance, it is not possible to replace "UNITED" by "COMBINED" in "UNITED STATES OF AMERICA"

- Institutionalization

  The collocation is consistently used by a group of speakers. The particular meaning of that collocation arises from repetitive use in some context. Idioms and slang are probably the best examples, e.g. "MY BAD".

- Two or more words

  At least two lexical units compose the collocation. Single words may have multiple meanings but they are not considered collocations, instead they are said to be *ambiguous* or *polysemic*.

- The expression forms a syntactic unit on its own

  This is the syntactic counterpart of non-compositionality, i.e. just as the meaning is not the sum of the parts, the syntactic function is not predictable from its components either. For instance,"BY AND LARGE" is a sentence adverbial.

- Lexicogrammatical fixedness or non-modifiability

  The addition of a term or grammatical transformation creates an invalid expression. For example, the expression "RAINING WHITE CATS AND BLACK DOGS" would not be expected by an English speaker.

- Sequentiality

  Sequences of words that have a non-compositional meaning and are institutionalized are normally considered to be collocations. Examples include "UNITED STATES OF AMERICA" and "RANCID BUTTER".

- Non-translatability

  Collocations are language dependent. Similar expressions may exist in other languages but they would probably be composed of different words. This is a side-effect of non-compositionality, since the words do not yield the meaning. As consequence, word translations would be meaningless. E.g. the translation of "KICK THE BUCKET" would not have the same meaning in French if literally translated.

- Collocational degree

  Some collocations are more evident than others. There is a fuzzy limit on where the combinations of words are collocations or not.

Note that not all the definitions agree with the list above. Furthermore, some characteristics or features are not clear or easily observable. For instance, many would agree that "UNITED STATES" is a collocation, however it is literally translatable to French ("ETATS-UNIS") and to Spanish ("ESTADOS UNIDOS"). Another example of collocation that allows translation is "BLACK SHEEP", which can mean outcast and has the same meaning in the German "SCHWARZES SCHAF". Because of the disagreement on the definition of collocation, it becomes hard to use the term without violating some of the definitions above. We prefer the term *lexical affinity*, which allows the relationship of lexical items to be captured in any kind of linguistic relationship. Therefore the term collocation is not used in this work.

2.3 Semantic Similarity and Lexicons

Another related subject is that of semantic similarity, since some lexical units with high affinity will also be semantically related. However, given that lexical affinities can also be used as basis for other models, such as a statistical language model, then the overlap between semantic similarity and lexical affinity is not complete. Nonetheless, some experiments performed in Chapter 5 can be performed using more traditional semantic similarity approaches. One such approach for semantic similarity is through the use of knowledge bases, such as dictionaries, thesaurus and

other lexicons. These knowledge bases are normally created manually, or semi-automatically, and as such are expensive to build and to extend. These knowledge bases tend to have high quality information contained in them, however they also tend to be incomplete. In particular, Benson [5] discusses the lack of common agreement on how and what to include in collocation dictionaries.

WordNet is a popular online lexicon manually created with some semantic relationships, such as *synonymy*, *is-a*, and *part-of* [40]. It is used in many different natural language related applications, including question answering, information retrieval and word sense disambiguation. It is structured as an hierarchy of concepts, where the concepts are connected via relationships. These connections create a network that can be explored in many distinct forms. Some concepts *glosses* that illustrate how the concept is used in the sense described by the entry. A word is listed as many times as the number of senses assigned to it.

Patwardhan et al. [80] summarize some popular approaches used to explore the WordNet structure[2]. For two given words b and d, the distinct semantic similarities are calculated as follows:

- Lesk

  Compute the overlap between glosses of the two words. Since glosses are brief, there is a good chance of a zero overlap. In Patwardhan et al. [80] this measure has been extended to include the glosses of other words occurring close to b or d.

- Leacock-Chodorow

  Compute similarity by path distance between b and d. However, it is only applicable to nouns.

- Resnik

  This measure uses the information content of concepts $-\log P(concept)$. The concept used is the lowest common subsumer of b and d, which explores the hierarchical aspect of wordnet.

[2]implementation has been made available as a module for Perl WordNet::Similarity, available at http://www.d.umn.edu/ tpederse/similarity.html

$R(b, d) = ic(lcs(b, d))$, where ic stands for information content and lcs for lowest common subsummer.

- Jiang-Conrath

  This measure is related to Resnik's. It also uses information content of the lowest common subsumer, but subtracts it from the information content of b and d. $JC(b, d) = \frac{1}{ic(b)+ic(d)-2 \cdot ic(lcs(b,d))}$.

- Lin

  Related to Jiang-Conrath, but using harmonic mean instead: $Lin(b, d) = \frac{2 \cdot ic(lcs(b,d))}{ic(b)+ic(d)}$.

- Hirst-St.Onge

  Also explores path length but allows for change in the direction. It works as a search in the network and uses 2 parameters: $PW = C - path_length - (k \cdot changes_in_direction)$, where C and k are parameters.

Note that WordNet contains adjectives, adverbs, nouns, and verbs. Also, many words are cross-listed in these parts-of-speech, thus the use of WordNet implies also the use of some syntatical information.

2.4 Large Corpora

In order to model language phenomena statistically, a large body of examples is required. Ideally, the whole set of sentences composing the language would be used, but that is clearly impossible given the infinite number of natural language sentences. The statistical method addresses this problem by creating a inference model from a sample. This sample is called the *corpus* in statistical natural language processing. The corpus needs to be *representative* of the aspects of the language under study [9], and *balanced* (i.e. it needs to address all aspects in the same proportion to the language). Other aspects on the use of corpora for linguistics studies can be found in the literature [65, 68, 70, 71, 74].

The inference process is achieved by creating estimators for each parameter in the chosen model. For instance, the word "THE" is normally the most frequent in any English text. It occurs 26,830,535 times in the AQUAINT corpus, composed of newswire articles from 1998-2000 and distributed by LDC[3]. This corpus contains 463,003,511 token occurrences. A common estimator, the maximum likelihood estimator (MLE) would assign a occurrence probability of $26,830,535/463,003,511$. Therefore, the MLE would predicts that the word "THE" is expected to occur in 5.79% of an English text.

Estimators have properties, such as *efficiency*, *unbiasedness* and *consistency*. An estimator, $\hat{\theta}$, is said to be unbiased if the estimation it produces is equal to the real parameter, θ, from the population, i.e. $E(\hat{\theta}) = \theta$. The bias is then the amount of deviation of the estimation: $B(\hat{\theta}) = E(\hat{\theta}) - \theta$. However, the error of the estimator is not only given by its bias. If we take the expected mean square error of the estimator:

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta} - \theta)^2] \\ &= E[(\hat{\theta} - E[\hat{\theta}] + E[\hat{\theta}] - \theta)^2] \\ &= E[(\hat{\theta} - E[\hat{\theta}])^2 + 2 \cdot E[(\hat{\theta} - E[\hat{\theta}]) \cdot (E[\hat{\theta}] - \theta)] + (E[\hat{\theta}] - \theta)^2 \\ &= E[(\hat{\theta} - E[\hat{\theta}])^2] + (E[\hat{\theta}] - \theta)^2 \\ &= Var(\hat{\theta}) + B^2(\hat{\theta}) \end{aligned}$$

we see that the error is also related to the variance of the estimator. Ideally, the variance and bias are small, in which case the estimator is more *efficient*.

The benefit from using a larger corpus is best seen in the property of *consistency*. A consistent estimator $\hat{\theta}$ converges to the real parameter θ as the size of the corpus increases, i.e.,

$$\lim_{N \to \infty} P[|\hat{\theta} - \theta| > \epsilon] = 0, \forall \epsilon > 0 \tag{2.1}$$

[3]http://www.ldc.upenn.edu/

which is also known as the *weak law of large numbers*. This law can be derived from the bounds in probabilities given by Tchebyshev's inequality:

$$P[|x - \mu| \geq c] = \frac{\sigma^2}{c^2} \tag{2.2}$$

which follows from Markov's inequality:

$$P[u(x) \geq c] \leq \frac{E[u(x)]}{c} \tag{2.3}$$

then

$$\begin{aligned} P[|x - \mu| \geq c] &= P[(x - \mu)^2 \geq c^2] \\ &\leq E\left[\frac{(x - \mu)^2}{c^2}\right] \\ &= \frac{\sigma_x^2}{c^2} \end{aligned}$$

The weak law of large numbers can be derived as follows: given a set of random variables $X_1, .., X_n$ independent and identically distributed with the same mean μ and variance σ^2 then we have

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n} = \frac{n \cdot \mu}{n} = \mu$$

and,

$$\begin{aligned} \sigma_{\overline{X}}^2 &= Var\left(\frac{X_1 + \ldots + X_n}{n}\right) \\ &= Var\left(\frac{X_1}{n}\right) + \ldots + Var\left(\frac{X_n}{n}\right) \\ &= \frac{\sigma^2}{n^2} + \ldots + \frac{\sigma^2}{n^2} \\ &= \frac{\sigma^2}{n} \end{aligned}$$

thus, using Tchebyshev's inequality,

$$\begin{aligned} P[|x-\mu| \geq c] &\leq \frac{\sigma^2_{\overline{X}}}{c^2} \\ &\leq \frac{\sigma^2}{n \cdot c^2} \end{aligned}$$

Thus, $\lim_{N\to\infty} P[|\hat{\theta} - \theta| > \epsilon] = 0$ ∎

Natural language problems are normally complex in nature and the underlying mental processes are often not sufficiently understood to be modeled. A common strategy of solving NL problems is to use heuristics. In particular, when the statistical approach to natural language problems is used along with these heuristics, it is common to use consistent estimators, such as the maximum likelihood estimator, as one of the data sources in the problem solution.

2.4.1 Large Corpora in Practice

We illustrate the effect of a large corpus applied to factoid Question Answering (QA), a natural language problem that has received increasing research attention in the recent years [104, 105, 106, 107, 108]. The problem is defined as follows: given a set of questions in natural language, find the answers in a target corpus. No other information is supplied; thus the systems have to handle natural language directly. From a statistical perspective, there is no estimator for this problem, which is too broad and susceptible to idiosyncrasies of the language.

A recent trend in QA and/or natural language problems is to use the World Wide Web as a corpus to address many natural language problems [2, 53, 58, 24, 86, 91]. It is immense, free and instantly available, as Kilgarriff and Grefenstette describe in the Computational Linguistics journal special issue on the Web as a Corpus [54]. We use a terabyte of HTTP, crawled from the general Web in mid-2001, as the corpus in this QA experiment. Starting with a seed set of URLs representing the home pages of 2392 universities and other educational organizations, pages were gathered in breadth-first order with one exception: if a breadth-first ordering would place an excessive load on a single host, defined as more than 0.2% of total crawler activity over

a time period of approximately one hour, URLs associated with that host were removed and requeued until the anticipated load dropped to an acceptable level. Pages at a given depth from the seed set were crawled in random order. During the crawl, duplicate pages were detected and eliminated, and do not form part of the final collection. A breadth-first ordering is known to produce high-quality pages [75], and we expected the crawl to contain the answer to many simple factual questions.

A common approach for QA, as taken, e.g., by Clarke *et al.* [24], Kwok *et al.* [58] and Brill *et al.* [10], is to focus on some part of the corpus with greater chances of being relevant to the question, and find the most frequently occurring string. As pointed out by Brill *et al.* "the larger the data set from which we can draw answers, the greater the chance we can find an answer that holds a simple, easily discovered relationship to the query string."

The questions from a standard evaluation, TREC 2001 QA track [106], were executed over a range of target corpus sizes, representing from 3% to 100% of the available Web collection. During the experiment a small portion of the full terabyte collection was off-line, and the experiment was run over an actual Web collection consisting of 960GB of HTTP. We used MultiText's Question Answering system to answer these questions. For each question, we retrieved 40 passages as raw material for the answer selection component, which tries to find 50 byte answers to it. Each passage is 1000 bytes long [24].

Responses are judged using an automatic evaluation script provided by the National Institute of Standard and Technologies (NIST), the organization that runs TREC. The script executes a series of question-specific regular expressions over the responses returned for each question. Whenever a match occurs, the response is marked as correct. The script contains patterns for only 433 of the original 500 TREC 2001 questions. Most of the remaining questions either did not have an answer in the TREC 2001 target corpus or were discarded by NIST due to typographical errors. The 67 missing questions are not considered. The evaluation is made in two cut points in the QA system: 1) after the passages are extracted; 2) on the final 50-byte snippets. The same script was used to judge the passages and 50-byte snippets; thus the passages have a greater chance of matching the pattern then the snippets.

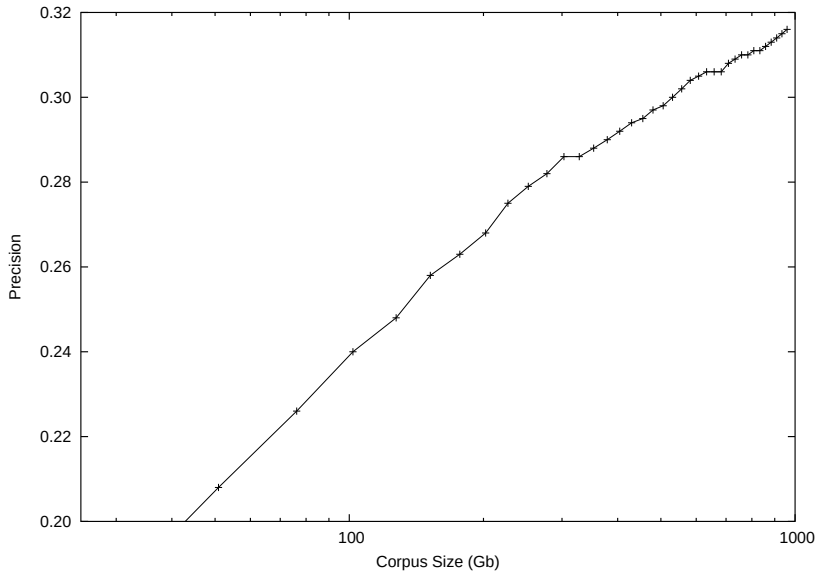


Figure 2.1: The Impact of Corpus Size on Passage Retrieval performance

The results for passages are presented in Figure 2.1. It reports the precision at 40 passages for a variety of fractions of the entire corpus. Experience with traditional IR systems indicates that system performance is directly related to corpus size [28]. The results presented in Figure 2.1 observed a similar relationship.

The results for the whole QA system at different corpus size, measured by mean reciprocal rank (MRR) on the top five 50-byte snippets and number of questions with correct answer in one of the top five snippets, is presented in Figure 2.2. While performance does improve up to 400-500GB, it then appears to reach an asymptote and actually declines slightly after that. An examination of the answers returned by the system suggests a possible weakness in the evaluation methodology. Generally, automatic evaluation by a script is not as accurate as manual evaluation by a human judge. Automatic scripts err both by marking responses as correct when the surrounding context does not support the answer and by missing correct answers that do not match the expected syntactic form. In this case, the script may be "overfitted" to the syntactic forms that appear in the TREC corpus. Certainly many correct answers are missed. Finally, it is possible that the heuristics employed when extracting answers from passages may be the cause of the non-increasing

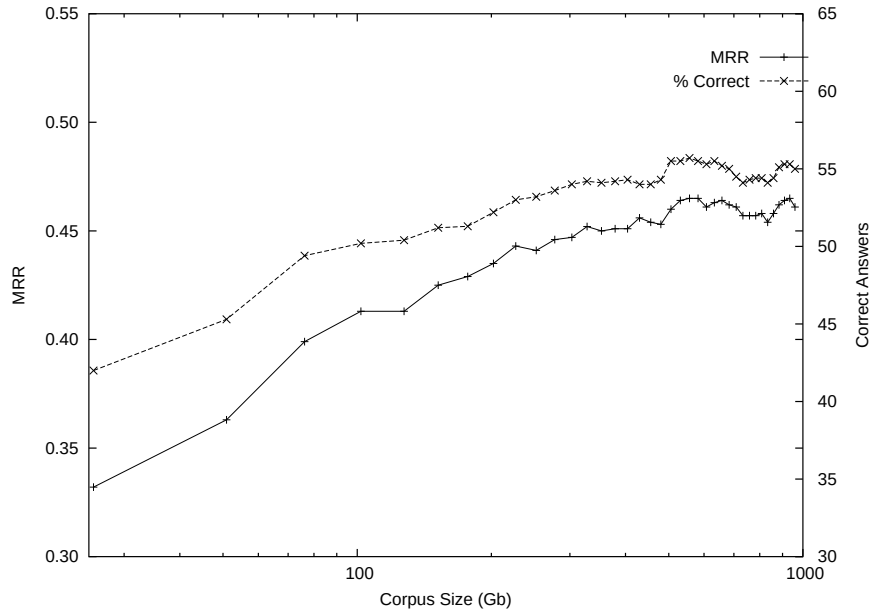


Figure 2.2: The Impact of Corpus Size on Question Answering Performance

behavior of the QA system.

Chapter 3

Affinity based on Point Estimation

In this Chapter we investigate alternatives for computing affinity between lexical units by examining their co-occurrences at specific distances or intervals. If the co-occurrence distance between pairs of lexical units is described by a given random variable, then the methods in this Chapter explore the co-occurrence frequency at particular points in this random variable's domain. We describe an existing method and propose two new methods for co-occurrence frequency point estimation. There are many proposed affinity measures that make use of co-occurrence frequencies to compute the strength of the affinity. We discuss these measures as background for use in Chapters 5 and 6.

3.1 Co-occurrence Frequency Estimation

The simplest way to measure co-occurrence frequency is to estimate the number of occurrences in a fixed structure—a context—such as a sliding window or a document. In some cases, it is not necessary to compute the whole distance distribution, just specific points of it. The *n-gram* model is an example, given that the only frequencies required are those from adjacent lexical units. In other cases, only a maximum interval is needed, since the co-occurrences in the same excerpt are relevant for lexical units with high affinity.

Context can be expressed in the form of a window around the lexical units, where the window size can vary depending on the desired lexical relationship. For instance, Church and Hanks [20] used windows of size 5 as a "compromise; this setting is large enough to show some of the constraints between verbs and arguments, but not so large that it would wash out constraints that make use of strict adjacency." Martin [69] indicates that five words is enough to retrieve 95% of significant collocates in a corpus of 70 million words, where significant collocation means "one in which the two items co-occur more often than couldd be predicted on the basis of their relative frequencies and the length of the text under consideration." Dunning [36] used windows of size two. Smadja et al. [94] also used windows of five words: "most of the lexical relations involving a word w can be retrieved by examining the neighborhood of w, wherever it occurs, within a span of five words." Choueka [18] examines sequential expressions of length two to six.

Using documents or subdocuments as the co-occurrence context for lexical units is a common approach in information retrieval. The context is a linguistic unit such as a sentence, a paragraph or, as usual in information retrieval, a document [33, 64, 82, 110, 98]. Many examples are found in the information retrieval literature, such as the work of Lesk [64], where associated words, i.e., words co-occurring in documents, are used for query expansion. Peat and Willett [82] investigate intra-document word co-occurrence and its limitations. Xu and Croft [110] extract terms for pseudo-relevance feedback from passages previously retrieved.

We investigate two new point estimators: weighted-window and a modified document estimator. As baseline for our experiments, we also describe the method proposed by Church and Hanks [20] and also used by Smadja [94], which we call *simple estimator*.

Simple Estimator

This estimator is the simplest case of the *window* model. All the co-occurrences of the lexical units being investigated are computed at distances $\delta = 1..K$. Let f_δ be the frequency at distance δ between two lexical units b and d; K be the maximum distance (window size-1). The joint frequency is just the sum of these counts:

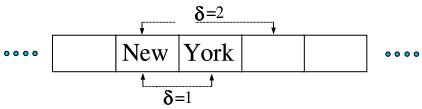


Figure 3.1: Sliding windows from the simple estimator

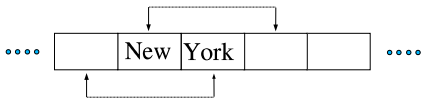


Figure 3.2: Sliding weighted-window estimator ($K = 2$)

$$\hat{f}(b,d) = \sum_{\delta=1..K} f_\delta \tag{3.1}$$

Let N be the size of the corpus in words and N_{joint} the number of possible co-occurrences. The MLE for the joint probability is:

$$\hat{P}(b,d) = \frac{\hat{f}(b,d)}{N_{joint}} \tag{3.2}$$

In order to compute all possible word pairs, every position in the corpus must start a window. The maximum number of co-occurrence counts, N_{joint}, is used to normalize the joint frequency and is equal to

$$N_{joint} = K \cdot N - \frac{K \cdot (K+1)}{2} \tag{3.3}$$

In the simple estimator, every occurrence of a pair of words accounts for the same weight, with no regard to their distance; the window size works as a cut-off.

Church and Hanks [20] use the simple estimator with the window size equal to five words (maximum distance $K = 4$). They give an example of this normalization for the sentence: "LIBRARY WORKERS WERE PROHIBITED FROM SAVING BOOKS FROM THIS HEAP OF RUINS". The frequencies are $\hat{f}(prohibited) = 1$, $\hat{f}(from) = 2$ and $\hat{f}(prohibited, from) = 2$. The number of co-occurrences in the example is inflated and needs normalization, for which Church and Hanks propose "divide the $f(x, y)$ by window size - 1". They proceed to say that "This adjustment has the additional benefit of assuring that $\sum \hat{f}(x, y) = \sum \hat{f}(x) = \sum \hat{f}(y) = N$". The normalization in equation 3.3 is similar to that of Church and Hanks, but we also consider windows of size smaller than specified in the extremes of the database, that is a window starting at the second

Pairs		$\hat{f}(b,d)$ (Window Frequency)			
b	d	$K=1$	$K=31$	$K=63$	$K=127$
NEW	YORK	11,515,513	14,758,180	16,140,332	17,578,673
COFFEE	CHOCOLATE	3,327	47,067	58,947	72,102
SUCCINCTLY	FREELY	0	34	77	190
		$\hat{P}(b,d)$ (Estimated Probability)			
NEW	YORK	2.09E-04	8.36E-06	4.57E-06	2.49E-06
COFFEE	CHOCOLATE	6.03E-08	2.67E-08	1.67E-08	1.02E-08
SUCCINCTLY	FREELY	0.00	1.93E-11	2.18E-11	2.69E-11

Table 3.1: Simple estimator smoothing effect (Terabyte Corpus)

Pairs		$\hat{f}(b,d)$ (Window Frequency)			
b	d	$K=1$	$K=31$	$K=63$	$K=127$
NEW	YORK	11,515,513	306,198,055	564,234,295	1.014E+09
COFFEE	CHOCOLATE	3,327	914,556	2,263,829	5,296,455
SUCCINCTLY	FREELY	0	408	2,332	11,074
		$\hat{P}(b,d)$ (Estimated Probability)			
NEW	YORK	2.09E-04	1.12E-05	5.07E-06	2.26E-06
COFFEE	CHOCOLATE	6.03E-08	3.34E-08	2.04E-08	1.18E-08
SUCCINCTLY	FREELY	0.00	1.49E-11	2.10E-11	2.47E-11

Table 3.2: Weighted-window estimator smoothing effect (Terabyte Corpus)

last position of database (position $N-2$) up to the $K-1$ last position ($N-K-1$).

Weighted-Window Estimator

The simple estimator makes no assumption about the effects of proximity. In some cases, it may be desirable to treat nearby co-occurrences differently from farther ones. An alternative is to weight the co-occurrence based on the distance of the lexical units. As in the simple estimator, a cut-off of maximum distance K is set but, unlike the simple estimator all pairs in the window are counted, from distance 1 to K. The window slides one position at time, so the number of windows in the corpus is $N-K$. This will inflate the counts of closely occurring pairs since closer pairs will be counted many times in different windows. Figure 3.2 shows an example where $K=2$ and the lexical units "New" and "York" are counted twice. If the same terms had a word in between them then they would have be counted in only one window. The MLE for this estimator is, once again,

$$\hat{P}(b,d) = \frac{\hat{f}(b,d)}{N_{joint}} \tag{3.4}$$

where $\hat{f}(b,d)$ is the number of co-occurrence of lexical units pair b and d, and is computed as follows:

$$\hat{f}(b,d) = |W_k| \tag{3.5}$$

where W_k is the set of all windows containing both lexical units b and d. The normalization constant is the sum of co-occurrence lexical units pairs, N_{joint}. This value can be calculated by investigating each window: every window where the maximum distance is K has exactly one pair at that distance, two at $K-1$, and successively down to distance one for which there are K pairs in the window. Therefore, every window will have C_w counts:

$$C_W = \sum_{\delta=1}^{K} \delta = \frac{K \cdot (K+1)}{2}$$

since there are $N-K$ such windows in the corpus, N_{joint} follows,

$$N_{joint} = (N-K) \cdot \left[\frac{K \cdot (K+1)}{2} \right] \tag{3.6}$$

The weight can be explained from a different perspective. We can examine pairs at different distances and check the number of windows in which they are counted. If the maximum distance in the window is K, then the number of windows that will slide over a adjacent pair is K, as shown in the example of Figure 3.2 for the pair "New York". The number of windows will decrease linearly with regard to pair distance, thus adjacent pairs will be counted K times, pairs with distance two will count $(K-1)$ times, and so on down to pairs at distance K, which will be counted only once. Thus weight decay is linear in this case, and the window size defines the slope of the weight decay. In the case of Figure 3.2, where $K=2$, adjacent pairs will have weight two times greater than pairs at distance two.

The weighted-window estimate has a smoothing factor. Since more pairs can be observed at higher distances (and all pairs observable if $K = N$), this estimator has a liner decay smoothing that does not consider individual probabilities as other smoothing and discounting techniques do; instead, the smoothing is based on each lexical unit context. The probability mass from adjacent pairs is distributed among other pairs containing exactly one of the two lexical units in adjacency and the other lexical unit in the surrounding text. Table 3.2 depicts the smoothing effect of the weighted-window estimator for three pairs of words. As the window size increases the number of distinct pairs will increase and the probability mass distributed accordingly. For example, the probability of "NEW" and "YORK" drops from $2.09E-04$ (window size equals 2 words) to $2.49E-06$ (window size equals 128 words), whereas the probability of "SUCCINCTLY" and "FREELY" increases from 0 to $2.69E-11$.

Document estimator

This estimator uses document frequencies rather than corpus frequencies. The frequency of a lexical unit b is denoted by D_b and corresponds to the number of documents in which the word appears, with no regard to how frequently it occurs in a particular document. The number of documents in the corpus is denoted by D.

The co-occurrence frequency of two lexical units b and d, denoted by $D_{b,d}$,

$$\hat{f}(b,d) = |D_{b,d}| \tag{3.7}$$

where $D_{b,d}$ is the set of documents where the both lexical units are present. If we require only that the words co-occur in the same document, no distinction is made between distantly occurring words and adjacent words. This effect can be reduced by imposing a maximal distance for co-occurrence, (i.e. a fixed-sized window). In this case, the frequency will be the number of documents where the lexical units co-occur within that distance. The MLE for the co-occurrence under this approach is

$$\hat{P}(b,d) = \frac{\hat{f}(b,d)}{D} \tag{3.8}$$

An early use of this estimator for co-occurrence probabilities is due to Lesk [64]. He used three collections to extract co-occurrence frequencies: the ADI collection, containing 82 short papers; the IRE collection, with 780 abstracts in computer science; and the Cranfield collection, containing 200 abstracts in aeronautics. Unfortunately, those are rather small in comparison with today's collections. Also, in all three collections the documents are small, thus a window for co-occurrence within the document has little impact. Peat and Willet [82] also used a document estimator for co-occurrence frequency with no regard to the co-occurrence distance. More recently, Lafferty and Zhai [59] used documents to extract words to expand a query based on each of its terms, with no regard to the co-occurrence distance. The approach of Lafferty and Zhai is thus similar to that of Lesk; however, it uses sampling techniques to accomplish its goals.

3.2 Affinity Measures

The affinity between two lexical units can be calculated by their direct co-occurrence or through the use of supporting evidence (i.e., by their mutual affinity with other lexical units). For instance, the lexical units "Doctor" and "Nurse" have high affinity because they co-occur in many texts, but they are also related because they co-occur with "Hospital", "Patient", "Intensive Care Unit", and "Surgery". The first approach—direct comparison—is simple and efficient since only the lexical units under consideration need to be measured from the corpus. The latter—using supporting evidence—is expensive, particularly for large corpora, since affinity between the two lexical units under consideration and the supporting lexical units must be measured.

Alternatively, the measures presented below can be categorized in four groups [38]:

- Statistical tests

  These measures are used to in hypothesis testing and the null hypothesis is that the co-occurrence is due by chance, such as the case of χ^2 and Z-Score. On the other hand, the log-likelihood is a measure of the unexpectedness of co-occurrence based on the assumption that the occurrences are described by a binomial distribution. This measures are normally not intended to be used to rank co-occurrence significance but provide a numerical outcome

that can be used for that purpose.

- Association Strength

  These measures are not intended to capture the significance of the co-occurrence with a null hypothesis in hand. Instead, their aim is to assign a score for the strength of the affinity. Examples include Dice, Jaccard and Cosine coefficients.

- Information Theoretic measures

  These measures have their origin in information theory and their aim is to compute information content of events. Examples include Mutual Information, Pointwise Mutual Information and Jensen-Shannon divergence.

- Heuristic Measures

  Derived measures fall in this category since their foundations are not formally derived. Skew divergence is an example of a heuristic measure.

3.2.1 Direct Comparison

Along with the co-occurrence frequencies, or corresponding probabilities, the strength of the association needs to address the marginal probabilities, since more frequent lexical units have a greater chance of co-occurring with other lexical units. We describe some of the most common formulas used in the literature.

Notation: Let X be a binary random variable for some lexical unit x in the language, with range $\mathcal{A}_x = \{x, \overline{x}\}$, indicating the presence or absence of the respective lexical unit. Let $\hat{P}(x)$ be the estimation for the marginal probability of x given its individual frequency $\hat{f}(x)$. Let Y be a binary random variable for lexical unit y and $x \neq y$. The estimated joint probability between x and y is denoted by $\hat{P}(x, y)$ and the co-occurrence frequency is $\hat{f}(x, y)$.

Mutual Information

The Mutual Information (MI) is a measure of divergence between random variables as defined in information theory:

$$MI(X,Y) = \sum_{x \in X, y \in Y} \hat{P}(x,y) \log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)} \tag{3.9}$$

Pointwise Mutual Information

The Pointwise Mutual Information (PMI) is a measure of divergence between the observed joint and the expected probabilities under independence. It is related to the Expected Mutual Information. However, while the latter calculates the divergence of the random variables, PMI is calculated on one point of the random variables.

$$PMI(x,y) = log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)} \tag{3.10}$$

It is interesting to note that this measure is biased toward infrequent words and is not proportional to the conditional probabilities. For example:

$$PMI(x,y) = 3.00 \text{ for } \hat{P}(x,y) = 0.08,\ \hat{P}(x) = 0.1, \hat{P}(y) = 0.1$$
$$PMI(x,y) = 6.32 \text{ for } \hat{P}(x,y) = 0.008, \hat{P}(x) = 0.01, \hat{P}(y) = 0.01$$

Dice and Jaccard coefficients

The Dice coefficient was originally proposed under the name of *coincidence index* [35]. Dice's intention was to correct the high variance of the ratio between observed and expected values (as used in PMI). This correction is obtained by taking the harmonic mean of the individual probabilities, thus reducing the effect of unbalanced marginals:

$$D(x,y) = \frac{\hat{P}(x,y)}{\frac{1}{2} \cdot \hat{P}(x) + \frac{1}{2} \cdot \hat{P}(y)} = \frac{2 \cdot \hat{P}(x,y)}{\hat{P}(x) + \hat{P}(y)} \tag{3.11}$$

Like Dice's coefficient, the Jaccard coefficient is a measure that compares the observed joint frequency with the maximum value it could assume [102]. The difference between them stems from the fact that the Jaccard coefficient subtracts the joint probability from both the numerator and denominator:

$$J(x,y) = \frac{\hat{P}(x,y)}{\hat{P}(x) + \hat{P}(y) - \hat{P}(x,y)} \tag{3.12}$$

These coefficients are monotonic to each other [102] and, unlike PMI, these coefficients are proportional to the conditional probability. For example:

$$D(x,y) = 0.8 \; J(x,y) = 0.67 \text{ for } \hat{P}(x,y) = 0.08, \hat{P}(x) = 0.1, \hat{P}(y) = 0.1$$
$$D(x,y) = 0.8 \; J(x,y) = 0.67 \text{ for } \hat{P}(x,y) = 0.008, \hat{P}(x) = 0.01, \hat{P}(y) = 0.01$$

χ^2-test

The χ^2-test is a test of statistical significance for bivariate tabular analysis. The null hypothesis in this test is that the two variables are not different and this is calculated by summing the square of the difference between the observed values and expected values computed from the marginals probabilities.

$$\chi^2 = \sum_{x \in X} \sum_{y \in Y} \frac{[\hat{P}(x,y) - \hat{P}(x) \cdot \hat{P}(y)]^2}{\hat{P}(x) \cdot \hat{P}(y)} \tag{3.13}$$

The use of χ^2 as an alternative for measuring association between words is described in Manning and Schütze. Brin *et al.* [11] also use χ^2 as an association measure.

Cosine coefficient

This measure is similar to Dice and Jaccard. The biggest difference is that the normalization is done against the geometric mean of the marginal probabilities. The affinity value is given by

$$Cos(x,y) = \frac{\hat{f}(x,y)}{\sqrt{\hat{f}(x) \cdot \hat{f}(y)}} \tag{3.14}$$

Lesk [64] was one of the first to apply the cosine measure to concept association; he used a document estimator for the frequencies. Peat and Will*et al.* also used this measure [82] to analyze the limits of co-occurrence for information retrieval.

Z-Score

The Z-Score is a normalized value that describes how far a value is from the expected value of a random variable representing independence. There are two assumptions here: the first is that the words follow a binomial distribution; the second is that the expected joint frequency $\hat{f}(x, y)$ approximates the normal distribution. The general formula for the Z-Score is

$$Z = \frac{v - \mu}{\sqrt{\sigma^2}} \tag{3.15}$$

where v is the observed value to which we calculate the Z-Score. In our case, v is the measured co-occurrence frequencies, μ is the expected value under independence, and σ is the standard deviation under independence:

$$v = \hat{f}(x, y) \tag{3.16}$$

$$\mu = N \cdot \frac{\hat{f}(x) \cdot \hat{f}(y)}{N^2} = \frac{\hat{f}(x) \cdot \hat{f}(y)}{N} \tag{3.17}$$

$$\sigma^2 = N \cdot \frac{\hat{f}(x)\hat{f}(y)}{N^2} \cdot \left(1 - \frac{\hat{f}(x)\hat{f}(y)}{N^2}\right) \tag{3.18}$$

For practical purposes we assume that

$$1 - \frac{\hat{f}(x)\hat{f}(y)}{N^2} \quad \approx \quad 1$$

since even the most frequent words have a low absolute frequency. Thus, we can rewrite equation 3.18 as

$$\sigma^2 \approx N \cdot \frac{\hat{f}(x)\hat{f}(y)}{N^2} = \frac{\hat{f}(x) \cdot \hat{f}(y)}{N} \tag{3.19}$$

Finally, we can write equation 3.15 as

$$Z(x,y) = \frac{\hat{f}(x,y) - \frac{\hat{f}(x)\cdot\hat{f}(y)}{N}}{\sqrt{\frac{\hat{f}(x)\cdot\hat{f}(y)}{N}}} \tag{3.20}$$

An early application of the Z-score as an associative measure is due to Berry-Rogghe [8]. The use of the Z-score with a binomial approximation was used by Vechtomova *et al.* [103] to analyze candidate terms for implicit query expansion in information retrieval.

Log-Likelihood

The likelihood ratio test provides an alternative for checking two simple hypotheses. Dunning [36] used a likelihood ratio to test word similarity under the assumption that the words in text have a binomial distribution.

The two hypotheses used by Dunning are: H1:$P(d|b) = P(d|\neg b)$ (i.e. they occur independently); and H2: $P(d|b) \neq P(d|\neg b)$ (i.e. not independent). These two conditionals are used in the likelihood function $L(P(d|b), P(d|\neg b); \theta)$, where θ for H2 represents the parameter of the binomial distribution $b(n, k; \theta)$. Under hypothesis H1, $P(d|b) = P(d|\neg b) = p$, and for H2, $P(d|b) = p_1, P(d|\neg b) = p_2$.

$$\log\lambda = \log\frac{L(P(d|b); p)\cdot L(P(d|\neg b); p)}{L(P(d|b); p_1)\cdot L(P(d|\neg b); p_2)} \tag{3.21}$$

Effects of the Marginals

A difference among the presented measures can be seen by plotting the effect of the marginals on the outcome of the formulas, given a fixed co-occurrence probability (or frequency), as shown in Figures 3.3 to 3.8. For these, the fixed probability is $P(x,y) = 0.00002$ and $P(x)$ and $P(y)$ are in the range $[0.0002; 0.001]$. The Dice and Jaccard coefficients have the same gradient; the same occur with Cosine coefficient and Z-score. It is interesting to note that the Dice and Jaccard coefficients are monotonic.

3.2.2 Comparison using Supporting Evidence

Similarity between two lexical units can also be derived by the use of supporting evidence $C = \{w'_1, w'_2, ..., w'_n\}$. The affinities between two lexical units b and d is calculated indirectly by the affinity between b and C and d and C. This approach is commonly referred to as a second-order affinity or statistics.

Cosine of Pointwise Mutual Information

To compare the affinity between b and d using C and this measure, two vectors are created: the first contains the PMI between b and every element of C; the same is done to d in the second vector. The cosine value between the two vectors corresponding to b and d represents the similarity between the two lexical units as depicted in equation 3.22.

$$CP(b;d) = \frac{\sum\limits_{w' \in C} PMI(w',b)PMI(w',d)}{\sqrt{\sum\limits_{w'} PMI(w',b)^2}\sqrt{\sum\limits_{w'} PMI(w',d)^2}} \tag{3.22}$$

The outcome is a value from zero to one where values closer to one indicate greater similarity. Pantel [78] used the cosine of pointwise mutual information to uncover word sense from text.

L_1 norm

In this method, the conditional probability of each word w'_i in C given b (and d) is computed. The accumulated distance between the conditionals for all words in context represents the similarity between the two lexical units, as shown in equation 3.23. This method was proposed as an alternative word similarity measure in language modeling to overcome zero-frequency problems of bigrams [31]. Rapp [84] uses this measure for word associations.

$$L_1(b;d) = \sum_{w' \in C} |\hat{P}(w'|b) - \hat{P}(w'|d)| \tag{3.23}$$

Contextual Average Mutual Information

In this measure, the conditional probabilities between each lexical unit in the context and the two lexical units b and d are used to calculate the mutual information of the conditionals (equation 3.24). This method was also used in Dagan *et al.* [31].

$$AMIC(b;d) = \sum_{w'} \hat{P}(w'|b) log \frac{\hat{P}(w'|b)}{\hat{P}(w'|d)} \tag{3.24}$$

Contextual Jensen-Shannon Divergence and Skew Divergence

These measures can be seen as alternative to address the zero frequency problem of the Mutual Information formula (equation 3.24). Instead of using the probabilities directly, they are averaged between the two distributions. It also provides a symmetric measure (AMIC is not symmetric). This method was also used in Dagan *et al.* [31]. Given the Kullback-Leibler divergence between two distributions R and Q:

$$KL(R(x)\|Q(x)) \quad = \quad E\left[R(x)\ log\frac{R(x)}{Q(x)}\right]$$

and the average between the two probabilities,

$$AVGP \quad = \quad \frac{\hat{P}(w'|b) + \hat{P}(w'|d)}{2}$$

the Jensen-Shannon Divergence is defined as

$$JS(b;d) = \frac{KL(\hat{P}(w'|b)\|AVGP) + KL(\hat{P}(w'|d)\|AVGP)}{2} \tag{3.25}$$

The Skew divergence [62] also addresses the zero frequency problem, but instead of using the average it skews one of the distribution towards the other by a small amount,

$$SD(b;d) = KL(\hat{P}(w'|b) \parallel [\alpha \cdot \hat{P}(w'|d) + (1-\alpha) \cdot \hat{P}(w'|b)]) \tag{3.26}$$

for any α such that $0 < \alpha < 1$; $\alpha = 0.99$ is a typical value for good performance [63].

Pointwise Mutual Information of Multiple words

Turney [100] proposes a different formula for Pointwise Mutual Information when context is available, as depicted in equation 3.27. The context is represented by C', which is any subset of the context C. In fact, Turney argued that bigger C' sets are worse because the resulting frequencies are smaller and as consequence can be affected by noise. Therefore, Turney used only one word c_i from the context, discarding the remaining words. The chosen word was the one that has biggest pointwise information with b.

It is interesting to note that the equation $P(b|d, C)$ (or $P(d|b, C)$) is not equivalent to $P(b|H)$ or $P(d|H)$ from *n-gram* model, since no ordering is imposed on the lexical units and also due to the fact that they can be separated from one another by other words.

$$PMIC(b, d; C') = \frac{\hat{P}(b, d, C')}{\hat{P}(b, C')\hat{P}(d, C')} \tag{3.27}$$

Latent Semantic Analysis - LSA

This technique uses a matrix decomposition based on its singular values in order to capture the latent information in the matrix. The matrix M contains in one dimension one entry for each word in the collection and on the other dimension an entry for each document (i.e. $t \times d$); thus the frequencies are obtained by means of a document estimator. The singular value decomposition is $M = MT \times MS \times MD$, where MT and MD have orthonormal columns and MS is a diagonal matrix in which the diagonal values are in decreasing order. The dimensions of the matrices are $MT = t \times k$, $MS = k \times k$ and $MS = k \times d$, where k is the rank of M. After the decomposition, a cut-off can be applied on k, which simulates the process of removing the less informative singular values of M. From another perspective, this process represents the removal of factors that contribute less to the "semantics" of the original matrix, i.e. it is a filtering process.

The association is obtained by computing the cosine of the words in the matrix resulting from the decomposition plus a singular value cut-off process.

This technique was proposed by Deerwester *et al.* for information retrieval purposes [33]. Landauer and Dumais [60] used LSA to answer synonym questions.

One of the main challenges for LSA is the initial computation resources it requires since the matrix M should be first constructed and then the singular values extracted. Particularly, when the dimensions correspond to vocabulary and documents in the corpus the curse of dimensionality is a real issue. It happens that, in most cases, the original matrix is sparse. This can be used to make the process more efficient [7].

3.3 Summary

We presented two new point estimators for co-occurrence frequency. In the first, weighted-window estimator, co-occurrences in proximity have more weight than those at farther distance. In the second estimator, document estimator, the co-occurrence frequency is measured in the number of documents and not in the occurrences in the corpus. For that estimator, a window for co-occurrence within the document is also used, filtering out co-occurrences at farther distances. Along with the new estimators we described another estimator, the simple estimator, to be used as a baseline in our experimental evaluation. We also presented existing methods that use co-occurrence to compute affinity, making use of context or not.

Figure 3.3: Effect of marginals on PMI

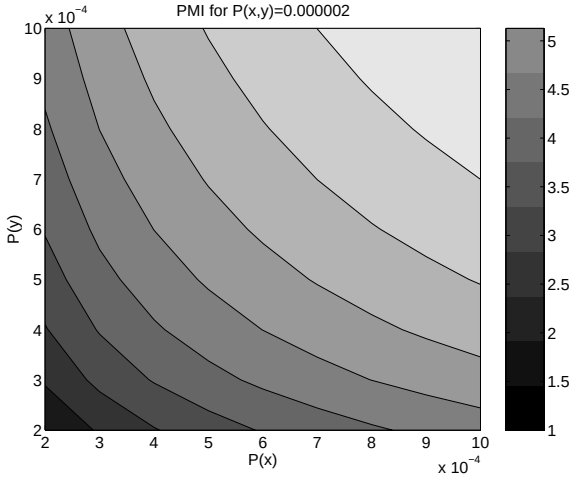


Figure 3.4: Effect of marginals on χ^2 score

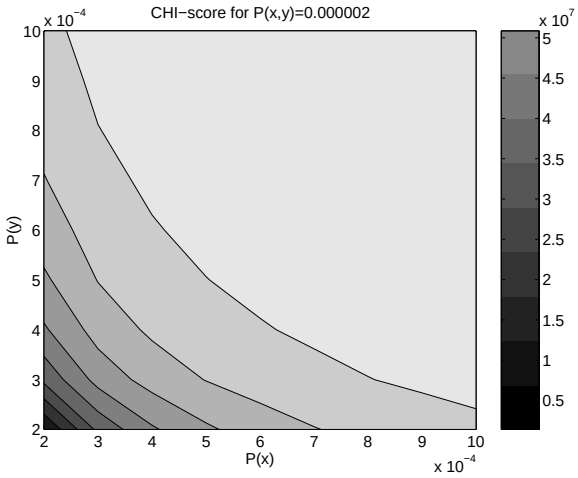


Figure 3.5: Effect of marginals on Dice

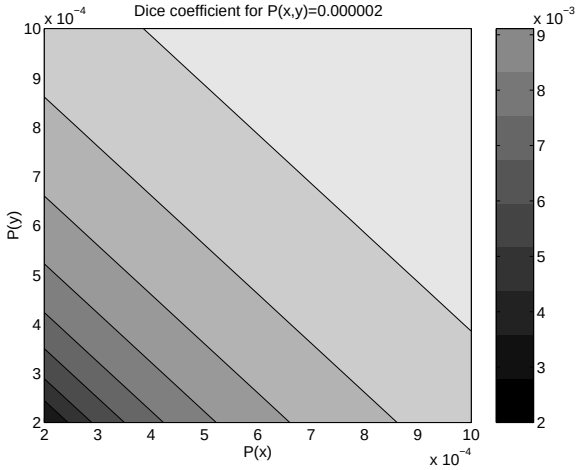


Figure 3.6: Effect of marginals on Jaccard

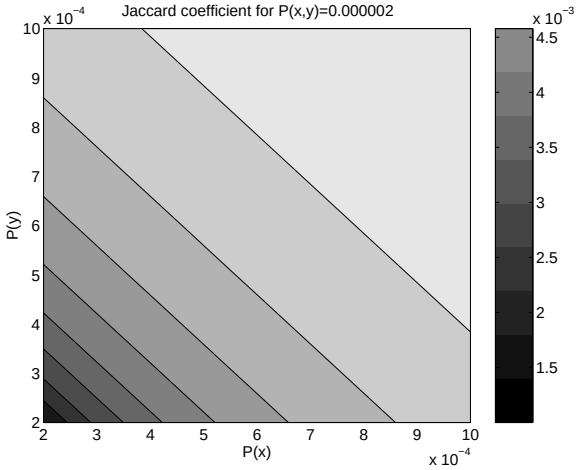


Figure 3.7: Effect of marginals on Cosine

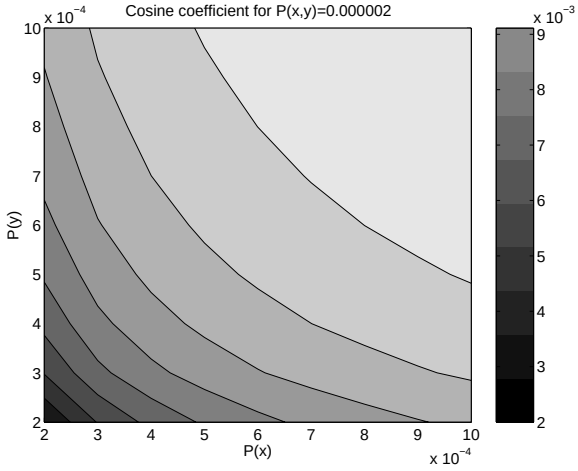
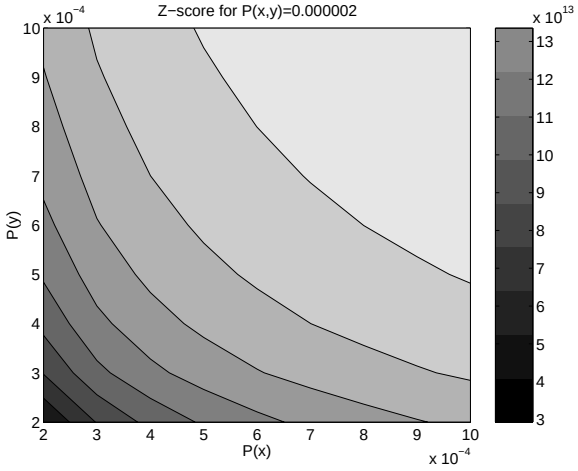


Figure 3.8: Effect of marginals on Z-score



Chapter 4

Affinity Models

This Chapter presents a novel and completely different approach to compute affinity. A new independence assumption is formulated and a model for affinity is presented. The whole distance distribution is recovered for the creation of these models and can also be used against the independence model to compute affinity between lexical units. A fast algorithm to compute the empirical distribution is also presented.

4.1 Empirical Distribution and Models for Lexical Affinities

We now present new models for lexical affinity. They are based on the empirical distribution and are made more accurate by using consistent estimators applied to large corpora. Two models are presented: an independence model for pairs of lexical units, and an affinity model. The independence model estimates the likelihood of co-occurrence at any given distance when no relationship is expected between the two lexical units. The affinity model is used to fit the observed data and can estimate expected number of co-occurrences at any given distance between two lexical units, taking the lexical affinity between the units into account.

Notation: Let G be a random variable with range comprising all 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 $C_\Delta(b,d)$.

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, $\Delta \sim Geometric(\delta;p)$. 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 certain position for b, then if independent, the next d will occur with probability $P_g(d) \cdot (1 - P_g(d))^{\delta-1}$ at distance δ from b. Therefore, the observed mean is then the expected distance of the geometric distribution with parameter p.

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

$$p = \frac{1}{\mu} = \frac{1}{\frac{1}{n}\sum_{\delta=1}^{\infty} f_\delta} \tag{4.1}$$

By scanning a large corpus, we can observe μ; thus fitting the independence model is straightforward.

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 Beeferman *et al.* [3] and Niesler [77] call *self-triggers*. In practice, this assumption leads to the frequency counting of pairs with no intervening b or d.

Figure 4.1 shows that the geometric distribution fits well the observed distance of independent words, in this case the words "DEMOCRACY" and "WATERMELON". When a dependency exists, the geometric model does not fit the data well, as can be seen in Figure 4.2. Since the geometric

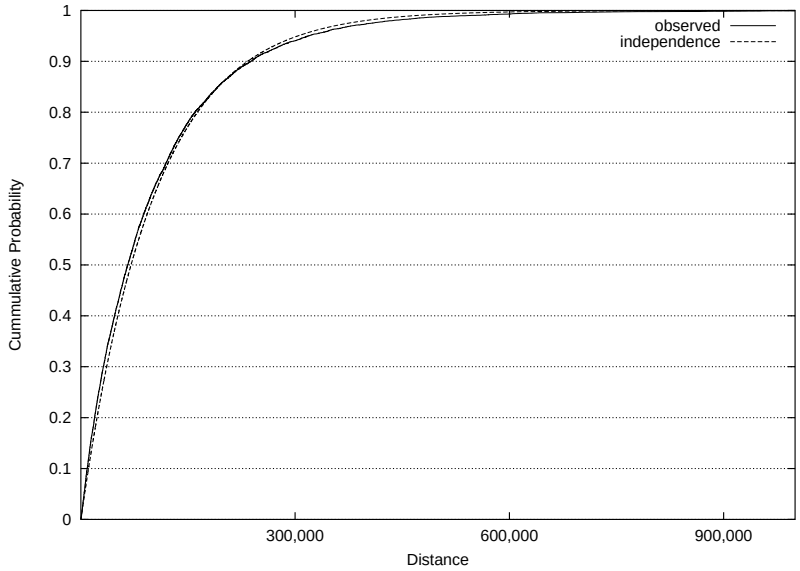


Figure 4.1: $C_{\Delta}(watermelon, democracy)$

and exponential distributions represent related idea in discrete/continuous spaces it is expected that both have similar results, especially when $p \ll 1$.

Affinity Model

The model of affinity follows a exponential-like distribution, as in the independence model. Other researchers have also used exponential models for affinity [3, 77]. 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 each other, and the gamma distribution is a good choice to model this phenomenon.

$$Gamma(\Delta = \delta; \alpha, \beta) = \frac{\delta^{\alpha-1} e^{-\delta/\beta}}{\beta^{\alpha}\Gamma(\alpha)} \tag{4.2}$$

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 are calculated using maximum likelihood estimation for the gamma distribution: Let $x_1, x_2, ..., x_n$ be the observed values we want to fit with the gamma distribution. The likelihood of observing these points is $L(x_1, x_2, .., x_n)$:

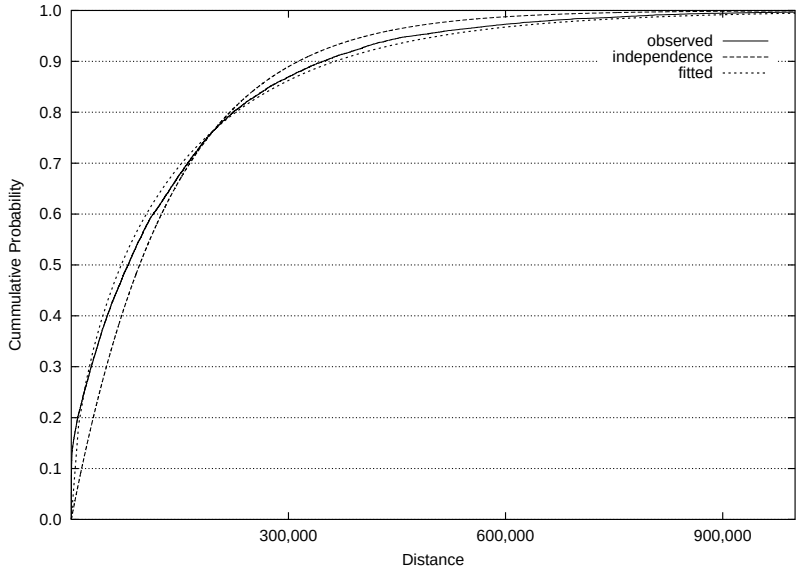


Figure 4.2: $C_{\Delta}(watermelon, fruits)$

$$
\begin{aligned}
L(x_1, x_2, .., x_n) &= \frac{x_1^{\alpha-1}e^{-x_1/\beta}}{\beta^{\alpha}\Gamma(\alpha)}\frac{x_2^{\alpha-1}e^{-x_2/\beta}}{\beta^{\alpha}\Gamma(\alpha)}...\frac{x_n^{\alpha-1}e^{-x_n/\beta}}{\beta^{\alpha}\Gamma(\alpha)} \\
logL(x_1, x_2, .., x_n) &= log\frac{x_1^{\alpha-1}e^{-x_1/\beta}}{\beta^{\alpha}\Gamma(\alpha)}\frac{x_2^{\alpha-1}e^{-x_2/\beta}}{\beta^{\alpha}\Gamma(\alpha)}...\frac{x_n^{\alpha-1}e^{-x_n/\beta}}{\beta^{\alpha}\Gamma(\alpha)} \\
logL(x_1, x_2, .., x_n) &= (\alpha-1)\left(\sum_{i=1..n}\log x_i\right) - \left(\sum_{i=1..n}\frac{x_i}{\beta}\right) - n\alpha\log\beta - n\log\Gamma(a)
\end{aligned}
$$

The likelihood is maximized by setting the partial derivatives to zero,

$$
\frac{\partial}{\partial\beta}\log L(x_1, x_2, .., x_n) = \left(\sum_{i=1..n}\frac{x_i}{\beta^2}\right) - \frac{n\alpha}{\beta} = 0
$$

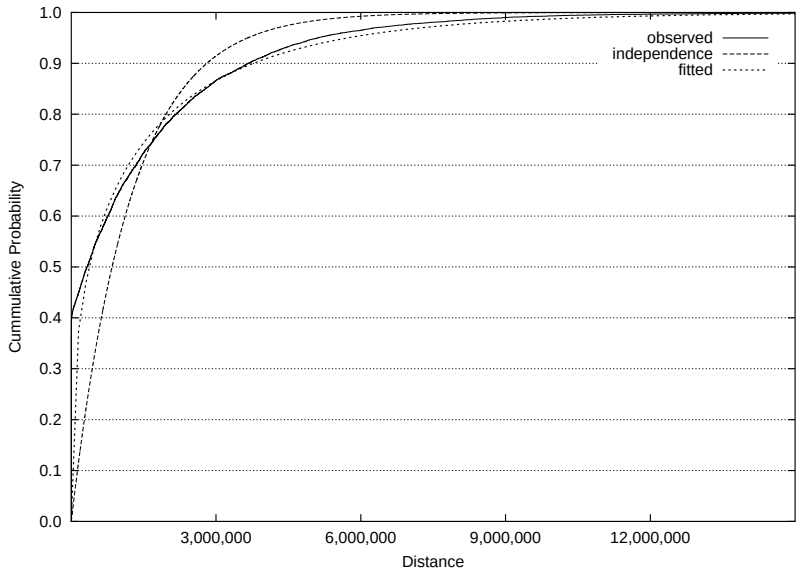


Figure 4.3: $C_\Delta(watermelon, watermelon)$

or

$$\beta = \frac{1}{\alpha} \sum_{i=1..n} \frac{x_i}{n} = \frac{\mu}{\alpha} \tag{4.3}$$

and by

$$\frac{\partial}{\partial \alpha} \log L(x_1, x_2, .., x_n) = \left(\sum_{i=1..n} \log x_i \right) - n \log \beta - n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} = 0$$

where we can substitute β from equation 4.3:

$$\begin{aligned} \sum_{i=1..n} \log x_i - n \log \beta - n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} &= 0 \\ \sum_{i=1..n} \log x_i - n \log \mu - n \log \alpha - n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} &= 0 \\ \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} - \log \alpha &= \frac{1}{n} \left(\sum_{i=1..n} \log x_i \right) - \log \left(\frac{1}{n} \sum_{i=1..n} x_i \right) \end{aligned}$$

The general case, using the histogram frequencies of all pair distances (instead of sample

values) is given by

$$\beta = \frac{1}{\alpha} \sum_{\delta=1}^{\infty} \frac{x_i}{n} = \frac{\mu}{\alpha} \tag{4.4}$$

$$\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} - \log \alpha = \frac{1}{n} \left(\sum_{\delta=1}^{\infty} f_\delta \log \delta \right) - \log \left(\frac{1}{n} \sum_{\delta=1}^{\infty} \delta f_\delta \right) \tag{4.5}$$

Figure 4.2 shows the fit of the gamma distribution to the word pair FRUITS and WATERMELON. The affinity model can also be used to fit self-affinity as depicted in Figure 4.3. As shown, the affinity between same lexical units is far from the independence, a fact also noted by Church [19].

To fit the affinity model we observe μ directly from the corpus, as in the independence model. The value of the parameter α is computed by numerically solving equation 4.5, which can be accomplished by observing the right-hand side values from the corpus. The value of β is trivially calculated from equation 4.4 and from the values of α and μ.

Affinity Models and Smoothing

The existence of a function to estimate the number of occurrences of a pair of lexical unit at any distance provides a solution to the zero frequency problem. We can now infer the number of occurrences for pairs at distances that are not seen in the training data. This is a desirable feature for a language model. For example, in the sequential bigram model, a common smoothing strategy is to back-off to unigram probabilities or interpolate bigram and unigram probabilities. The distance model could be used in that case by estimating the probability of the unseen sequential bigram based on a model for distant bigrams.

Another smoothing effect can be achieved by using the cumulative probability of distance co-occurrences. Pairs at farther distances are smoothed by the counts of closer co-occurrences. Using the cumulative is optional, but if used it will favor closer co-occurrences. It is a natural way to handle the problem that the weighted-window estimator addresses by means of an artificial parameter (window size).

4.2 Efficient Retrieval

The independence and affinity models presented in chapter 4 depend on a good approximation to the mean distance μ of lexical unit pairs. The estimator for μ is a consistent estimator, and as such will provide a better estimation as the corpus size increases. Therefore, we want to scan the whole corpus efficiently in order to make this framework usable.

Fast Computation of Distance Models

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 document 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. 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 of 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 be 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:

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

and

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

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 $O(\log f_t) \le O(\log N)$ 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;
  $k \leftarrow v+1$;
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;
  $k \leftarrow v+1$;
end while;

These two algorithms are implemented in the MultiText engine [22]. They correspond to standard operators of its language (GCL) which extend boolean operators by including, among others, containment and ordering operators. The symmetric algorithm above is similar to the boolean operator *and* ($\triangle$) and the asymmetric algorithm is similar to the ordering operator *followed by* ($\diamond$).

	Time
Fastest	20ms
Average	310.32 ms
Slowest	744ms

Table 4.1: Scanning performance on 99 word pairs

Pair		Time
b	d	(ms)
BREAD	BUTTER	20
SMOOTH	ROUGH	22
SOFT	HARD	65
QUIET	LOUD	67
THIRSTY	WATER	169
WOMAN	MAN	218
TABLE	CHAIR	370
WHITE	BLACK	451
WISH	WANT	533
HIGH	LOW	698
HOUSE	HOME	744

Table 4.2: Examples of scanning performance

Table 4.1 illustrates the time required to scan all co-occurrences of given pairs of terms. Table 4.2 shows scanning performance of some pair examples. The collection is distributed over 17 hosts. We report the time for one host to return its results.

4.3 Summary

We present a framework for the fast computation of lexical affinity models. The framework is composed of an algorithm to efficiently compute the co-occurrence distribution between pairs of terms, an independence model, and a parametric affinity model. In comparison with point estimation models, which either use arbitrary windows to compute similarity between words or use lexical affinity to create sequential models, in this chapter we focus on models intended to capture the co-occurrence patterns of any pair of words or phrases at any distance in the corpus.

Chapter 5

Human-oriented Language Tests

In this chapter we examine the application of different lexical affinity methods to solve two human-oriented language tests: a set of synonym questions from TOEFL (section 5.1) and a set of GRE fill-in-the-blanks practice questions (section 5.2). For the synonym questions, we use both point estimation and affinity models from Chapters 3 and 4 and propose the use of new measures, the *skew* and *log-likelihood ratio over intervals*, that can be calculated if the affinity models are available. In the fill-in-the-blanks practice questions we use the parametric affinity and independence models. Our evaluation on the synonym questions also aims to determine, among all alternatives, which settings perform well in order to generalize the results to other applications. In particular, the results obtained in this evaluation drive the choice of affinity measures used in Chapter 6.

5.1 Synonym questions

We evaluate the affinity measures proposed in Chapters 3 and 4 using three test sets. The first test set is a set of TOEFL questions first used by Landauer and Dumais [60]. This test set contains 80 synonym questions. For each question there is one target word—TW—and a set of synonym alternatives A with four options. The other two test sets, which we will refer to as TS1 and TS2, are practice questions for the TOEFL. These two test sets also contain four alternative options, $|A| = 4$, and TW is given in context C (i.e., TW appears in the context of a sentence). TS1 has

TW = "CONCISELY"
A = { 'SUCCINCTLY','POWERFULLY','POSITIVELY','FREELY'}

C = "THE COUNTRY IS PLAGUED BY TURMOIL."
TW = "TURMOIL"
A = { 'CONSTANT CHANGE','UTTER CONFUSION','BAD WEATHER', 'FUEL SHORTAGES'}

C = "FOR ALL THEIR PROTESTATIONS, THEY HEEDED THE JUDGE'S RULING."
TW = "FOR"
A = { 'IN SPITE OF','BECAUSE OF','ON BEHALF OF','WITHOUT'}

Figure 5.1: Examples of synonym questions

50 questions and was also used by Turney [100]. TS2 has 60 questions extracted from a TOEFL practice guide [55]. These three test sets have particular compositions. TOEFL contains only single words and balanced parts-of-speech: verbs, adjectives, nouns and adverbs are in similar proportions. TS1 contains 20% adverbs and adjectives, 80% verbs and nouns, and 2 compounds. TS2 has 18 compounds and prepositions, such as the third example in Figure 5.1.

For all test sets the answer to each question is known and unique. For comparison purposes, we also use TS1 and TS2 without the context (i.e. comparing TW against all elements A and disregarding the corresponding C). Figure 5.1 shows some examples of questions with and without context. For all of the experiments, the statistics were extracted from the terabyte corpus as described in Section 2.4.1.

The TOEFL synonym test set has been used by several other researchers. It was first used in the context of Latent Semantic Analysis(LSA) [60], where 64.4% of the questions were answered correctly. Turney [100] used PMI in context and statistical estimates from a web search engine to answer the questions, achieving 73.8% correct answers. Jarmasz [49] used a thesaurus to compute the distance between the alternatives and the target word, answering 78.8% correctly. Recently, Turney [101] 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 97.5% of the questions correctly after being trained over 331 examples. With the exception of recent results of Turney [101], all previous approaches were not exclusively designed for the task of answering TOEFL synonym questions.

In fact, one of the goals of this evaluation is to generalize the methods to other applications and for that it is preferable to use simple and fast approaches to compute affinity. The efficacy of Turney's combined components approach is due to the fact that it requires components to answer the questions. From these components' answers the weights for combination are trained in a hill-climbing search procedure repeated many times to avoid getting stuck in local minima. In many applications, such as the ones presented in Chapter 6, Turney's recent approach of combination is not as well suited.

Our evaluation on the synonym questions is divided into two parts. First, we address the point estimation methods presented in Chapter 3, including the two new co-occurrence estimators. We also investigate the effect of corpus size and the effect of context in the point estimation evaluation. In the second part, two new measures derived from the affinity models presented in Chapter 4 are used to solve the sets of questions. These measures are *skew* and *log-likelihood ratio over intervals*.

5.1.1 Point Estimation

For the three test sets—TOEFL, TS1 and TS2 without context—we applied the three point estimators presented in Chapter 3. We investigated a variety of window sizes, varying the window size from 2 to 256 by powers of 2.

From all of the measures presented in Chapter 3, the log-likelihood ratio as proposed by Dunning [36] is not discussed in this section because it is provides the same ordering as MI. The Jaccard coefficient is monotonic to Dice, and as such will not be discussed either. In some of the questions, TW or one or more of the A_i's are multi-word strings. For these questions, we assume that the strings may be treated as phrases and use them "as is", adjusting the size of the windows by the phrase size when applicable.

The results for the TOEFL test set using the three point estimators are presented in Figures 5.2, 5.3, and 5.4. In terms of absolute performance, the peak—81.3%—is reached under different conditions: using PMI along with document estimator and windows of 16–32 words; using Z-score with a window of size 64 in the document estimator and windows of size 128 and 256 in the weighted-window estimator; and using the Cosine and Z-score with 16-word windows and the simple estimator.

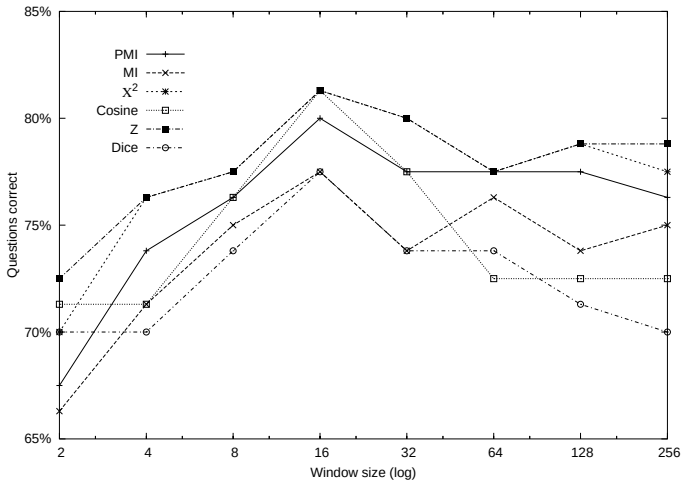


Figure 5.2: Results for TOEFL test set with Simple Estimator

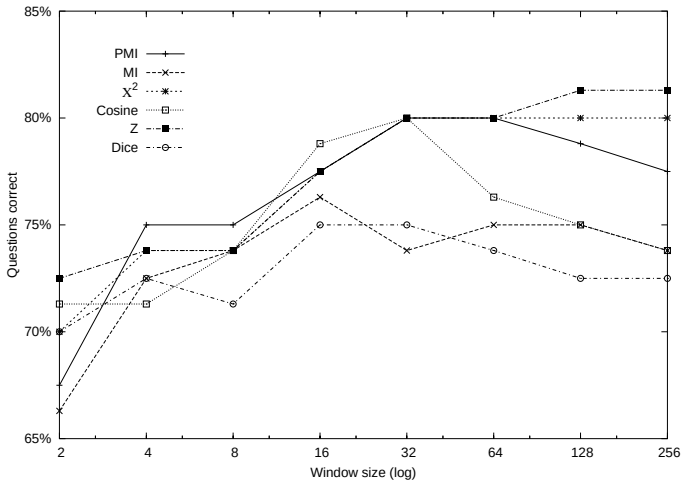


Figure 5.3: Results for TOEFL test set with Weighted-window Estimator

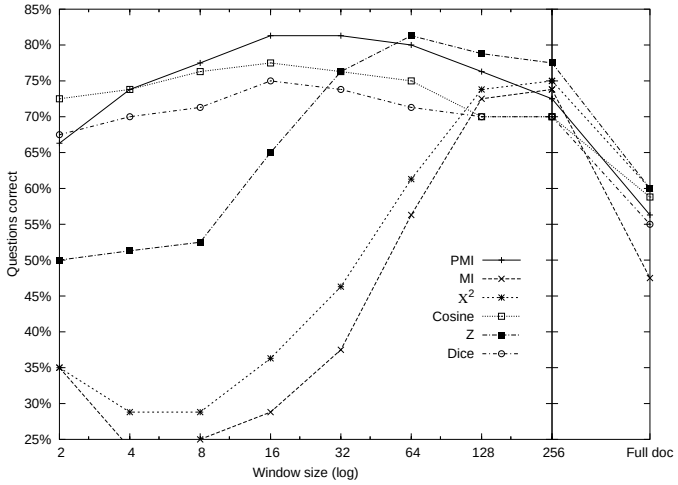


Figure 5.4: Results for TOEFL test set with Document Estimator

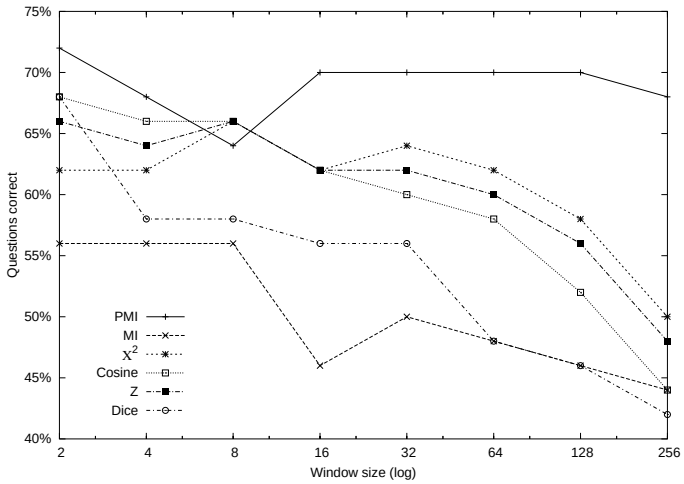


Figure 5.5: Results for TS1 test set with Simple Estimator

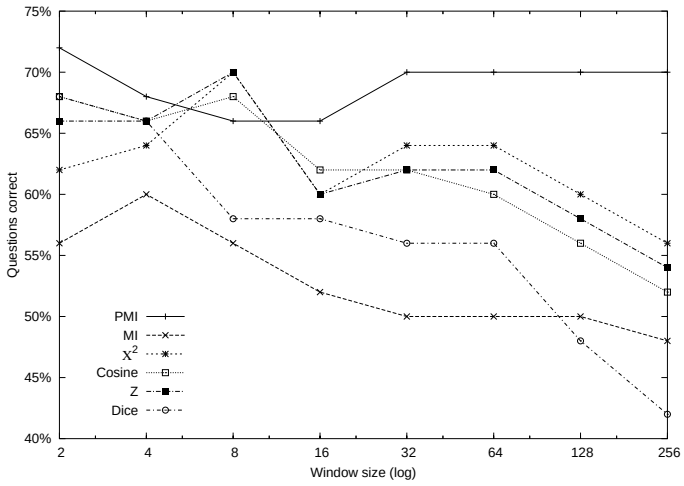


Figure 5.6: Results for TS1 test set with Weighted-window Estimator

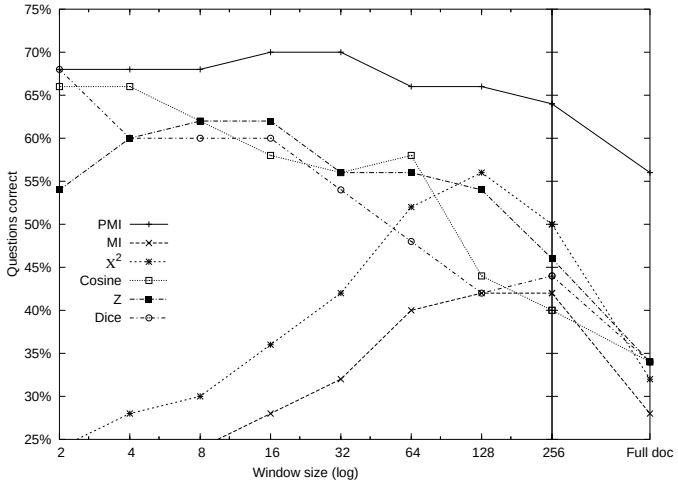


Figure 5.7: Results for TS1 test set with Document Estimator

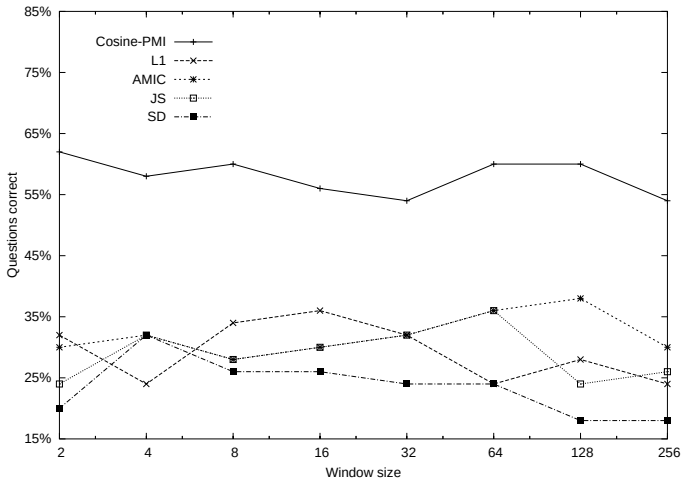


Figure 5.8: Results for TS1 using context and Simple Estimator

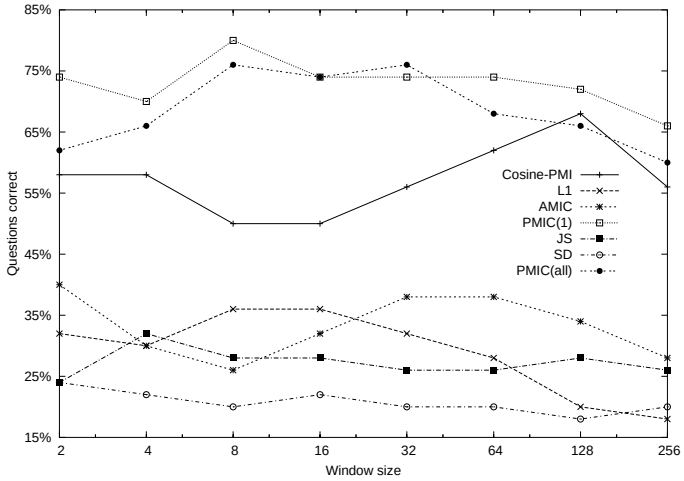


Figure 5.9: Results for TS1 using context and Document Estimator

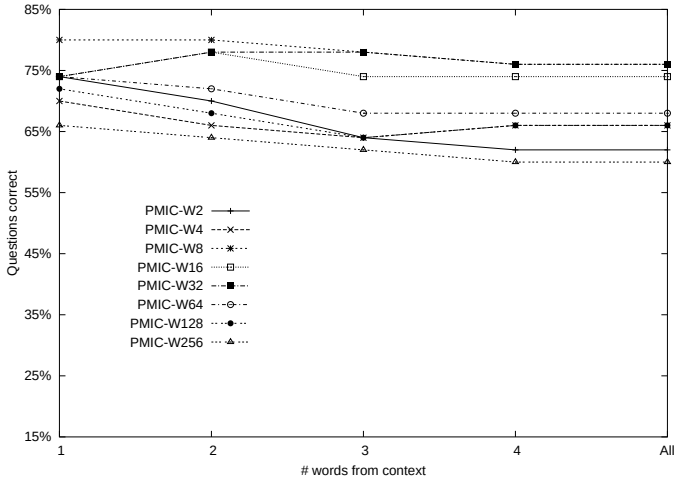


Figure 5.10: Influence from the context on TS1

Turney [100] described a run on the same tests set using a similar method: document estimator and PMI. He was able to answer 72.5% of TOEFL questions correctly, which is 10% under our best result. The difference between the results presented here and Turney's results may be due to differences in the corpora and differences in the queries. Turney used Altavista and we used our own crawl of web data. We cannot directly compare the collections since we do not know how the Altavista collection was created[1]. As for the queries, in his best result, Turney used the operator *near*[2] available at the time the experiments were performed, which specified a maximum distance of 10 words apart for its operands. In our case, we have more control in the query since we can precisely specify the window size; in fact, we can use both window estimators and document estimators and for each of those the window size can be specified.

The performance of the measures for direct comparison depends on the window size and the estimator and, in some situations such as the document estimator, exhibit a poor performance. This is the case for χ^2, MI and Z-score for small window sizes (2–16).

The results for test set TS1 using direct comparison measures are presented in Figures 5.5, 5.6, and 5.7 for the three point estimators. The best performance is 72.0%, in the simple and weighted window estimator. At this distance these two estimators yield exactly the same frequency (normalization is similar when maximum distance $K = 1$). Turney [100] also uses this test set using PMI and estimation based on Altavista (as in TOEFL test set), achieving 66.0% peak performance, 6% under of our best.

Figures 5.11, 5.12, and 5.13 show the performance of the three point estimators and six measures for test set TS2. The peak performance is 75.0% which occurs in four different situations: using PMI and document estimator with a window size of 64; Z-score and 8-word window with both simple and weighted-window estimators; and simple estimator and 16-word windows with Z-score, χ^2 and Cosine measures.

For this evaluation we wish to determine the best point estimator and measure to use. The peak performance in absolute numbers can be misleading since they may not be statistically significant. In order to compare the point estimators we summed the results of the three test sets

[1] Furthermore, it is no longer available.

[2] This operator is no longer available in Altavista either.

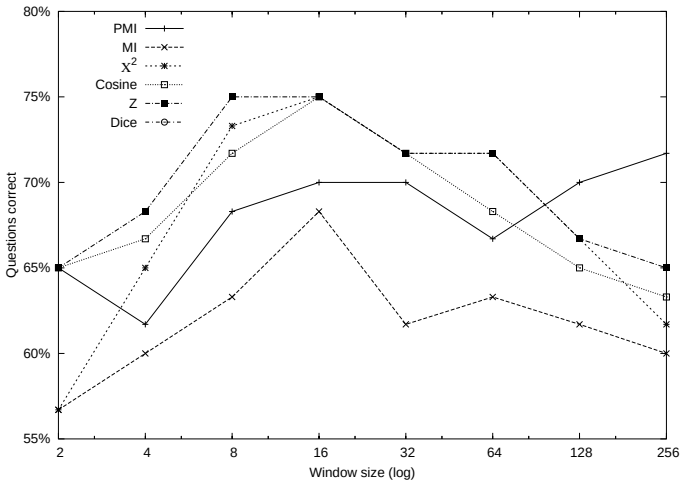


Figure 5.11: Results for TS2 test set with Simple Estimator

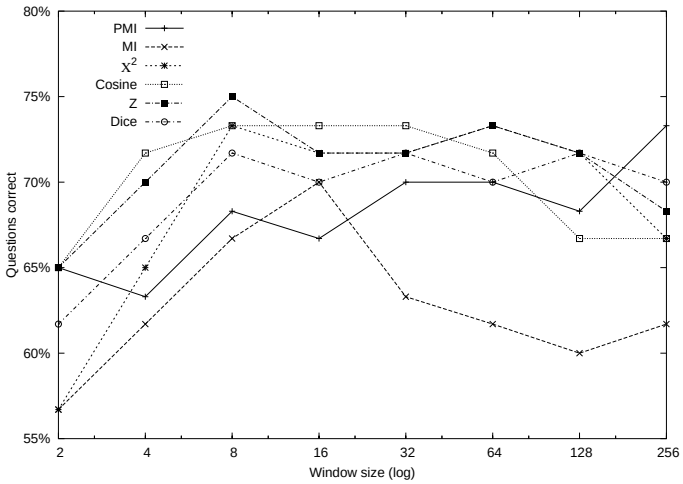


Figure 5.12: Results for TS2 test set with Weighted-window Estimator

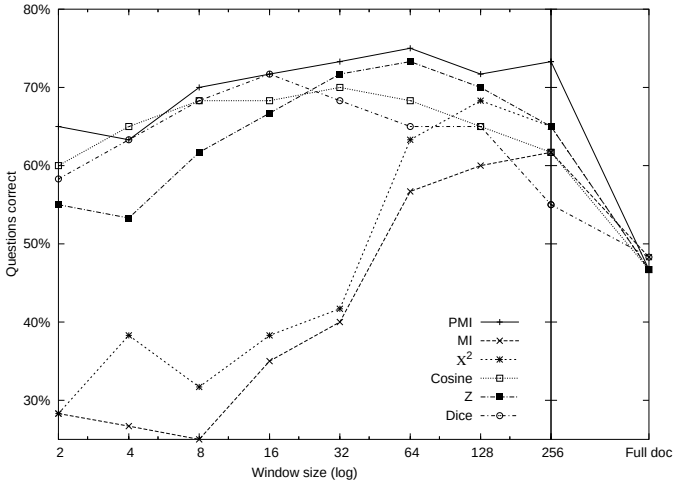


Figure 5.13: Results for TS2 test set with Document Estimator

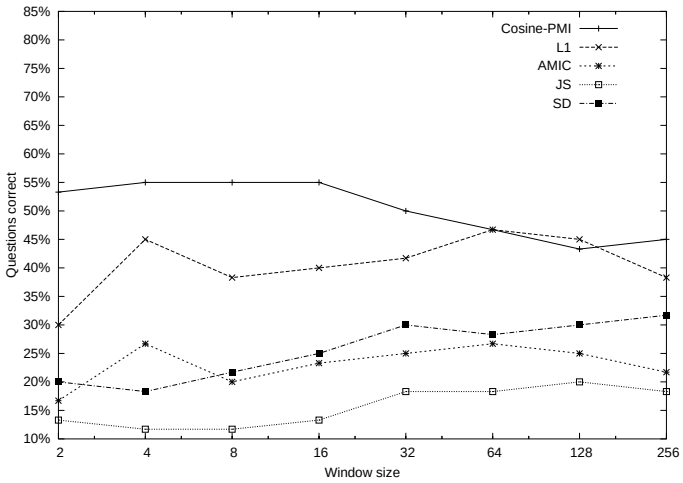


Figure 5.14: Results for TS2 using context and Simple Estimator

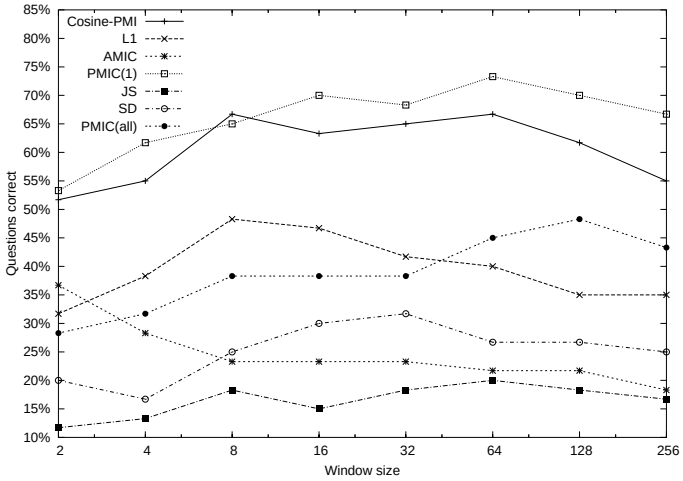


Figure 5.15: Results for TS2 using context and Document Estimator

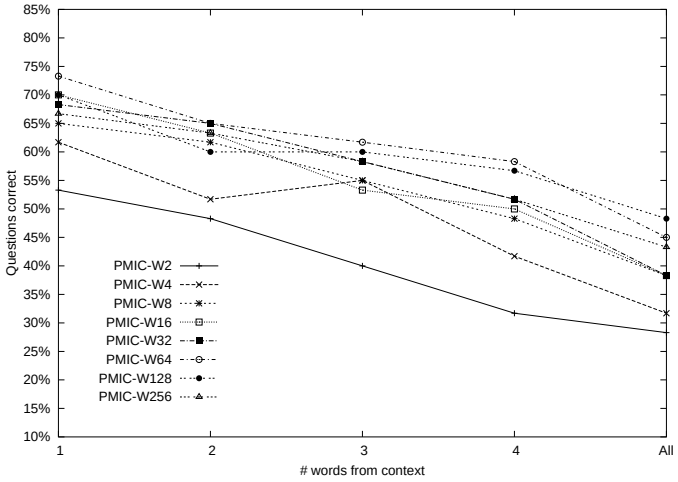


Figure 5.16: Influence from the context on TS2

Size	PMI	MI	χ^2	Cosine	Z	Dice
2	67.9	60.5	63.7	68.4	68.4	66.8
4	68.4	65.3	70.5	68.9	71.6	67.9
8	71.1	65.8	72.6	71.6	73.2	69.5
16	74.2	66.3	74.2	74.2	74.2	69.5
32	73.2	63.7	73.2	71.1	72.6	67.9
64	72.1	64.7	71.6	67.4	71.1	66.3
128	73.2	62.6	69.5	64.7	68.9	64.2
256	72.6	62.1	65.3	62.1	66.3	60.0

Table 5.1: % correct answers on the three test sets with Simple estimator

Size	PMI	MI	χ^2	Cosine	Z	Dice
2	67.9	60.5	63.7	68.4	68.4	66.8
4	69.5	65.8	68.4	70.0	70.5	68.9
8	70.5	66.8	72.6	72.1	73.2	67.9
16	71.1	67.9	71.1	72.6	71.1	68.9
32	74.2	64.2	73.2	73.2	72.6	68.9
64	74.2	64.2	73.7	70.5	73.2	67.9
128	73.2	63.7	72.1	67.4	72.1	65.8
256	74.2	63.2	69.5	65.8	70.0	63.7

Table 5.2: % correct answers on the three test sets with Weighted-window estimator

Size	PMI	MI	χ^2	Cosine	Z	Dice
2	66.3	29.5	30.0	66.8	52.6	64.7
4	68.9	24.7	31.6	68.9	54.2	65.3
8	72.6	24.7	30.0	70.0	57.9	67.4
16	75.3	30.5	36.8	69.5	64.7	70.0
32	75.8	36.8	43.7	68.9	69.5	66.8
64	74.7	52.1	59.5	68.4	72.1	63.2
128	72.1	60.5	67.4	61.6	69.5	61.1
256	70.5	61.6	65.3	59.5	65.3	58.4

Table 5.3: % correct answers on the three test sets with document estimator

respecting the window size and measure; we move to a bigger test set composed of 190 questions. The final results for the simple estimator are in Table 5.1; in Table 5.2 are the results for the weighted estimator; the document estimator is shown in Table 5.3. Since no prior preference exists for window size and measure, we compare the estimators for all combinations of window size and measure. Each of the 190 questions is paired on the same conditions in order to compute

	Simple and Weighted Estimators			Simple and Doc. Estimators			Weighted and Doc. Estimators		
	Simple better	Tie	Weighted better	Simple better	Tie	Doc better	Weighted better	Tie	Doc better
PMI	0	7	0	0	8	0	0	7	1
MI	0	7	0	6	2	0	6	2	0
χ^2	0	6	1	6	2	0	7	1	0
Cosine	0	6	1	1	7	0	3	5	0
Z	0	6	1	4	4	0	4	4	0
Dice	0	6	1	0	8	0	0	8	0

Table 5.4: Statistical comparison of the three estimators

	Dice	Tie	PMI
Simple	0	6	2
Weighted	0	6	2
Document	0	3	5

Table 5.5: PMI vs. DICE

the differences and McNemar's test was used to verify statistical significance.

McNemar's test is used to compare treatments on paired experiments. The treatments in our case are the combinations of window size, measure of affinity, and co-occurrence frequency estimator. The test ignores pairs where the two treatments' outcome is the same. For the disagreements, the null hypothesis is that they are equally distributed for the two treatments. The exact test uses a binomial distribution with probability of success $p = 0.5$ (and consequently $q = 0.5$ as well). Thus, given a number of disagreements, it is only necessary to check the probability of having this level of disagreement, accepting or rejecting the null hypothesis.

Table 5.4 shows the pair comparison of point estimators. For the simple and weighted estimators, the only significant differences are in larger windows (256 words) on χ^2, Cosine, Z-score and Dice. The comparison between simple and document estimators shows that for all measures but PMI there are one or more window sizes in which the simple estimator yields better performance. The document estimator is better than the weighted-window estimator for window size 16 using PMI; in all other situations the weighted estimator is better or equal to the document estimator. Thus, if the window size and measure are chosen with no prior preference, then the weighted estimator is like to perform better or the same in all cases but one. The simple estimator is likely

Window	PMI	MI	χ^2	Cosine	Z	Dice
2	−	+	+	−	0	−
4	−	+	+	−	0	−
8	−	+	+	−	0	−
16	−	+	+	−	−	−
32	−	0	0	−	−	−
64	−	−	−	−	−	−
128	−	−	−	−	−	−
256	−	−	−	−	−	−

Table 5.6: Full document vs. document with window constraints. "+" indicates full doc is statistically better, "−" indicates that full doc is statistically worse. "0" indicates that the difference is not significant.

to perform better or the same in all cases but four and the document estimator is likely to perform better or the same if the measure chosen is PMI.

As for the measures, PMI and the Dice coefficient are more robust in the sense that the choice of estimator will affect them in only one specific case (as long as the window size is fixed). However, PMI is never significantly worse than the Dice coefficient for the same window size, as shown in Table 5.5; thus choosing PMI will result in a better chance that the estimator will not affect the results, for whichever point estimator is chosen. The window size for PMI is best in the range 16–32 words, being statistically significant with regard to smaller window sizes [98]. In fact, a window of 32 words has the best absolute performance in the three sets combined when the document estimator is used (75.8% over the 190 questions).

The document estimator's performance is normally worse when no window for co-occurrence within the document is imposed (i.e. measuring joint frequency simply by counting the document in which both lexical units occur, regardless of their distance). As this estimation process is used in information retrieval, in particular pseudo-relevance feedback, our results suggest that these applications may be suboptimal. See Zhai and Lafferty for a recent example of such use in the IR domain [59]. This degradation in performance when no bounds on distance are imposed is common to the three test sets. In fact, as shown in Table 5.6, with the exception of small windows in MI and χ^2, using the full document as a co-occurrence unit is normally worse statistically. The problem with MI and χ^2 is due to the fact that document frequencies are much coarser than those provided by window estimators and, as such, they are more likely to generate low frequency

estimates and thus are more susceptible to noise.

Using context

The context available in TS1 and TS2 consists of sentences where the synonyms are to be chosen for the target terms. Since the two window estimators perform similarly, the results for TS1 are shown only for the simple window estimator and the document estimator in Figures 5.8 and 5.9. The performance of all measures but PMIC are worse than the non-contextual measures. However, PMIC with a window size of eight words performs as better than other measures in TS1. For TS2, no measure using context was able to perform better than the non-contextual measures. PMIC performs best overall but has worse performance than CP with a window size of 8. In this test set, the performance of CP with the document estimator is better than CP with the simple estimator. L_1 performs better than AMIC but both have poor results, JS is never better than chance and SD is an improvement over JS . The context in TS2 has more words than TS1 but the questions seem to be harder, as shown in Figure 5.1. In some of the TS2 questions, the target word or one of its alternatives uses functional words.

These results for TS1 and TS2 were not what one would expect when context is taken into account. L_1, AMIC and JS perform poorly, worse than chance for some window sizes. One difference in the results is that for PMIC only the best word from the context is used, as proposed by Turney [100], while the other methods used all words but stopwords (as proposed by different authors). In fact, the context of a sentence is not helpful in these questios since adding more words from it degrades the performance in PMIC for all different window sizes, as shown in figures 5.10 and 5.16. While the differences are not significant in TS1, they are for TS2. Using all words except stopwords, the result from PMIC is better than any other contextual measure—76% correct answers in TS1 (with PMIC and a window size of 8). In TS2, CP is better than PMIC when all the words from context are used.

The results for TS1 and TS2 suggest that the available context is not very useful or that it is not being used properly. It is possible that using other lexical units from the context and not occuring in the given sentence could be helpful in decididing which alternative synonym is the best for the given TW. However, it may be possible to increase the performance of the other

contextual measures by using less context.

The context provided by a sentence is not the same context used in other methods, in particular Latent Semantic Analysis (LSA) [60] and Hyperspace Analogue to Language (HAL) [67]. LSA calculates the latent aspects of documents and computes the similarities between words based on those aspects. The latent aspects are other words from documents. In the case of the TOEFL test set questions, the only information is the target word and the set of alternatives. It is not clear what is the best strategy for using the other words from the sentences in TS1 and TS2 when using LSA as the model to answer the questions. In HAL, co-occurrences are measure with a weighted-window estimator[3] and the co-occurrence frequency as the cells of a word-word matrix. For the affinity strength between lexical units a family of functions is proposed. Let b and d two lexical units and V_b and V_d be vectors with co-occurrences of b and d with the remaining lexical units in the matrix. Let $D = V_b - V_d$. The family of functions (or norms) is then $\|D\|_p = \left(\sum_i |D_i|^p \right)^{1/p}$, for which common values for p are $1, 1.5, 2$ [67]. For $p = 1$, this measure is the L_1 norm; for $p = 2$ it is similar to the cosine of pointwise mutual information (CP) and raw frequencies are used. In HAL, such as in LSA, the standard way of using context is to obtain it from the corpus and not from the sentence. However, in the case of HAL, as the result is expected to be similar to L_1 norm and CP, there is no strong reason to pursue this model in our evaluation.

Impact of corpus size

The terabyte corpus is a valuable resource for estimation. It is possible that the same results obtained in the point estimation procedures do not require such a large corpus since most words in the test sets used are common in English texts. We further analyze the test sets with regard to corpus size.

The corpus is distributed in 38 separate databases and we can use any subset to answer the synonym questions. In addition, we split one database into five smaller pieces: 1/3, 1/6, 1/12, 1/24 and 1/48 of the 25 gigabytes contained in that database. We chose measures based on their performance—in at least one condition, the measures used have top absolute performance in the test set.

[3] not normalized

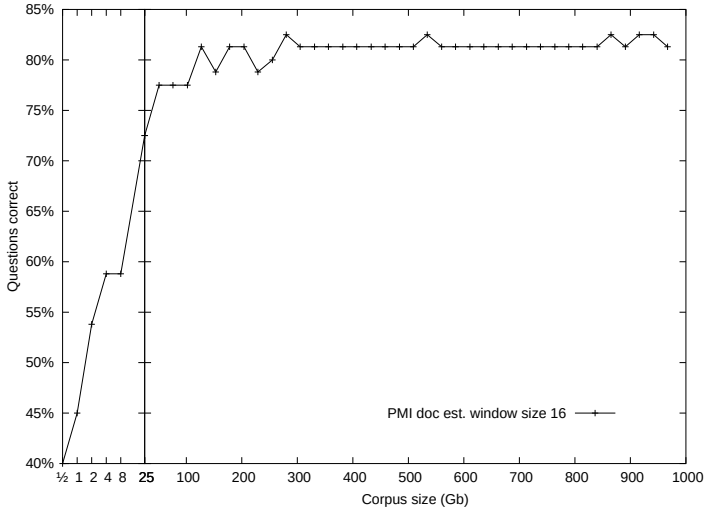


Figure 5.17: Impact of corpus size on TOEFL

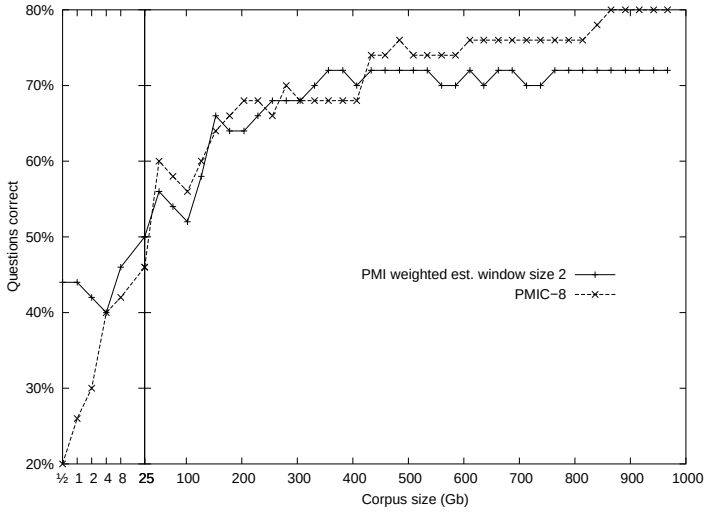


Figure 5.18: Impact of corpus size on TS1

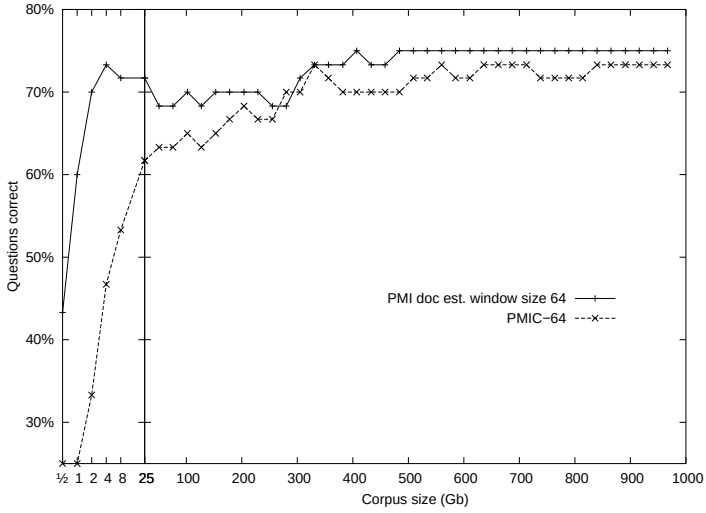


Figure 5.19: Impact of corpus size on TS2

Pair		Pair	
b	*d*	*b*	*d*
afraid	fear	anger	mad
baby	boy	bath	clean
beautiful	ugly	bed	sleep
bible	god	bitter	sweet
black	white	blossom	flower
blue	sky	boy	girl
bread	butter	butter	bread
butterfly	moth	cabbage	head
thief	steal	thirsty	water
tobacco	smoke	trouble	bad
whiskey	drink	whistle	stop

Table 5.7: Association norms examples

The impact of the corpus size is shown in Figures 5.17, 5.18, and 5.19 for TOEFL, TS1 and TS2, respectively. These graphs show that initial performance is very poor but that, at some point, the number of co-occurrences become stable and the results tend to saturate. In no single case was necessary to use more than 500 gigabytes for the best absolute performance. In fact, for TOEFL and TS2 the performance improvement is very small for corpus sizes greater than 50 gigabytes. The reason these tests reach an asymptote is due to the convergence of the estimators to their actual value as the corpus size increases. Some lexical units' estimators will converge faster than others to their real value as the corpus size increases; this is due to the fact that some occur more frequently than others [30].

5.1.2 Skew

Our second evaluation uses the parametric affinity model in a new approach to solve synonym questions. This method is completely new and it is quite different from other methods used for affinity. The parametric model for affinity, the gamma distribution, is a statistical distribution. As with other statistical distributions, the gamma distribution has moments about the mean, from which the third moment is the skew of the distribution. The gamma distribution fits the data by maximum likelihood and from that we can compute the skew. Our hypothesis is that the degree of affinity of two terms is related to the skewness of the fitted model.

Pair Sets	γ
Minnesota association norm	3.1425
Random set	2.1630

Table 5.8: Skewness, $\gamma = 2.0$ indicates independence

Test set	Correct Answers
TS1	76.0%
TS2	75.0%
TOEFL	78.8%

Table 5.9: Skew results on synonym questions

In order to validate our hypothesis that a greater positive skew corresponds to more affinity, we used a list of pairs from word association norms and a list of randomly picked word pairs. Word association is a common test in psychology [76], 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 [50]. Table 5.7 shows some examples. It is composed of 100 stimulus words and the most frequent answer given by 1000 individuals who took the test. The list of randomly picked pairs used as baseline also comprises 100 word pairs, but is generated by randomly choosing words from a small dictionary[4]. The skew in the gamma distribution is $\gamma = 2/\sqrt{\alpha}$ and Table 5.8 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 high skew of the norms in the association norms compared to random pairs is an indication that γ can used directly to identify related words, including synonyms.

In order to estimate α and β we compute the empirical distribution. This distribution provides us with the right-hand side of equation 4.5 and for which α can be solved numerically. The calculation of β is then straightforward. Table 5.9 show the results for the three test sets using the skew as the only information to answer the questions.

Since skew represents the degree of asymmetry of the affinity model, this result suggests that skew and synonymy are strongly related.

This result is not significantly better or worse than the top point estimation results. However, using the skew implies that the user will not have to set or tune any parameter, such as window size, window or document estimator. Thus, it is a good alternative as a measure of affinity.

[4]Linux's /usr/dict/words

Test Set	Correct Answers		
	initial cut-off		
	0	4	7
TS1	76.0%	75.0%	72.0%
TS2	80.0%	71.7%	73.3%
TOEFL	80.0%	86.3%	83.8%

Table 5.10: Results of log-likelihood ratio over intervals in the synonym questions

5.1.3 Log-likelihood Ratio over Intervals

A new method based on the log-likelihood is also used to solve the TOEFL synonym questions. Dunning [36] proposed the use of the log-likelihood ratio as a measure of association strength for sequential bigrams. In the bigram case the estimations are simple since the co-occurrence frequency is the number of times the two words occur in sequence.

We extend the idea of computing the log-likelihood by making use of the affinity models. Instead of using point estimation to determine the co-occurrence frequency, we use the parametric model for independence and the empirical distribution in a log-likelihood ratio. Since the distance between lexical units is important for affinity, we sum the log-likelihood in a interval as follows: for each target-alternative pair, the log-likelihood of the number of co-occurrences for every distance in the range up to a maximum distance of j words apart, giving

$$log\lambda_{b,d} = \sum_{\delta=i..j} log\frac{L(P_\delta(b,d);p_O)}{L(P_\delta(b,d);p_I)} \tag{5.1}$$

where p_O is the empirical or the value from the gamma distribution that fits the data and p_I is the the number of co-occurrences given by the independence model. An initial cut-off i can be used to discard the affinity caused by phrases containing both target and alternative words.

In our experiments the upper cut-off was set to be 750, 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. The results for log-likelihood are shown for different initial cut-offs in Table 5.10. It is interesting to note that the log-likelihood method yields best absolute performance among all the methods presented in this chapter: 86.3% for TOEFL, 78% for TS1 and 80% for TS2.

Method	TOEFL	TS1	TS2	Overall
Leacock & Chodorow	45.0	60.0	46.7	49.5
Jiang & Conrath	41.3	60.0	43.3	46.8
Lesk	85.0	58.0	68.3	72.6
Lin	44.0	40.0	43.3	42.6
Hirst & St-Onge	78.0	62.0	58.3	67.3

Table 5.11: % correct answers using similarity based on WordNet

Unlike the method using the skew, the log-likelihood has user parameters that can affect the outcome. Both initial and upper cut-offs can affect the result but simply using no initial cut-off and an upper cut-off of the average document length appears to be a reasonable choice.

5.1.4 Knowledge-based approach for Semantic Similarity

The final experiment for the synonym questions are performed using WordNet as the source for semantic similarity. The methods presented in Section 2.3 are used to find the best alternative for the question, where *best* means the alternative that has more similarity, as defined by each individual method, to the target word TW.

Table 5.11 depicts the results of the semantic similarity based on the three test sets. Two methods perform closely to the statistical affinity methods but never outperform them. The best one, *Lesk* modified method, uses context from the knowledge base but ignores the sentence given in the case of TS1 and TS2.

These methods also suffer from incomplete lexicon information. Although WordNet[5] has many entries, it still misses some of the words in the synonym tests; in particular, in the cases where the alternatives or the target word are phrases. The number of look ups in word net is 950 (190 questions times four alternatives plus the target word) for which wordnet has no information for 24.

[5]version 1.7

1. 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*

2. The spellings of many Old English words have been _____ in the living language, although their pronunciations have changed.

a) *preserved*
b) *shortened*
c) *preempted*
d) *revised*
e) *improved*

Figure 5.20: Examples of fill-in-the-blanks questions

5.2 GRE fill-in-the-blanks

5.2.1 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 the log-likelihood ratio test:

$$log\lambda = log\frac{L(P_\Delta(b,d);p_O)}{L(P_\Delta(b,d);p_I)} \tag{5.2}$$

where p_O and p_I are the parameters for $P_\Delta(b,d)$ under the empirical distribution and independence models, respectively. It is also possible to use the cumulative C_Δ instead of P_Δ. Figure 5.21 shows log-likelihood ratios using the asymmetric empirical distribution, and Figure 5.22 depicts log-likelihood ratios 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 5.20.

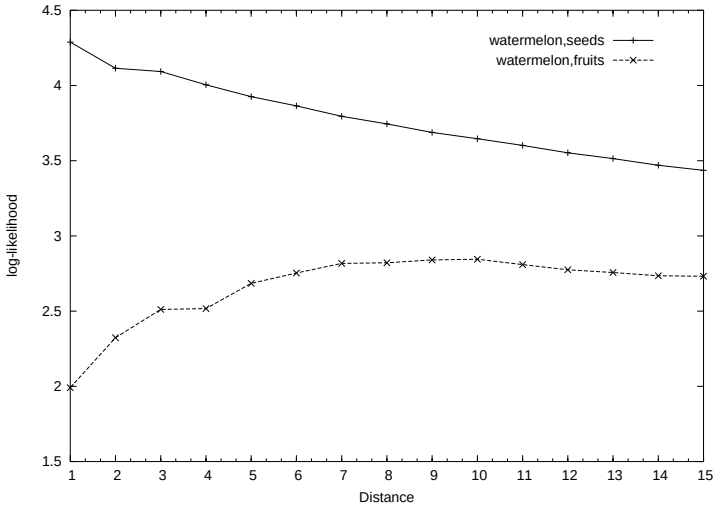


Figure 5.21: Log-likelihood – WATERMELON pairs

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

$$log\lambda = log\frac{\prod_{b\in S}\prod_{d\in S,b\neq d} L(P_{\delta_{b,d}}(b,d);p_O)}{\prod_{b\in S}\prod_{d\in S,b\neq d} L(P_{\delta_{b,d}}(b,d);p_I)} \tag{5.3}$$

where $\delta_{b,d}$ is distance of b and d in the sentence (and $P_{\delta_{b,d}}$ is a short for $P_{\Delta=\delta_{b,d}}$). 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:

$$log\lambda_b = log\frac{\prod_{d\in S,b\neq d} L(P_{\delta_{b,d}}(b,d);p_O)}{\prod_{d\in S,b\neq d} L(P_{\delta_{b,d}}(b,d);p_I)} \tag{5.4}$$

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 surrounding words as context. Second, it causes an indirect effect of smoothing, since we use cumulative probabilities to compute the

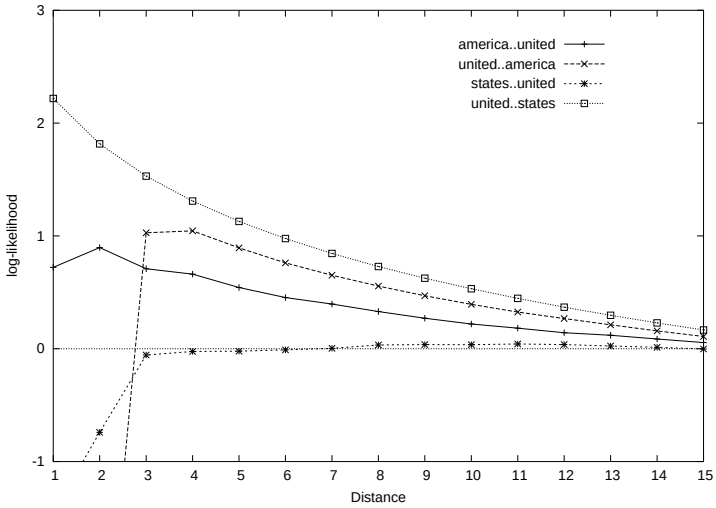


Figure 5.22: Log-likelihood – UNITED pairs

likelihood. As with any distance model, this approach has the drawback of allowing constructions that are not syntactically valid.

Another important issue is the zero-frequency problem. Even using cumulative probabilities as a smoothing effect, there can be cases where the first co-occurrence is observed at a farther distance. In this case, the probability of observing the pair is obviously zero in the maximum likelihood estimator. Many alternatives for this problem exist; Chen [16] gives a survey of smoothing techniques for language modeling and these techniques can be applied in our case. However, the fact that we have a function to compute the expected number, in both independence and affinity models, works as an alternative method for smoothing.

The tests used are from GRE practice tests extracted from the web sites: `gre.org` (9 questions), `PrincetonReview.com` (11 questions), `Syvum.com` (15 questions) and from `Microedu.com` (28 questions). Table 5.12 shows the results for a cut-off of seven words. Every question has five options, and thus selecting the answer at random gives an expected score of 20%. Our framework answers 55% of the questions, that even with the limited number of questions, is substantial improvement over the baseline.

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

Table 5.12: Fill-in-the-blanks results

5.3 Summary

For the synonym questions, the point estimation and the affinity models perform similarly. Some practical considerations are in order. First, affinity based on point estimation is easier to compute in the sense that the co-occurrence is measured in only one window size. In contrast, the affinity models require the whole distribution to be computed and, in the case of the affinity model, calculating the two parameters of the gamma distribution is more expensive. If computational burden is a problem then point estimation is more adequate. For point estimation, based on the synonym questions, PMI and window size of 16–32 is likely to perform consistently in any estimator. The weighted-estimator has more chance to produce statistically significant results than simple estimator; the document estimator is also a good alternative for PMI.

On the other hand, the skew as a measure of affinity is the only one that does not require the user to choose an arbitrary parameter: the window size. The performance of the skew is not worse than any other method.

Using the log-likelihood ratio of intervals between the affinity and independence models results in the best absolute performance. The difference for the skew is not statistically significant but this method provides the best results for the three test sets individually. It is possible that the differences might become statistically significant if the test set were bigger.

The use of the affinity models has the additional benefit of creating a function for smoothing; the same cannot be said for point estimation where the smoothing occurs in a heuristic way.

The context sentence as provided in two test sets, TS1 and TS2, did not help improve the results on those sets. In some cases, the results are similar but for most methods that take advantage of context this was not the case.

The results on fill-in-the-blanks questions show that the parametric models of affinity can be

applied to problems where *n-gram* models are normally used, such as predicting the next word given a history, with reasonable performance.

Chapter 6

Scoring Missing Terms in IR

In this Chapter we examine the application of lexical affinity using a large corpus to two information retrieval problems: document retrieval and passage retrieval. We propose a new method to replace missing query terms when scoring documents and passages. We assess the improvements of our method using standard evaluation suites for *ad hoc* document retrieval and question answering. Our replacement method modifies two well-known scoring functions: the Okapi BM25 formula for document retrieval [51]; and MultiText's passage retrieval formula [21, 23].

6.1 Missing Term problem

A user query for a retrieval system expresses both the user's information need and the knowledge he/she has about the query topic. All of these surrounding factors in an information retrieval setting make it hard to capture other aspects of a query, such as topic, specificity, and genre, among others. In particular, a word used in a query can have different meanings or have other words that may replace it in documents. This causes problems such as query drift and mismatching vocabulary that deteriorate the accuracy of the retrieval process. One way to address the mismatching problem is through automatic query expansion (AQE), where new terms are added to create a new expanded query to be submitted to the retrieval engine [13, 14, 51, 88, 111, 115]. On the other hand, AQE increases the chance of query drift [29, 72].

#	Original query	Replacement	Document
51	airbus subsidies	aircraft subsidies	SJMN91-06350209
58	rail strikes	railroad strikes	SJMN91-06339333
59	weather related fatalities	weather related casualties	SJMN91-06017078
59	weather related fatalities	weather related deaths	SJMN91-06017055
70	surrogate motherhood	surrogate pregnancy	SJMN91-06338198
72	demographic shifts u.s.	population shifts u.s.	SJMN91-06363136

Table 6.1: Replacements examples

An alternative for handling the problem of mismatching vocabularies is the use of translation language models and methods from cross-language information retrieval (CLIR). The lack of query terms in a document is addressed by using one or more words in the document as a translation for the missing term [6, 32, 44, 92]. In a sense, the translation of document words into query terms is not the same thing as expanding the query with extra terms. Translation focuses on replacing query terms while the AQE focus is on complementing the query with some other aspects.

We take a different approach to address the mismatching vocabulary problem. Unlike AQE and translation models, instead of augmenting the query to score documents, we use the original query and replace missing terms only when necessary. The idea is to use the original query terms to score documents whenever possible. This can be viewed as a type of translation, however we do not try to translate query terms that are present in the document. Depending on how we choose the replacement terms, we can also capture relationship types other than word translation.

Since the vocabulary changes from one document to another, it is likely that our approach will score different documents using different queries but forming each new query with minimal changes to the original user query. Table 6.1 shows some examples of the queries and documents that partially match them; the replacement is chosen from the same document. In the same situation, traditional AQE will use one query for all documents, regardless of the mismatching vocabulary problem. In order to prevent the original query terms from being outweighed by replacement terms, we adjust the weights of replacement terms based on their affinity with the missing query term. While our approach is a form of query expansion, it does not exclude the possibility that a traditional AQE could be performed later in the retrieval process.

6.2 Related Work

Information Retrieval models, with the exception of certain queries of the classic boolean model, allow documents to be scored when not all of the query terms are present. In general, the score of a document is given by weights assigned to the query terms present in it. In the vector space model [90] a missing term will have zero value in the document vector, thus contributing no weight towards the document's score. In the *tf.idf* probabilistic models [51], a missing term will not count either since its term frequency is zero. In these models a common approach to handle mismatched vocabulary is to use pseudo-relevance feedback [13, 14, 51, 88, 111].

In CLIR, the query is specified in one language and the documents in another. As a consequence, the query terms will not usually occur in the documents. To address the language barrier, a common approach is to translate the query into the document language [92]. Darwish and Oard use the idea of replacement of query terms by document words at query-time in CLIR and in the retrieval of scanned OCR documents [32]. In their CLIR application a number of translation resources, such as dictionaries and parallel corpora, are used. A parallel corpus was used in their OCR application, having on one side the corrected digital version of the document and on the other the version resulting from OCR (containing errors). These translation resources are then used in a document retrieval task.

In language models, instead of using maximum likelihood estimators, the term frequencies are smoothed in order to assign some probability mass for missing terms in all documents [83]. Pseudo-relevance feedback is also used in language modeling [61, 115], normally by expanding the query term set to form a query language model.

One particular language model [6], the statistical translation model for IR, is related to the work presented here. It is inspired by statistical translation models for natural language and relies on the idea of parallel corpora, where there exists an alignment between texts written in different languages. When these models are adapted to IR, a translation is made from a document to a query and the retrieval process comprises word translations from document into query terms by means of translation probabilities. The relevance of a document is assumed to be monotonically increasing with the likelihood of generating (translating) the query from the document. The translation probabilities enable the use of all query terms for every document, even when they

are not present in the document. Berger *et al.* propose two translation models for Information Retrieval [6], and both models (1 and 1′) compute the weight for every query term as the sum of the product of the translations of every document word into the query term and document frequency. The use of all words as a possible translation of a query term is a way to capture all possible alignments between the document and the query. This also has the effect of relating all terms in the query and document, even when they do not have any affinity. It is also interesting to note that the queries are expanded to form a query model before the actual "translation" (i.e. scoring) occurs, which can lead to query drift.

In monolingual information retrieval the idea of translation is not natural. It may be arguable that one synonym may translate to its counterparts; however, that is not the idea behind AQE. Rather, in AQE, the expansion terms tend to complement the original query terms by including not only synonyms but also other types of relationships, such as morphological variants of the term, and also other semantic relations (e.g. hyponyms and hypernyms). Furthermore, the translation models rely either on the availability of alignments, such as in CLIR, or on brute force alignments, such as the statistical translation model for IR.

6.3 Modified Retrieval Methods

Two probabilistic models, one for passage retrieval and one for document retrieval, are modified in order to accommodate non-zero scoring of missing terms. In this method the goal is to make as few changes as possible in order to prevent query drift.

6.3.1 Passage Retrieval

We use the passage retrieval component of MultiText. It has been successfully applied to question answering [23, 25, 66] and pseudo-relevance feedback [113]. From a query $Q = \{t_1, t_2, .., t_k\}$ let $T \subseteq Q$. Given an extent of text comprising all words in the interval (u, v) with length $l = v - u + 1$, the probability $S(t, l)$ that the extent contains one or more occurrences of t is

$$\begin{aligned} S(t,l) &= 1 - [1 - P(t)]^l \\ &= 1 - [1 - lP(t) + O(P(t)^2)] \\ &\approx lP(t). \end{aligned} \tag{6.1}$$

The probability that an extent (u, v) contains all the terms from T is then

$$\begin{aligned} S(T,l) &= \textstyle\prod_{t\in T} S(t,l) \\ &\approx \textstyle\prod_{t\in T} lP(t) \\ &= l^{|T|} \textstyle\prod_{t\in T} P(t). \end{aligned} \tag{6.2}$$

The estimation of $P(t)$ is given by the Maximum Likelihood Estimator (MLE) for t in the collection

$$\hat{P}(t) = \frac{f(t)}{N} \tag{6.3}$$

where $f(t)$ is the collection frequency of t and N is the corpus size in words. The score for an extent of length l containing the terms in Q is the self-information of $S(T, l)$

$$\sum_{t\in T} \log\left(\frac{N}{f(t)}\right) - |T| \log(l) \tag{6.4}$$

The score is higher for short passages containing all terms in T and there is a trade-off between the number of terms and size of the passage.

For the original passage retrieval method presented by Clarke *et al.* [23], an efficient algorithm to retrieve all passages comprising 1 to $|Q|$ query terms is presented by Clarke [21]. The running time to extract all extents of size $|T|$ is $O(|Q|\mathcal{J}_l log(N))$ where $|Q|$ is the total number of query terms, $\mathcal{J}_l$ is the number of extents containing $|T|$ query terms and N is the corpus size. The algorithm is based on the positions of query terms, checking for close occurrence of other query terms and skipping repetitions of the same term. This algorithm benefits from the sorted position entries in the inverted list used to index the underlying collection and quickly locate terms.

To accommodate scoring of missing terms, the modified version only considers the whole query Q since every extent has a representative for missing query terms. We assume $P(t, t) = P(t)$ if the

term t is present in the extent. If the query term t is not in the extent, a replacement term r will be chosen in the extent. The weight of the replacement is the conditional probability $P(t|r)$, which is calculated by estimating the maximum likelihood for $P(r)$ from the corpus and estimating the joint probability by

$$\hat{P}(t,r) = \frac{f(t,r)}{N_{joint}} \tag{6.5}$$

where $f(t,r)$ is the joint frequency and N_{joint} is the total number of pairs considered for the joint frequency in the corpus. This is the same notation as in Chapter 3.

We take a winner-takes-it-all approach and choose the best r in the extent,

$$\arg\max_{r \in (u,v)} \hat{P}(t|r) \tag{6.6}$$

Finally, the modified version of equation 6.4 using replacements is given by

$$\sum_{t_i \in Q} \log\left[\frac{N}{f(t_i)} \cdot \hat{P}(t_i|r)\right] - |Q| \log(l) \tag{6.7}$$

We should note that since every non-empty extent has a representative for a query term, we can make arbitrary decisions on the extent size. This creates a trade-off between extent size and replacement quality. On the other hand, the fact that any extent can have a representative does not allow us to use the efficient algorithm used in the original method. Instead of selecting the extent in sub-linear time complexity (log of the corpus size) as in the original method, our approximation extracts the passages in linear time.

The implementation of the replacement method does not look for passages in the corpus directly, as the original method does. Instead, a subset of the documents in the collection is used to find the passages, reducing the search space. Since the algorithm runs in linear time, this restriction makes the replacement method feasible. For every document, $\hat{P}(t|q_i)$ is calculated for every pair containing a word t from the document and query term q_i (i.e., the algorithm runs in $O(|Q| \cdot N)$ and we heuristically reduce the size of N by selecting documents that will contain, potentially, good extents). The resulting data is scanned to find and score extents. A sliding

window is used to keep track of the query term representatives in it.

6.3.2 Document Retrieval

For document retrieval, we use Okapi BM25 formula [51], a *tf.idf* model that uses the bag-of-words approach. In this approach, the order or relationship between the query terms is ignored. The weights of query terms are calculated from the collection, and relevancy is used if available. A document's score is the sum of weights of query terms in that document and taking into account the in-document frequency of these terms. Specifically, given an query $Q = \{t_1, t_2, .., t_k\}$, a document d is assigned the score

$$\sum_{t_i \in Q'} w^{(1)} \frac{(k_1+1)d_{ti}}{K+d_{ti}} \frac{(k_3+1)q_{ti}}{k_3+q_{ti}} + k_2 \cdot |Q| \cdot \frac{avdl - dl}{avdl + dl}, \tag{6.8}$$

where

$$\begin{aligned}
w^{(1)} &= \log \frac{(r_{ti}+0.5)/(R-r_{ti}+0.5)}{(d_{ti}-r_{ti}+0.5)/(D-d_{ti}-R+r_{ti}+0.5)} \\
Q' &= \text{subset of unique terms in } Q \\
D &= \text{number of documents in the collection} \\
d_{ti} &= \#\text{ documents containing the term } t_i \\
q_{ti} &= \text{frequency of } t_i \text{ in the query } Q \\
d_{ti} &= \text{frequency of } t_i \text{ in the document } d \\
dl &= \text{document length in words} \\
avdl &= \text{average document length in the collection} \\
R &= \#\text{ relevant documents for the query} \\
r_{ti} &= \#\text{ relevant documents containing } t_i \\
K &= k_1((1-b) + b \cdot dl/avgdl) \\
k_1, b, k_2, k_3 &= \text{query nature and database parameters}
\end{aligned}$$

In cases where relevance information is not available, the values of R and r_{ti} are set to zero.

Usual values for query nature and database parameters are $k_1 = 1.2$, $b = 0.75$, $k_2 = 0$, and $k_3 = \infty$, as result the main *tf.idf* components are kept in the short version of the formula:

$$\sum_{t_i \in Q'} \log \frac{D - d_{ti}}{d_{ti}} \cdot q_{ti} \cdot \frac{(k_1 + 1)d_{ti}}{K + d_{ti}} \tag{6.9}$$

To allow missing query terms to be scored we modified the short formula (equation 6.9) by adding the relatedness factor for term r as a replacement for term t_i in similar fashion to the approach taken for passage retrieval. We calculate the conditional $P(t_i|r)$ by the maximum likelihood of $P(r)$ and the joint probability :

$$\hat{P}(t_i, r) = \frac{f(t_i, r)}{N_{joint}} \tag{6.10}$$

where $f(t_i, r)$ is the joint frequency and N_{joint} is the total number of pairs considered for the joint frequency in the corpus.

As in the modified passage retrieval method, we use the replacement r that satisfies

$$\arg\max_{r \in d_m} \hat{P}(t_i|r) \tag{6.11}$$

where d_m is a document that does not contain t_i.

Our modified version of BM25 uses a modified *idf*,

$$\sum_{t_i \in Q'} \log\left[\frac{D}{d_{ti}} \cdot \hat{P}(t_i|r)\right] \cdot q_{ti} \cdot \frac{(k_1 + 1)d_{ti}}{K + d_{ti}} \tag{6.12}$$

Equation 6.12 is similar to the modified *tf.idf* presented by Darwish and Oard [32] and used in CLIR and OCR retrieval:

$$tf_i = \sum_{k \in R(t_i)} tf_k \cdot w_k \tag{6.13}$$

and

$$idf_i = 1/\sum_{k \in R(t_i)} d_k \cdot w_k \tag{6.14}$$

where tf_i and tf_k are frequencies of terms i and k in the document being scored, w_k is the replacement weight of the term k and $R(t_i)$ is the set of replacements for t_i. There some other major differences between our modified BM25 and Darwish and Oard's formula. First, they recommend the use of the replacement weight twice, once in the *tf* component and another in the *idf*. The way the replacements are computed also differs from our method, which is explained in section 6.4. A last major difference is the fact that the original terms are not present in the scored documents in both CLIR and OCR; thus they do not need to handle the case when the query term is present.

6.4 Finding Term Replacements

To prevent query drift, it is desirable to have a replacement term that represents the original term's abstract concept when used in the context specified by the user query. The actual type of semantic relationship is not easily predicted; it can be just a synonym or a hypernym, or it can have any other relationship with the original query term. We use the lexical affinity approach to find replacements.

In particular, we use the point estimation described in Chapter 3. Since the number of pairs we have to score for both document retrieval and passage retrieval is high. The pointwise mutual information (PMI) is used as the similarity measure to score relatedness within any pair of terms b and d:

$$PMI(b,d) = log\frac{P(b,d)}{P(b)P(d)} \tag{6.15}$$

The reason for choosing PMI is twofold. First, it was demonstrated to be effective for language phenomena, as described in Chapter 5. Second, it has a relationship with *idf*. This relationship

comes from the assumption that $P(b,b) = P(b)$, thus

$$\begin{aligned} PMI(b,b) &= log\ \frac{P(b,b)}{P(b)\cdot P(b)} \\ &= log\frac{P(b)}{P(b)\cdot P(b)} \\ &= -log\ P(b) \\ &= idf_b \end{aligned}$$

In the case of the pair of words b and d, the maximum value for the pointwise mutual information is bounded by $PMI(b,d) \leq idf_b$ and $PMI(b,d) \leq idf_b$. This can be easily verified since the PMI formula has maximum value when the joint probability is equal to the smallest marginal (if marginals are different). Therefore, we can use *idf* to normalize the PMI for a given word we want to replace

$$CondPMI(b,d) = \frac{\log\ P(b,d)/[P(b)\cdot P(d)]}{\log\ 1/P(b)}, \tag{6.16}$$

which produces the same ranking that would be generated by

$$\frac{P(b,d)/[P(b)\cdot P(d)]}{1/P(b)} = P(b|d) \tag{6.17}$$

Thus, if we fix one word, in this case the missing query term, we can rank its affinity with the remaining words of the vocabulary. Since the goal is to find a replacement for one query term at a time, the denominator of the equation 6.16 is fixed for every missing term. We should note that there is a problem with the normalization in the conditional PMI. The problem occurs when PMI is negative, in which case we just set it to zero. Setting the negative value to zero could be avoided if we offset both *idf* and PMI by the minimal PMI value. We ignore pairs of terms with negative PMI, thus we use a self-regulated cut-off for the minimal value for a conditional PMI. We assume that any word in the document with a negative PMI with respect to the missing query term is not a good candidate for replacement.

The estimation for $P(b,d)$ uses the weighted-window estimator (section 3.1) with distances ranging from four to 40 words apart. The lower cut-off prevents phrasal relationships, as described

Word	Frequencies
New	104,483,262
York	12,205,261
population	4,854,401
demographic	428,641

Table 6.2: Corpus Individual Frequencies

Pair		Distance range		
b	d	1–3	4–40	41–∞
NEW	YORK	11,784,589	3,365,934	8,215,334
POPULATION	DEMOGRAPHIC	10,509	89,772	485,491

Table 6.3: Corpus Frequencies of Pairs at specific distance intervals

in Chapter 5. For example, if the term "New" is a query term but "York" is not, then the latter is probably not a good replacement for the first. As most of the co-occurrences of "New" and "York" happen at distance one, this cut-off will avoid this bias for pairs in the same phrase. The frequencies values for "New" and "York" and for the pair "demographic" and "population" are shown in the tables 6.2 and 6.3 over a terabyte corpus. The pairs counting in table 6.3 do not include nesting, thus "new New York" will count only once towards the joint frequency. As seen in the results of point estimation experiments in Chapter 5, when pointwise mutual information is used a window size of around 32 words is a good setting for an upper bound on the distance. This was also pointed out in earlier studies of these frequency estimators by Terra and Clarke [98].

6.5 Empirical Evaluation

6.5.1 Methodology

A standard evaluation in information retrieval has been held by the National Institute of Standards and Technology (NIST) in the context of the Text Retrieval Conference (TREC) since 1992. In TREC many different types of retrieval have been evaluated over the years, which started with an *ad hoc* document retrieval task and has, since then, evaluated cross-language retrieval, on-line retrieval (filtering), retrieval in hypertext collections (web), interactive retrieval, and question answering among others. Every task evaluation in TREC starts by the creation information

```
<head> Tipster Topic Description
<num> Number:  051
<dom> Domain:  International Economics
<title> Topic:  Airbus Subsidies
<desc> Description:
Document will discuss government assistance to Airbus Industrie, or
mention a trade dispute between Airbus and a U.S. aircraft producer
over the issue of subsidies.
```

Figure 6.1: *ad hoc* topic

```
<num> Number:  201
<desc> Description:
What was the name of the first Russian astronaut to do a spacewalk?

<num> Number:  1397
<desc> Description:
What was the largest crowd to ever come see Michael Jordan?
```

Figure 6.2: Question answering topics

needs. Each information need is called a *topic* in TREC. Figures 6.1 and 6.2 show TREC topic examples for *ad hoc* and question answering tasks, respectively. The topics may contain different fields depending on the task (and year). For *ad hoc*, the title field and/or description fields are normally used. The QA topic is normally composed of the question itself. For each topic, the systems retrieve objects based on their specific rules and methods, and submit these results for assessments. The judgments are then performed by a group of human assessors. Among other things, the results of TREC are evaluation suites for different retrieval tasks. NIST also supplies the corpora for the tasks.

To evaluate the new passage retrieval method we use the question answering (QA) evaluation suite from TREC, which started in eighth edition of the conference in 1999 and is still running to this day.

The new document retrieval method is evaluated using *ad hoc* retrieval from TREC topics 51–100.

For both the passage and document retrieval experiments, all of the replacements were calculated using the statistics of the terabyte corpus described in Section 2.4.1.

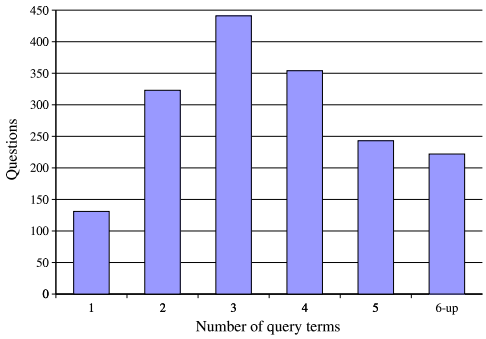


Figure 6.3: QA query terms histogram

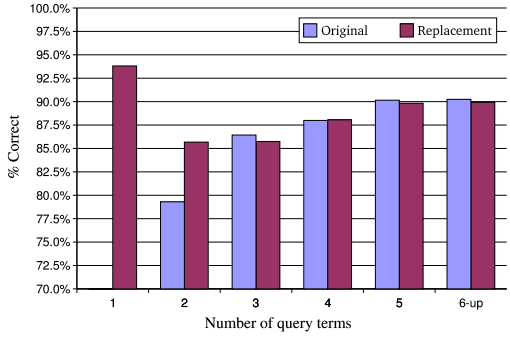


Figure 6.4: Passages correct

6.5.2 Passage Retrieval

We assess the performance of the modified passage retrieval method using QA test sets from TREC 9 through 12. TREC 9 contains some question variants, with some rewording of questions. Those questions are left out because the important query terms are the same and, as such, they would not add new information to this evaluation. The remaining 1,732 questions with known answers in the TREC official collections were used. As we are particularly interested in the evaluation of the passage retrieval method, we only extract passages from documents in TREC collections (TREC disks 4–5 and AQUAINT) that contain answers to questions. A similar approach was used in Tellex *et al.* [97]. For these 1,732 questions, the total number of relevant documents is 10,561.

The queries used in passage retrieval methods were generated from questions by simple stopword exclusion. The query size distribution is given by Figure 6.3. Since many of the queries are short, a missing query term can harm the effectiveness of the passage retrieval. We perform automatic judgments in this evaluation, using the regular expression patterns available from the NIST web site for TREC[1]. We consider a passage correct if it matches the pattern for the question.

For each pair <query number, relevant document> we find the best passages using the original and the modified methods. For the modified method, we scan the whole document to find the best scoring passage among all possible candidates using equation 6.7. For every candidate passage

[1] trec.nist.gov

Method	Coverage	% Correct
TREC10 - Original	96.1%	87.97%
TREC10 - Replacem.	96.1%	89.21%
TRECs 9-12 (1+ missing) - Original	89.9%	85.88%
TRECs 9-12 (1+ missing) - Replacem.	94.5%	87.88%
TRECs 9-12 (all) - Original	94.5%	89.59%
TRECs 9-12 (all) - Replacem.	95.3%	89.50%

Table 6.4: Replacement Method results

Method	Coverage
IBM	92.9%
SiteQ	92.6%
ISI	91.4%
Alicante	91.0%
MultiText	89.8%

Table 6.5: Top five passage retrieval in Tellex *et al.*

we want a representative for each query term to be present. The number of candidate passages is $O(|DL_i|^2)$ for each document, where DL_i is the number of the words in the i-th document. Since the goal of passage retrieval is to find a fragment of text smaller than the whole document, we limit our reported passages to 170 words for comparison purposes. Tellex *et al.* [97] used snippets of 1000 bytes in a similar passage retrieval evaluation (170 words $\sim$ 1000 bytes using our tokenizer). Every 170-word passage has a smaller fragment we call a "hotspot", that contains all the query term representatives; we seek representatives in hotspots of 20 words using a sliding window. Limiting the size of the hotspot is necessary to prevent representatives from being located too far apart, preventing weaker representatives from being used even if they are close to other query terms. This makes the number of passages $O(|DL_i|)$, but we may discard some passages that would have a better score if we considered a larger window. The best hotspot in the document is later extended to 170 words. The choice of hotspot size is a trade-off between execution time and effectiveness.

The baseline is the original passage retrieval method using the scoring function from equation 6.4. To evaluate the difference between the two methods, we first compute the effectiveness measures when at least one of the query terms is missing in the passage retrieved using the original method. Since we retrieve exactly one passage from each document, we can compare the passages

from the two methods side by side. Figure 6.4 plots the percent of correct passages against the number of original query terms. It shows only passages where at least one original term is missing. The y-axis is the percent of *correct* passages, i.e. containing the answer for the question. For instance, for the more than 300 questions that have query size of 2, the original method retrieves 79% of passages correctly (in this case the passages contain exactly one query term). The modified method replaces the missing term with another in the document and improves the percent of correct passages to 86%.

The improvements are higher for short queries, comprised of one or two query terms. For queries of size one, a missing term means no information is available to select a passage in the original method; in this case our new method of replacement can only improve. When more query terms are available, replacements do not help or harm (i.e. the differences are not significant). The difference in the percentage of correct passages, when one or more query terms are not present, is significant at 99% confidence level using the Wilcoxon signed rank test.

We also calculated the *coverage*, the percentage of the 1,732 questions where at least one retrieved passage contained the answer [27]. As many QA systems use the output of the passage retrieval as the input to an answer extraction component, it is important to have at least one passage containing the answer so that upstream components of the system can have a chance to find it.

The new method provides better coverage than the original baseline method. Table 6.4 shows the results of the two methods. In the whole test set, TRECs 9-12, the coverage is a little better in the replacement method. The difference is greater if we compare only passages where all the query terms are not present.

We further compare the results of our new method with the evaluation presented by Tellex *et al.* [97], where different passage retrieval methods were evaluated using the TREC 10 questions. Tellex *et al.* report effectiveness by means of Mean Reciprocal Rank (MRR) and the percent of incorrect *questions* (instead of passages). The MRR is calculated by averaging the inverse rank of the first correct answer to each question. It is not clear that MRR is appropriate for evaluating the passage retrieval component of a QA system. It is an intuitive measure if considered in terms of the end-user. Instead, the passages are going to be further processed by an answer extraction

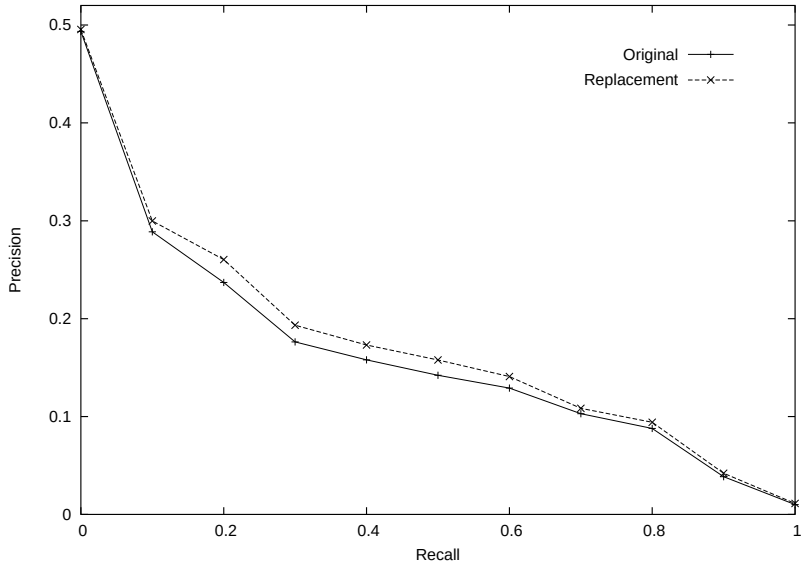


Figure 6.5: Interpolated Precision-Recall for topics 51-100 on SJMN

component, thus their retrieval rank may not be as important as it would be for the end-user. For this reason, we do not report MRR. The latter measure, percent of incorrect questions, is the complement of *coverage* (i.e. 1-*coverage*), thus the results are directly comparable. The reported coverage by Tellex *et al.* [97] is reproduced in Table 6.5. The coverage is higher in our experiments and the differences can be explained by two factors: Tellex *et al.* use *idf* in equation 6.4, which is not appropriate since in its derivation the collection frequency is used (rather than document frequency); the statistics used in both original and modified passage retrieval, and reported in Table 6.4, are drawn from the terabyte corpus and not from TREC collections.

6.5.3 Document Retrieval

For document retrieval, our evaluation was performed on the *ad hoc* queries corresponding to TREC topics 51–100. The target corpus was the San Jose Mercury News sub-collection of TIPSTER/TREC disk 3, containing 90,257 documents. The queries were extracted from the title field, stopwords removed and stemming was not used.

As the retrieval models score only documents containing at least one of the query terms (original+expanded), the number of documents that can be scored is normally smaller a subset

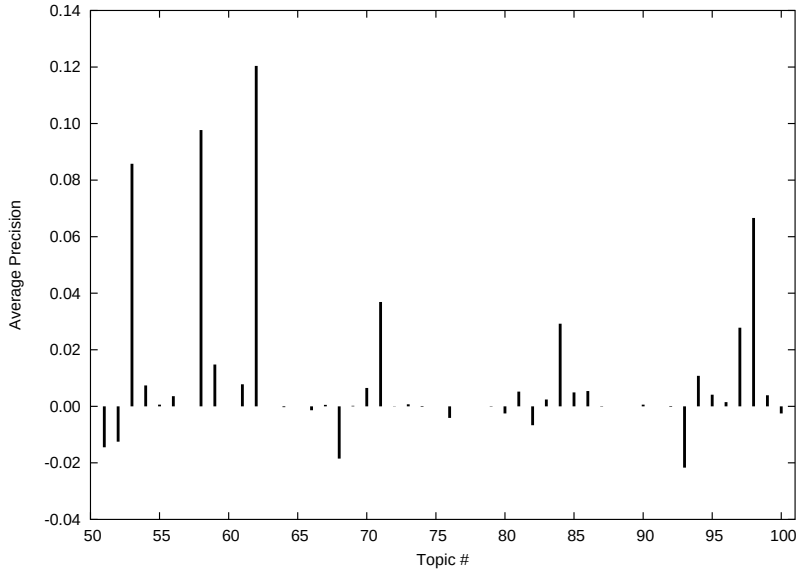


Figure 6.6: Difference in average precision per topic

of all documents in the collection, since documents containing no query term will have zero score. For the case where query terms can be replaced, this limitation is obviously not present; however, it is unlikely that all query terms will need to be replaced. The relevance judgments available for the topics used in this evaluation tend to favor documents with original query terms, since many runs in TREC use original query terms in all runs, and occasionally, expanded terms. Even when expanded terms are used, their weights are usually reduced relative to the original terms. An exception can be found in Smeaton *et al.* [95] in TREC-4: "When the query is expanded we then delete all the original query terms in order to add to the judged pool documents that our expansion would find that would not have been found by other retrieval." For this reason, our evaluation uses documents that contain at least one query term. As a result, four topics (57, 75, 77 and 78) were discarded from our evaluation since they always have exactly one word in the title field; thus our method will score documents the same way the original method does. Two other topics - 65 and 88 - were not considered since they do not have any document judged relevant in the SJMN sub-collection. The remaining 44 topics were used in our evaluation.

For each document, every original query term is weighted as in the normal BM25 formula. If the query term is not present, all the words in the document are considered for replacement and

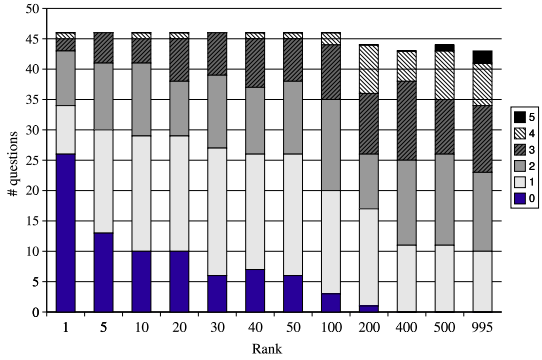


Figure 6.7: Rank by # missing terms - original

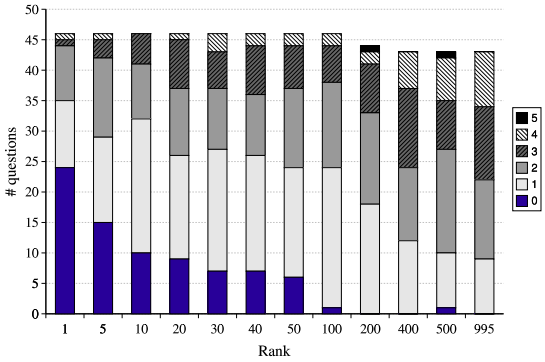


Figure 6.8: Rank by # missing terms - replacement

the corresponding weight is calculated by using equation 6.16. The best replacement is selected for each missing query term and final document score is given by equation 6.12.

Tables 6.6 to 6.11 show some examples of replacements for topic 51 ("Airbus Subsidies"); topic 53 ("Leveraged Buyouts"); topic 62 ("Coup d'Etat"); topic 68 ("Health Hazards from Fine-Diameter Fibers"); topic 71 ("Border Incursions"); and topic 94 ("Computer-aided Crime"). Some replacements are morphological variants of the original term, but other semantic relationships are present as well. For topic 94, no relevant document had the query term *computer* replaced. The representative terms for query term *aided* were not as good as the ones used for the original term *crime*. Replacement in topic 62 tend to focus on the people involved in a specific *coup d'etat*, and as in topic 94 one term is always present in the relevant judgments - *Military*. This shows that some query terms are really important in the query and documents not containing them are unlikely to be relevant.

The precision-recall curves of the original and the modified formula with replacements are depicted in figure 6.5. There is a consistent improvement over the original BM25 and the difference in the mean average precision between the original and the modified methods is statistically significant at 99% level using the Wilcoxon signed rank test. The analysis of the average precision in the individual topics, depicted in Figure 6.6, shows that for most topics the precision improved substantially. In fact, 28 out of the 44 topics improved on average 0.0206, four stayed the same and in 12 topics where the precision dropped the reduction was on average 0.0058.

Table 6.6: Replacements in TREC topic 51

Query Term	Replacement	COND-PMI
Airbus	aeroflot	0.2060
Airbus	mcdonnell	0.1680
Airbus	aerospace	0.1234
subsidies	subsidized	0.1808
subsidies	revamping	0.1135
subsidies	taxpayers	0.0444

Table 6.7: Replacements in TREC topic 53

Query Term	Replacement	COND-PMI
buyouts	buyout	0.3224
buyouts	divestitures	0.1862
buyouts	mergers	0.1756
leveraged	buyout	0.1667
leveraged	takeovers	0.1587
leveraged	mergers	0.1050

It is interesting to note that the recall in the replacement method improved as well, from 1227 to 1315 relevant documents retrieved, which corresponds to retrieving 8.70% of the remaining relevant documents not retrieved by the original Okapi BM25 (at 1000 documents). A run with all terms stemmed also improved mean average precision but maintained the recall at exactly the same level as the original method.

We performed a failure analysis on the four topics responsible for the big drops in average precision: 51, 52, 68 and 93. In two of them, topics 52 ("South African Sanctions"); and 93 ("What Backing Does the National Rifle Association Have?"), the replacement of components of a phrase were responsible for the decline in performance. This problem can be addressed by using the noun phrases from the query; however, as we will see in section 6.6, using noun phrases does not always lead to improvement. The use of proper noun phrases may be a more viable alternative. In topic 51 ("Airbus Subsidies"), the replacements for the proper name Airbus harmed the average precision. In topic 68 ("Health Hazards from Fine-Diameter Fibers"), the replacements for Fine-Diameter were not helpful, whereas Fibers and Hazards had good replacements in asbestosis and carcinogenicity.

An alternative way to see the differences between the original and the method with replace-

Table 6.8: Replacements in TREC topic 62

Query Term	Replacement	COND-PMI
coups	coup	0.1598
coups	dessalines	0.1721
coups	honasan	0.2376
etat	choonhavan	0.0889
etat	gqozo	0.1251
etat	aristide	0.0273

Table 6.9: Replacements in TREC topic 68

Query Term	Replacement	COND-PMI
hazards	carcinogenicity	0.5195
hazards	hazardous	0.0913
diameter	vesicle	0.4657
diameter	pipe	0.0218
fine	allo	0.4109
fibers	asbestosis	0.1773

ments is to look at their rankings. Figure 6.7 plots different rank positions in the original Okapi BM25 method, and Figure 6.8 shows the same cut points in the new method with replacements. The cumulative bars indicate how many missing query terms the documents ranked at that position have. For example, in the original method, 26 topics had documents with no missing query terms at rank 1. In the replacement method, this number is reduced to 24. The new method of replacement shuffles the ranking since every document has its own query term representative, and there is a slight tendency for documents not containing all of the query terms to move up in the rank. This effect is not stronger because we consider the original query terms to be more important. Nevertheless, we can see that the number of queries ranking documents with no missing query term at position one is reduced between the two methods. We also see a document with all query terms being ranked at position 500 by the new method, whereas in the original Okapi method the same does not occur.

Table 6.10: Replacements in TREC topic 71

Query Term	Replacement	COND-PMI
incursions	gissin	0.2554
incursions	infiltrations	0.2042
incursions	incursion	0.1735
incursions	militants	0.1122
incursions	militia	0.0662
border	Mexicans	0.0032

Table 6.11: Replacements in TREC topic 94

Query Term	Replacement	COND-PMI
aided	conspired	0.0867
aided	autocad	0.1150
aided	drafting	0.1294
crime	hacking	0.0443
crime	crimes	0.1256
crime	burglaries	0.1530

6.6 Replacement Method and Query Formulation Strategies

The replacement method used in both document and passage retrieval can be seen as a query expansion, but not as pseudo relevance feedback. In other methods new terms are normally added from dictionaries or thesauri in a manual or heuristic procedure. In this section we compare the replacement method for passage retrieval applied to QA against other query formulation strategies that can be used in QA.

We use some standard query formulation strategies to compare against the replacement method. For all of them we perform stopword exclusion:

- Bag-of-Words

  This is probably the simplest way to specify a query. In particular this method is preferred when the retrieval is the vector space, probabilistic or language model. The query comprises the question terms, and the order in which terms are specified is not important.

- Stemming

A common strategy in information retrieval is to apply a stemmer in the query terms. The intuition is that by using the stemmed form, and not the lemma, the mismatching vocabulary problem will be minimized. The collection index normally contains both the stemmed and lemma forms.

- Boolean conjunction

  In some QA systems, the queries are formed by creating a boolean expression of selected terms [97, 112]. Our boolean queries are formed as a conjunction of the question terms after stopword exclusion.

- Quotes

  For these queries we keep the original question quoted when supplied, e.g., WHAT COUNTRY IS KNOWN AS THE "LAND OF THE RISING SUN?" For the purpose of retrieval, these quotations are treated as phrases and their constituent words may or may not be used in the query other than in the phrasal component. In our experiments, quote components are not added to the query except with the verb expansion. The remaining of the question words (not stopwords) are used as in the bag-of-words approach.

- Quotes plus Noun Phrases

  To further investigate phrases in our passage retrieval method, we explore noun phrases in the questions that are not part of quotes. The words in the questions are tagged using a standard POS tagger and adjacent pairs were concatenated if the sequence matches one of the following : 1) adjective followed by noun; 2) a non-proper noun followed by any noun; 3) foreign word followed by any noun; 4) any noun followed by a foreign word; 5) proper-noun followed by proper noun; and 5) numeral followed by any noun. Quotations were kept from the question. We must note that the POS tagger sometimes fails: "HOW/WRB DID/VBD JERRY/NNP GARCIA/NNP DIE/NNP ?", where the main verb "DIE" is tagged as a proper noun (NNP) and "DID" (VBD) wrongly becomes the only verb in the sentence.

- Verb expansion (VE)

Query type	Coverage C@100	Questions Covered	Passages Correct	# Passages	Precision P@100	Precision P@20
Okapi BM25 + AQUAINT	0.903	327	5,368	36,200	0.1483	0.2381
Okapi BM25 + Terabyte	0.887	319	9,146	34,738	0.2633	0.3229

Table 6.12: Effectiveness of the document retrieval in the initial set

Query type	Coverage C@20	Questions Covered	Passages Correct	# Passages	Precision P@20
Bag-of-word	0.738	267	1269	7240	0.1753
Bag+stem	0.710	257	1251	7240	0.1728
Boolean (and)	0.483	175	669	3787	0.1767
Quote	0.735	266	1261	7240	0.1742
Quote+Phrases	0.669	242	1076	7032	0.1530
VE	0.746	270	1223	7240	0.1689
VE+Quote	0.749	271	1226	7240	0.1693
Replacement	0.749	271	1412	7240	0.1950

Table 6.13: Passage Retrieval from top 150 Okapi documents in the AQUAINT Corpus

In preliminary works, particularly in the context of TREC-QA, we noticed that expanding verbs tends to improve effectiveness. To identify the verbs we used the parser described by Clarke et al.in [21], and not the POS tagger. Each regular verb is stemmed and all irregular verbs are expanded.

- Verb expansion plus Quotes

  These queries have both expanded verbs and quotes from the original questions. These components, along with some heuristics expansions, form queries used in MultiText's participations on the QA task in TREC 10 through 12. The words in the quote are also added as single words to the query.

Along with these formulations, we used our replacement method described in section 6.3.1.

6.6.1 Evaluation

We evaluate the performance of the different query formulation strategies in passage retrieval using the TREC 12 QA task question as the test set. We focus on the 413 factoid questions from which

	Bag +stem	Bool	Quote	Quote +Phrase	VE	VE +Quote	Repl.
Bag-of-word	0.2096	0.1287	0.1003	2.76E-005	0.1632	0.1634	0.0003
Bag+stem	-	0.3234	0.3864	0.0148	0.9305	0.9258	1.29E-005
Bool	-	-	0.1568	0.8451	0.4246	0.4067	0.0014
Quote	-	-	-	8.90E-005	0.3033	0.3109	7.51E-005
Quote +Phrases	-	-	-	-	0.0086	0.0104	2.78E-009
VE	-	-	-	-	-	1.0000	1.69E-005
VE+Quote	-	-	-	-	-	-	1.91E-005

Table 6.14: Wilcoxon p-values for p@20 in documents from the AQUAINT corpus

Query type	Coverage C@20	Questions Covered	Passages Correct	# Passages	Precision P@20
Bag-of-word	0.751	272	1894	7240	0.2616
Bag+stem	0.735	266	1835	7240	0.2535
Boolean (and)	0.702	254	1474	5640	0.2613
Quote	0.754	273	1891	7240	0.2612
Quote+Phrases	0.718	260	1681	7090	0.2371
VE	0.785	284	1877	7240	0.2593
VE+Quote	0.785	284	1899	7240	0.2623
Replacement	0.757	274	2033	7240	0.2808

Table 6.15: Passage Retrieval from top 150 Okapi documents in the Terabyte Corpus

362 have available patterns for automatic judgments (lenient[2]). To produce a better understanding of the differences between the different query formulation and the replacement methods, we use the same queries in two target corpora: the official TREC corpus for QA task—the AQUAINT corpus—and the terabyte collection described in Section 2.4.1 and used in [25, 26, 98]. All of the passages retrieved are of the same size, 170 words (∼1000 bytes).

The effectiveness was measured by means of *coverage*, the percentage of the 362 questions where at least one retrieved passage contains the answer, at 20 documents (C@20); and *precision*, also at 20 documents (P@20).

The original passage retrieval method described in Section 6.3.1 was used for the different query formulations; the replacement method used the bag-of-words queries. However, since the replacement method may need to scan the whole corpus for replacements, we decided to use a

[2]In lenient judgment a match to the pattern is enough to consider the answer correct

	Bag +stem	Bool	Quote	Quote +Phrase	VE	VE +Quote	Repl.
Bag-of-word	0.0283	0.6338	0.5062	0.0001	0.8277	0.9937	0.0005
Bag+stem	-	0.1358	0.0354	0.1455	0.1474	0.1078	1.43E-006
Bool	-	-	0.6657	0.0060	0.4872	0.5710	0.0540
Quote	-	-	-	2.96E-005	0.8786	0.9336	0.0010
Quote +Phrases	-	-	-	-	0.0037	0.0007	1.80E-008
VE	-	-	-	-	-	0.2839	0.0013
VE+Quote	-	-	-	-	-	-	0.0031

Table 6.16: Wilcoxon p-values for p@20 in documents from the Terabyte corpus

strategy commonly adopted by many QA systems to speed up the process of passage selection: select an initial set of documents, using a standard document retrieval scoring function, from which the passages are extracted.

We use the Okapi BM25 formula to extract the initial set of 150 documents. The queries used to extract this initial set is the bag-of-words with stemming. The effectiveness of document retrieval when creating the initial set is shown in Table 6.12. Since passages are extracted from the initial set, the effectiveness of the document retrieval is an upper bound for the passage retrieval.

For each query formulation a single passage is extracted from each document using equation 6.4. The same procedure is executed for the replacement method: one passage per document, passages scored by equation 6.7 with hotspots of 20 words.

The results of the passage selection in the AQUAINT corpus are shown in Table 6.13. Both verb expansion strategies and the replacement methods cover the highest number of questions. In precision at 20 passages the replacement method is better: the difference with any other query formulation is statistically significant at 99% significance level using Wilcoxon signed rank test, as shown in Table 6.14.

For the Terabyte corpus the results are shown in Table 6.15. Once again, the verb expansion strategies yield better coverage. The replacement method is worse than verb expansion in coverage but it is again the best in precision, with the differences between the replacement and other methods, with exception of the boolean queries, being statistically significant at 99% using Wilcoxon signed rank test.

From all the strategies, the use of phrases has the worst outcome. Phrases can be rewritten in different forms and, as consequence, be absent from some relevant passages. This outcome can also be explained by the scoring functions being designed to handle individual terms in order to address the bag-of-word approach and assuming independence among query terms. The same is not observed when using quotes, since quotes are important as specified and must not be rewritten. Verb expansion consistently improves coverage but results in precision at 20 are mixed, mostly not statistically significant.

Boolean queries are more restrictive: fewer passages are retrieved when these queries are used. This reduction helps final precision since every correct passage will have a greater impact. The coverage of boolean queries is smaller, a result of the reduced number of passage (i.e. less chance to cover questions). These findings suggest an explanation for the successful adoption of boolean queries, used in multiple iterations, in some QA systems [112, 73, 97]. Nonetheless, it is arguable that a QA system that can take advantage of the redundancy of answer strings [23, 10] to find answers to questions would benefit from a large number of passages, if the precision is similar.

6.7 Summary

In this Chapter we presented a new method to score objects in information retrieval tasks, with particular a focus on passages and documents, when one or more query terms are missing. In this method, we find replacements for the query terms in each object we score, if necessary, and use the original scoring function afterwards, adjusting the weight of replacement according to its relation with the original query term.

The results in the document retrieval are better than the original method which ignores missing terms. The difference is statistically significant.

For passage retrieval the same trend found in document retrieval is repeated. The new method provides better effectiveness when missing terms are left out by the original method. We also compare the new method of replacement with some explicit query expansion strategies in the context of passage retrieval for question answering. The new method outperforms these original methods using these query expansion strategies.

Chapter 7

Conclusions and Future Work

We have presented new ways to compute and apply lexical affinity in natural language applications. For the computation of lexical affinity based on co-occurrence frequency we proposed two new methods for point estimation. These methods improve performance in a set of synonym questions when compared to existing methods. The first method explores the proximity by adding extra weight to co-occurrences in close range while the second method use a more coarse estimator, based on documents, but also with emphasis on proximity. All point estimation methods can be viewed as smoothing techniques that can be applied to other applications, such as speech recognition or information retrieval based on language models. However, unlike other smoothing techniques, the probability mass reserved from co-occurrences will be divided among words occurring in proximity. This can be viewed as a "semantic" smoothing since the redistribution of probability mass will be done on words in the same context.

We also presented new parametric models for lexical affinity based on distance distribution of lexical units. These distributions fit the data in two flavours: a model for independence and another to describe the strength of lexical affinity at different distances. The independence model uses the mean distance to calculate the parameter for the geometric distribution that governs the distance between lexical unit pairs. The lexical affinity model uses a gamma distribution, for which the parameters are calculated using a maximum likelihood estimator to fit the data. Along with these distributions, we also presented an algorithm to compute all of the observations

between pairs of lexical units in sub-linear time, by benefiting from the inverted list used to index the corpus. This allows for on-line computation of these models which can be very helpful given the sheer number of possible lexical unit pairs. Thus, instead of computing the model for all pairs ($O(|V|^2)$), from which many will not be used, we can defer the computation of the model until necessary to do so.

The parametric models provide a source of estimation from which other models or measures can be built. As an example, we use two measures from these parametric models: the skew of the lexical affinity distribution, and the log-likelihood ratio over intervals. These measures are used in the synonym questions. In particular, the log-likelihood ratio over intervals provides the best overall absolute performance in the TOEFL synonym questions. The use of skew eliminates the need to specify window sizes as required in the models based on point estimation. These new models of parametric lexical affinity can also be used as smoothing techniques. The availability of a parametric function allows us to compute the number of co-occurrences at any distance, including those for which no examples have been seen in the training data.

Another application of lexical affinity models is in information retrieval. In general, due to problems like vocabulary mismatch and query drift, the IR engines allow documents to be retrieved even when they only it partially match the query. We proposed a new way to score missing terms in probabilistic models: we search the document for a replacement, using lexical affinity models, and adapt the term weight based on how strong the relationship between the missing query term and the replacement is. Experiments in passage retrieval and document retrieval show significant improvement when missing terms are replaced.

Future Work

There are many applications where the lexical affinity models presented in this thesis can be applied. These models can be viewed as language models that are not biased to short-range grammatical constructs but also allow semantic relationships to be included and inferred from the model. Most of applications of language modeling such as speech recognition, information retrieval and others benefit from these models.

Our models for lexical affinities are based on pairs of lexical units; however, there is no con-

straint on using these same ideas for three or more lexical units. As happens in *n-gram* models, this also results in an increase the number of parameters in the model, and as such, poses a challenge for efficient resource management. Since inverted lists used to index corpora are all based on unigram models, they are not optimized for lexical unit pairs or higher-order approximation models. An alternative is, as mentioned earlier, to calculate the lexical affinity between pairs only when needed but, even in a sublinear fashion, it may be very slow to use these models since the number of pairs is high. Besides, it is not efficient to cache models for lexical unit pairs since there are many of them will not be used frequently in most applications.

Our evaluation on synonym questions is a step forward in the understanding of affinity and co-occurrence estimates. It is not exhaustive, however. Further evaluations are necessary, in particular, given the existing evaluations for affinity measures in different natural language applications/phenomena, it would be interesting to create a more controlled environment for both estimation and affinity measures that could be used as a general evaluation framework for any lexical phenomena. An interesting question that could be raised in our evaluation is the relatively small number of distractors (4) for each question. In applications such as the scoring of missing terms in Chapter 6, the number of lexical units tested for replacement is much larger than that.

It was also noted that the context available from the sentence was not helpful to disambiguate the choices in the synonym questions. An alternative to use of context is to employ the same strategy used by methods such LSA and HAL, which use the context from the corpus rather than only a sentence. Although we use second order statistics, where the similarity between terms, by making indirect comparison, we did not explore the full potential the corpus provides as supporting evidence. This is an usual approach for word sense disambiguation but could be adapted to help eliminate candidate synonyms that are not related with the target word.

Another possible alternative for semantic similarity is to combine statistical methods, such as the affinity models presented in this thesis, along with knowledge-based approaches such as lexicons and thesaurus. Although this approach was implemented by Turney [101], it is more expensive and its generalization may not be easily achieved. In particular, this approach could be attempted on scoring missing terms method.

The application of the log-likelihood measure derived from the affinity models in the fill-in-

the-blanks is new and more experiments could be attempted. In particular, since the model used to answer the questions completely ignores the syntatic structure of the questions, it would be interesting to assess how much impact the syntax would bring to this task.

In the scoring missing terms in information retrieval, the affinity is calculated by making use of point estimation and PMI. Although a normalized weight can be derived from that measure, it would be interesting to use the parametric models instead of point estimation. This would also require a weight normalizing procedure for the replacement. Another alternative is to use syntactic features in the replacements; it is not clear what would be the impact of applying syntatic constratins in the replacements.

Bibliography

[1] Steven Abney. Statistical methods and linguistics. In Judith L. Klavans and Philip Resnik, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 1–23. MIT Press, 1996.

[2] E. Agirre and D. Martinez. Exploring automatic word sense disambiguation with decision lists and the web. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 11–19, Saarbrucken, Germany, 2000.

[3] Doug Beeferman, Adam Berger, and John Lafferty. A model of lexical attraction and repulsion. In *Proceedings of the 35th Annual Meeting of the Association for Computation Linguistics and 8th Conference of the EACL*, pages 373–380, 1997.

[4] Morton Benson. The structure of the collocational dictionary. *International Journal of Lexicography*, 2(1):3–14, 1989.

[5] Morton Benson. Collocations and general-purpose dictionaries. *International Journal of Lexicography*, 3(1):23–35, 1990.

[6] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, 1999.

[7] Michael W. Berry. Large scale singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[8] Godelieve L. M. Berry-Rogghe. The computation of collocations and their relevance to lexical studies. In N. Hamilton-Smith A. J Aitken, R. W. Bailey, editor, *The Computer and Literary Studies*, pages 103–112. University Press, Edinburgh, New York, 1973.

[9] Douglas Biber. Representativeness in corpus design. *Literary and Linguistic Computing*, 8:1–15, 1993.

[10] Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. Data-intensive Question Answering. In *Proceedings of 2001 Text REtrieval Conference*, Gaithersburg, MD, 2001.

[11] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 265–276. ACM Press, 1997.

[12] P. F. Brown, P. V. deSouza, R. L. Mercer, T. J. Watson, V. J. Della Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[13] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART: TREC 3. In *Proceedings of Third Text REtrieval Conference*, Gaithersburg, MD, 1994.

[14] David Carmel, Eitan Farchi, Yael Petruschka, and Aya Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 283–290, Tampere, Finland, 2002.

[15] S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

[16] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th conference on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[17] Noam Chomsky. *Syntatic Structures*. The Hague, Mouton, 1957.

[18] Yaacov Choueka. Looking for needles in a haystack or locating interesting collocations expressions in large textual databases. In *Proceedings of the RIAO conference on User-Oriented Content-Based Text and Image Handling*, Cambridge, MA, 1988.

[19] Kenneth W. Church. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than p2. In *Proceedings of 18th International Conference on Computational Linguistics*, volume 1, pages 180–186, 2000.

[20] K.W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[21] C. L. A. Clarke, G. V. Cormack, T. R. Lynam, and E. Terra. *Advances in Open Domain Question Answering*, chapter Question answering by passage selection. Kluwer Academic Publishers. To appear, 2004.

[22] Charles L. A. Clarke, Gordon V. Cormack, and F. J. Burkowsky. An algebra for structured text search and a framework for its implementation. *Computer Journal*, 38(1), 1995.

[23] Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365. ACM Press, 2001.

[24] Charles. L. A. Clarke, Gordon V. Cormack, Thomas R. Lynam, C. M. Li, and G. L. McLearn. Web reinforced question answering. In *In proceedings of the 2001 Text REtrieval Conference*, Gaithersburg, MD, 2001.

[25] Charles L.A. Clarke, Gordon V. Cormack, M. Laszlo, Thomas R. Lynam, and Egidio Terra. The impact of corpus size on question answering performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 369–370, Tampere, Finland, 2002.

[26] Charles L.A. Clarke and Egidio Terra. Passage retrieval vs. document retrieval for factoid question answering. In *Proceedings of the 26th annual international ACM SIGIR conference*

on Research and development in information retrieval, pages 427–428, Toronto, Canada, 2003.

[27] Kevyn Collins-Thompson, Egidio Terra, Jamie Callan, and Charles L. A. Clarke. The effect of document retrieval quality on factoid question answering. In *ACM SIGIR Conference on Research and development in Information Retrieval*, 2004.

[28] Gordon V. Cormack, Ondrej Lhotak, and Christopher R. Palmer. Estimating precision by random sampling. In *Proceedings of the 22th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–274, 1999.

[29] Carolyn J. Crouch, Donald B. Crouch, Qingyan Chen, and Steven J. Holtz. Improving the retrieval effectiveness of very short queries. *Information Processing and Management*, 38(1):1–36, 2002.

[30] James R. Curran and Miles Osborne. A very very large corpus doesn't always yield reliable estimates. In *Proceedings of CoNLL-2002*, pages 126–131. Taipei, Taiwan, 2002.

[31] Ido Dagan, Lillian Jane Lee, and Fernando C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[32] Kareem Darwish and Douglas W. Oard. Probabilistic structured query methods. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 338–344, 2003.

[33] S. Deerwester, Susan T. Dumais, Thomas K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[34] Sabine Deligne and Yoshinori Sagisaka. Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 300–306, San Francisco, California, 1998. Morgan Kaufmann Publishers.

[35] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.

[36] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74, 1993.

[37] Stefan Evert. Significance tests for the evaluation of ranking methods. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 945–951, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.

[38] Stefan Evert. *The Statistics of Word Cooccurrences (Word Pairs and Collocations)*. Ph. D. dissertation, Universität Stuttgart, August 2004.

[39] Stefan Evert and Brigitte Kren. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 2001.

[40] C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[41] Olivier Ferret. Using collocations for topic segmentation and link detection. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[42] John Rupert Firth. *Studies In Linguistic Analisys*, chapter A Synopsis of Linguistic Theory, 1930-1955, pages 1–32. Basil Blackwell, Oxford, 3rd edition, 1957.

[43] Katerina T. Frantzi and Sophia Ananiadou. Extracting nested collocations. In *Proceedings of the 16th International Conference on Computational Linguistics*, 1996.

[44] Jianfeng Gao, Ming Zhou, Jian-Yun Nie, Hongzhao He, and Weijun Chen. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190, 2002.

[45] G. Grefenstette. 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*, 1993.

[46] Gregory Grefenstette. Use of syntactic context to produce term association lists for text retrieval. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 89–97. ACM Press, 1992.

[47] Toru Hisamitsu and Yoshiki Niwa. A measure of term representativeness based on the number of co-occurring salient words. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[48] C. Jacquemin. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 341–348, 1999.

[49] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. In *Proceedings of International Conference RANLP - 2003 (Recent Advances in Natural Language Processing)*, Borovets, Bulgaria, 2003.

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

[51] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1 and 2. *Information Processing and Management*, 36(6):779–808; 809–840, 2000.

[52] Stefan Kaufmann. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 591–595, 1999.

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

[54] Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–348, 2003.

[55] C. King and N. Stanley. *Building Skills for the TOEFL.* Thomas Nelson and Sons Ltd, second edition, 1989.

[56] Tibor Kiss and Jan Strunk. Scaled log likelihood ratios for the detection of abbreviations in text corpora. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[57] Kenji Kita, Yasuhiko Kato, Takasi Omoto, and Yoneo Yano. A comparative study of automatic extraction of collocations from corpora: Mutual information vs. cost criteria. *Journal of Natural Language Processing*, 1(1):21–33, 1994.

[58] Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the Web. In *Tenth International World Wide Web Conference*, pages 150–161, 2001.

[59] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, 2001.

[60] Thomas K. Landauer and Susan T. Dumais. 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, 1997.

[61] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, 2001.

[62] Lillian Jane Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 25–32, 1999.

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

[64] Michael E. Lesk. Word-word associations in document retrieval systems. *American Documentation*, 20(1):27–38, January 1969.

[65] Mark Liberman and Christopher Cieri. The creation, distribution and use of linguistic data.

In *Proceedings of the First International Conference on Language Resources and Evaluation*, May 1998.

[66] J. Lin, A. Fernandes, B. Katz, G. Marton, and S. Tellex. Extracting answers from the web using data annotation and knowledge mining techniques. In *The Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, 2002.

[67] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers*, 28(2):203–208, 1996.

[68] Chistopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, fifth edition, 1999.

[69] W.J.R Martin, B.P.F. Al, and P.J.G van Sterkenburg. *Lexicography: Principles and Practice*, chapter On the processing of a text corpus, pages 77–87. Academic Press, London, 1st edition, 1983.

[70] Tony McEnery and Andrew Wilson. *Corpus Linguistics*. Edinburgh University Press, 1996.

[71] Charles F. Meyer. *English Corpus Linguistics: An Introduction*. Cambridge University Press, 2002.

[72] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–214. ACM Press, 1998.

[73] Dan Moldovan, Marius Paca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 33–40, 2002.

[74] Rosamund Moon. *Fixed Expressions and Idioms in English*. Clarendon Press, 1998.

[75] M. Najork and J. L Wiener. Breadth-first search crawling yields high-quality pages. In *10th International World Wide Web Conference*, pages 114–118, 2001.

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

[77] T. Niesler and P. Woodland. Modelling word-pair relations in a category-based language model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 795–798, Munich, Germany, 1997.

[78] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619, 2002.

[79] Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. Towards terascale semantic acquisition. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 771–777, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.

[80] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2003.

[81] Darren Pearce. A comparative evaluation of collocation extraction techniques. In *Proceedings of the 3rd Language Resources Evaluation Conference*, 2002.

[82] Helen J. Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383, 1991.

[83] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, 1998.

[84] Reinhard Rapp. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, 2002.

[85] Philip Resnik and Mona Diab. Measuring verb similarity. In *22nd Annual Meeting of the Cognitive Science Society (COGSCI2000)*, Philadelphia, August 2000.

[86] Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, 2003.

[87] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of 17th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 232–241, Dublin, Ireland, 1994. Springer-Verlag New York, Inc.

[88] J. Rocchio. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance feedback in information retrieval, pages 313–323. Prentice-Hall Inc., 1971.

[89] Roni Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228, 1996.

[90] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[91] Celina Santamaría, Julio Gonzalo, and Felisa Verdejo. Automatic association of web directories to word senses. *Computational Linguistics*, 29(3):485–502, 2003.

[92] Peter Schäuble and Páraic Sheridan. Cross-language information retrieval (clir) track overview. In *The Sixth Text REtrieval Conference (TREC 6)*, Gaithersburg, MD, 1997.

[93] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

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

[95] Alan F. Smeaton, Fergus Kelledy, and Ruairi O Donnell. Trec-4 experiments at dublin city university: Thresholding posting lists, query expansion with wordnet and pos tagging of spanish. In *Proceedings of Fourth Text REtrieval Conference*, Gaithersburg, MD, 1995.

[96] Takaaki Tanaka. Measuring the similarity between compound nouns in different languages using non-parallel corpora. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[97] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47, 2003.

[98] Egidio Terra and Charles L. A. Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, pages 244–251, Edmonton, Alberta, 2003.

[99] A. Thanopoulos, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of collocation extraction metrics. In *Proceedings of the 3rd Language Resources Evaluation Conference*, pages 620–625, 2002.

[100] Peter D. Turney. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. In *Proceedings of European Conference on Machine Learning-2001*, pages 491–502, 2001.

[101] Peter .D. Turney, Littman M.L., J. Bigham, and V. Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of International Conference RANLP - 2003 (Recent Advances in Natural Language Processing)*, Borovets, Bulgaria, 2003.

[102] Cornelius J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

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

[104] Ellen M. Voorhees. The TREC-8 Question Answering track report. In *In proceedings of the 8th Text REtrieval Conference*, pages 77–82, Gaithersburg, MD, 1999.

[105] Ellen M. Voorhees. Overview of the TREC-9 Question Answering track. In *In proceedings of 9th Text REtrieval Conference*, pages 71–80, Gaithersburg, MD, 2000.

[106] Ellen M. Voorhees. Overview of the TREC 2001 Question Answering track. In *In proceedings of 2001 Text REtrieval Conference*, pages 42–51, Gaithersburg, MD, 2001.

[107] Ellen M. Voorhees. Overview of the TREC 2002 Question Answering track. In *In proceedings of 2002 Text REtrieval Conference*, pages 57–68, Gaithersburg, MD, 2002.

[108] Ellen M. Voorhees. Overview of the TREC 2003 Question Answering track. In *In proceedings of 2003 Text REtrieval Conference*, pages 14–27, Gaithersburg, MD, 2003.

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

[110] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of 19th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 4–11, Zurich, 1996.

[111] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with Local Context Analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.

[112] Hui Yang, Tat-Seng Chua, Shuguang Wang, and Chun-Keat Koh. Structured use of external knowledge for event-based open domain question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 33–40, 2003.

[113] D. L. Yeung, C. L. A. Clarke, G. V. Cormack, T. R. Lynam, , and E. Terra. Task-specific query expansion (multitext experiments for trec 2003). In *2002 Text REtrieval Conference*, Gaithersburg, MD, 2003.

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

[115] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.