Refining Query Expansion Terms using Query Context

Reuben Crimp
University of Otago
Dunedin, New Zealand
rcrimp@cs.otago.ac.nz

Andrew Trotman
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

ABSTRACT

Query expansion is commonly used to combat the vocabulary mismatch problem, it bridges the disparity between the vocabulary used in the corpus and search queries. However, if expansion terms are not chosen carefully, there is a risk of including spurious expansion terms, which can broaden the potential interpretations of the modified query. Unintentionally increasing the semantic ambiguity in this way is known as *query drift.*

In this short paper we propose using the *query context* to inform the expansion term selection process. Using WordNet as an initial source of expansion terms, we refine the candidate expansions by discriminating relevancy. We found that our term selection process is more effective than the standard approach. Our technique targets terms which relate to the entire query as a whole, but predominately focuses on excluding spurious expansion terms. Both help reduce query drift and increase query performance.

CCS CONCEPTS

• **Information systems → Query reformulation**;

KEYWORDS

Ad-hoc Retrieval, Query Expansion, Query Drift, Thesaurus, WordNet, Wu-Palmer Similarity

ACM Reference format:
Reuben Crimp and Andrew Trotman. 2018. Refining Query Expansion Terms using Query Context. In *Proceedings of 23rd Australasian Document Computing Symposium, Dunedin, New Zealand, December 11–12, 2018 (ADCS '18),* 4 pages.
https://doi.org/10.1145/3291992.3292000

1 INTRODUCTION

Human generated text is plagued with the vocabulary mismatch problem, human authors frequently use different words to describe the same concept. This obviously applies to distinct authors, but also the same author at different times. IR researchers coined another term for the same phenomenon, *the term mismatch problem.* Proving why we have the problem in the first place. In The Text REtrieval Conference (TREC) Ad Hoc Tasks, queries have been manually tagged to relevant documents by experts. Many of these documents do not contain all the terms from the associated query [19], i.e. there is a disparity between the vocabularies. This poses a problem for IR systems which are based on *term matching.*

Query expansion specifically addresses the term mismatch problem, by appending extra terms which are determined to be relevant to the original search query. The intention is to broaden the vocabulary used in the query, to more closely match the vocabulary of the corpus. If terms are not chosen carefully however, the modified query can drift from the intended meaning.

Automatic relevance feedback *(pseudo relevance feedback, blind relevance feedback)* is a common technique used to obtain expansion terms directly from the corpus itself. Choosing content bearing terms from the top documents retrieved when using the original query. Thus making the modified query more similar to the *relevant* documents, addressing the vocabulary disparity directly from the corpus itself. Relevance feedback is usually implemented using a variant of the Rocchio algorithm [12]. But also Binary Independence Model (BIM) [11], Chi-square [5], Robertson selection value (RSV) [10], and Kullback-Leibler distance (KLD) [2]. Language modelling has been applied to automatic query expansion problems with success [1]. Usually no term semantics are explicitly used, only co-occurence data mined from the corpus. Some recent experiments however have begun to include term relationships [1].

A thesaurus is another way to obtain expansion terms, which is much faster than relevance feedback techniques, but also generally less effective. Early experiments indicated general purpose thesauri were effective [17]. Later experiments proved their behaviour inconsistent, sometimes they would improve a query, other times degrade a query [13]. Voorhees claimed that lexical-semantic relationships provide little benefit, but have the 'potential to improve an initial query' [16]. Recent experiments have shown that effects of unintentional query drift from thesaurus based approaches can be minimized through term-frequency merging [4].

Using synonyms from a thesaurus for vocabulary expansion makes perfect sense. As synonyms are the words which share a similar meaning, but have different spellings. However, the problem is complicated by homographs (homonyms & heteronyms) and polysemes. The terms which share the same spelling, but have different meanings. This causes ambiguity at the individual term level. If we do not address the issue of ambiguity, we risk choosing the wrong synonyms, and causing query drift.

This short paper describes a technique to prevent query drift. Discriminating ideal expansion terms from spurious ones is achieved by inferring their relevancy to the original query's context.

ADCS '18, December 11–12, 2018, Dunedin, New Zealand

ACM ISBN 978-1-4503-6549-9/18/12…$15.00
https://doi.org/10.1145/3291992.3292000

2 EXPERIMENT CONDITIONS

We used the ATIRE search engine [14], which is a bag of words term matching information retrieval system. ATIRE uses a variant of Okapi BM25 [9] for it's ranking function [15]. We used the TREC ad-hoc retrieval tracks 1 through 8. Approximately 1,300,000 documents, 400 queries, and the set of binary relevance evaluations. We used WordNet [8] as a source of expansion terms.

2.1 WordNet 3.0

WordNet is a hierarchically organized lexical database. It identifies semantic relationships like synonymy, homonymy, antonymy, hypernymy, hyponymy, meronymy, metonymy, holonymy... as well as dictionary features, like *part of speech* labels. The majority of the following definitions are widely accepted in the lexicography community, and all are used to describe elements of WordNet.

Term A string of characters. Can exist independent of meaning.
Word A term authored with intended meaning, which is dependent on the context in which it was written. In rare cases, multiple meanings are simultaneously intended e.g. double entendre.
Word Sense A well defined meaning associated with a term.
Synonyms Terms that share same Word Sense.
SynSet (WN) Synonym sets (single thesaurus entry)
SynSets (WN) Set of all Word Senses for a term. i.e. potential candidates for the authors intended meaning(s).
Lemmas (WN) Set of distinct terms sharing the same Word Sense.

2.2 Measuring Similarity

Term similarity is often used to infer relevancy. The more similar a candidate expansion term is to a query, the more likely it will improve the performance of the query. In IR it is common to compare terms using the corpus statistics e.g. the dice coefficient, Jaccard Index and point-wise mutual information [3]. They all infer similarity from co-occurrence data, a form of *Syntactic Similarity.*

We focused on *Semantic Similarity*, using the structural relationships WordNet provides. specifically the Wu-Palmer function [18], formally defined in Equation 1. The Wu-Palmer similarity function finds the shortest path from one SynSet to another SynSet. Since Word Senses which are separated by a longer path are generally less similar semantically than those with a shorter path. Wu-Palmer also accounts for large conceptual leaps, by accounting for the distance to the least common subsumer (first common ancestor).

$$sim(a,b) = \frac{2 * depth(LCS(a,b))}{depth(a) * depth(b)} \tag{1}$$

$$LCS(a,b) = \text{Least common subsumer} \tag{2}$$

$$depth(a) = \text{Shortest path from root to } a \tag{3}$$

3 LOCAL CONTEXT DISAMBIGUATION

Let us briefly consider the following contrived sentence:

> *"I made all of my profit from the **rushes** on the **banks**"*

Teaching an IR system to understand *puns* should not be necessary, so let us consider something more reasonable, an *antanaclasis*:

> *"I was **banking**$_1$ on the aeroplane to **bank**$_2$ into the*
> *river **bank**$_3$, for I had over insured it with my **bank**$_4$"*

The term(s) of interest here is *bank*, which has many associated Word Senses. WordNet has 18 recorded for *bank.* A human reader is expected to be able to intuitively disambiguate each occurrence of *bank*, by referencing the surrounding context. One can infer the part of speech each term is (noun, verb, adjective...) using the conventional rules of grammar. And from the structural relationships grammar imposes, one can also infer semantic information using the terms preceding/following the term of interest. For example *bank*$_3$ is preceded by the term *river*, which is preceded by the determiner *the.* So the Word Sense of *bank*$_3$ is clearly a noun describing the land alongside a river. Table 1 shows all four intended definitions from the example sentence.

Table 1: Four (of 18) Possible Definitions of the Term *bank*

Term	Part of speech	Definition
bank on$_1$	phrasal verb	To rely on confidently.
bank$_2$	noun	The land alongside a river or lake.
bank$_3$	verb	To tilt sideways in making a turn.
bank$_4$	noun	A financial establishment.

3.1 Standard Term Selection

The naive approach to selecting expansion terms from a thesaurus, is to include ***all*** entries from the thesaurus, without restriction. This is naive since it is highly unlikely that the author of the original search query intended more than a single word sense for each term. For example, if expanding the term *"bank"*, it makes little sense to include expansion terms *"tilt"*, *"coast"* and *"treasury"*, since in the context of the query, *"bank"* would only mean one of those.

Using all synonyms to expand a query only makes sense if you are hedging your bets, if you are unable to empirically determine the Word Sense of individual terms in the query. While it does increase the chance of selecting relevant expansion terms, it also increases the chance of selecting spurious expansion terms. Especially for original query terms which are homonyms, heteronyms and/or polysemous. Another identifiable problem is that using a thesaurus in this way tends to retrieve terms that are related to only part of the query (i.e. a single term), rather than the entire query. In either case the modified query will experience query drift, towards unrelated concepts, or a subset of the queries intended meaning.

3.2 Refining the Expansion Terms

The obvious solution to the previous problem is to select expansion terms from a single thesaurus entry, the entry associated with the authors intended meaning. But how do we choose the correct entry? WordNet provides 18 possible Choices for *"bank"*.

We propose using the context of the query itself to help inform the expansion term selection process. The supposition is that most (or all) of the original query terms relate to the same concept, the user's desired information. Our objective is to find a shared Word Sense between the original query terms, by comparing the associated SynSets. We Perform a pairwise similarity comparison tournament between the SynSets, and identify the SynSet which correlates strongest with the other terms. The comparison function we used is the Wu-Palmer similarity function, mentioned previously.

We will refer to the term being expanded as the *term of interest*, and the set of remaining terms from the original query as the *other terms*. Each SynSet derived from the term of interest is compared with each SynSet of the *other terms*, using the Wu-Palmer Similarity as the comparison function. The highest scoring SynSet (belonging to the *term of interest*) wins the tournament, and is chosen as the most likely candidate for the intended Word Sense. Then each of the Lemma terms from the winning SynSet are added to the modified query. This process is repeated for each term in the original query, treating each as a *term of interest* during expansion. The algorithmic complexity of this process is exponential with regard to the number of query terms, and the number of SynSets for each query term. In practice however the performance is reasonable, since queries are usually short, and SynSets are usually quite small.

For the query $Q_1 = \{"river", "bank"\}$, *"river"* has only 1 SynSet, and *"bank"* has 18. So only 18 comparisons need to be made. Our method correctly identified the two SynSets as:
river *"a large natural stream of water (larger than a creek)"* and
bank *"bank "sloping land (especially the slope beside a body of water)"*
Let us consider another query $Q_2 = \{"pool", "cue"\}$, *"pool"* has 11 SynSets, and *"cue"* has 5:
pool *"(an excavation that is (usually) filled with water"*
cue *"sports implement consisting of a tapering rod used to strike a cue ball in pool or billiards"*

Our method here, was not able to infer the correct Word Senses. Even though both *"pool"* and *"cue"* include the term *"billiard"* in one of their respective SynSet Lemmas. This suggests that the Wu-Palmer similarity function is not perfect, and a more informed comparison function could be used.

4 RESULTS

The results in Table 2 includes a *baseline*, which is just naive term matching, no query expansion. The Table also includes a *Rocchio* based implementation of relevance feedback. With parameters, top 17 documents and top 5 terms. Rocchio was chosen over other more effective techniques as it is well established and commonly used as a familiar point of reference. The results from our experiments are labeled *All-SynSets*, as the standard approach described in section 3.1, and *One-SynSet* is our improved method described in section 3.2. We also included two separate query reformulation techniques. The standard approach of appending terms directly to the query, and also term-frequency-merging [4]. Term-frequency-merging attempts to normalize the disparity between terms which have many expansion terms, and terms that have few.

The results are promising, as can be seen in Table 2. The standard approach *All-SynSets* with *appending* only beats the *baseline* in one case (TREC-6). Whereas our method *One-SynSet* with *appending* improves upon the *baseline* in all but one case (TREC-6). And *One-SynSet* with *tf-merging* beats the *baseline* in all cases. Blind relevance feedback is still a strong contender, as it remains unbeaten in TREC-1, TREC-2 and TREC-3.

We did compute two-tailed t-tests on the 400 paired MAP samples, with the Bonferroni correction. We tested the baseline against, All-Synsets, and One-SynSet, and in every case we obtained p-values < 0.05. Which suggests that the observed differences cannot be explained by chance alone.

4.1 Failure Analysis

If we look at the results of query 2 from TREC-7, *"british chunnel impact"* in Table 3, we can see that our method has an enormously positive impact. This is in part because it only includes 8 extra terms from 3 SynSets (one for each term), instead of 21 terms from 9 different SynSets. But more specifically, the Lemmas of *impact* include *"touch, bear, shock"*, which caused significant query drift in the All-SynSet case.

However if we look at TREC-7 query 15 in Table 4, *"el nino"*. Our method is disastrous. Inspecting the expansion terms chosen shown in Table 5, we can see that the term "el" is incorrectly identified as the Chicago "L" Train. This example is a particularly bad, as *"el nino"* is a Spanish phrase adopted by English speakers, which WordNet has not accounted for.

4.2 Using the Similarity Score as a Predictor

The larger the similarity score, the more relevant the SynSet is assumed to be. So it is obvious to try and use the similarity score to predict the improvement of the modified query. Measuring the correlation using the Pearson bivariate method gives a score of approximately 0 (precisely -0.0617). This suggests *no* simple correlation between the *Wu-Palmer Similarity Score* and the *Mean Average Precision improvement (from the baseline).*

5 FUTURE WORK

There is still no guarantee that our method correctly identifies the intended Word Sense of the original query term(s). This method is still very naive, as it does not infer much information from the query context. A more sophisticated language model could do so much more.

5.1 Parts of Speech

WordNet provides part of speech tagging (noun, verb, adjective...), which could be used to improve our method. Since it is possible to select a noun SynSet for a query term that is clearly expressed as a verb. This obviously assumes that the user generates a search query that conforms to the expected rules of grammar, which is not guaranteed.

5.2 Improving the Comparison Function

The Wu-Palmer Similarity function is a one-to-one comparison, and is suited to queries with only 2 terms. For longer queries we used a tournament of comparisons, but performing a groupwise comparison directly would be more appropriate. Like the groupwise Jaccard Index and or the groupwise Resnik comparison [7].

Comparisons of SynSets within WordNet is based on finding the shortest path in the graph. A groupwise comparison would be

Table 2: Mean Average Precision Across TREC 1-8

Term selection	Expansion	TREC-1	TREC-2	TREC-3	TREC-4	TREC-5	TREC-6	TREC-7	TREC-8
None (baseline)		0.2181	0.1993	0.2324	0.1727	0.1432	0.1891	0.1905	0.2195
Rocchio	appending	**0.2601**	**0.2521**	**0.2988**	0.2041	0.1369	0.1646	0.2185	0.2460
All-SynSets	appending	0.2128	0.1961	0.2243	0.1265	0.1327	0.2041	0.1822	0.2187
All-SynSets	tf-merging	0.2323	0.2095	0.2379	0.1721	**0.1618**	**0.2214**	0.1953	**0.2440**
One-SynSet	appending	0.2318	0.2104	0.2398	0.2101	0.1476	0.1781	0.2163	0.2301
One-SynSet	tf-merging	0.2380	0.2286	0.2529	**0.2179**	0.1537	0.2007	**0.2211**	0.2310

Table 3: TREC-7 query 2 Mean Average Precision

Term selection	Expansion	MAP
None (baseline)		0.051273
All-SynSets	appending	0.042131
All-SynSets	tf-merging	0.050945
One-SynSet	appending	0.311783
One-SynSet	tf-merging	0.204493

Table 4: TREC-7 query 15 Mean Average Precision

Term selection	Expansion	MAP
None (baseline)		0.786084
All-SynSets	appending	0.784300
All-SynSets	tf-merging	0.846636
One-SynSet	appending	0.194717
One-SynSet	tf-merging	0.812008

Table 5: TREC-7 query 15 Expansion Terms for *"el"*

All-SynSets	alt	altitude	el	elevation	
One-SynSet	el	elevated	overhead	railroad	railway

equivalent to finding a subgraph that includes at least one SynSet from each query term. This subgraph would be a tree, since a minimal graph has no cycles. And any intermediary nodes (SynSets) used to construct the tree would be used for expansion terms. This can be described as a variant of the Steiner tree problem, but in this scenario we cannot predict the exact subset of vertices that would be included. Which makes an already NP-hard problem, even harder. It is also worth noting that query terms do not have equal importance, e.g *stop words* can often be excluded. So query terms do not require a participating SynSet in the subgraph, in which case The Prize-Collecting Steiner Tree Problem [6] fits best. Minimize edge cost and maximize vertex profit. In our case profit would be indicated by *stop words* having small values and *content bearing terms* having high values.

6 CONCLUSIONS

Overall results are promising but unsurprising. Relevance feedback is still unbeaten, as it has the potential to find expansion terms which are not semantically related to any of the original query terms. i.e. it can include related concept(s), that the user did not think to include, which is beyond the scope of vocabulary mismatch.

Using our query context informed method, we refined the expansion terms obtained from a thesaurus, which is more effective than using a thesaurus blindly. Term frequency merging was able to be applied to both methods, and improved them in both cases.

REFERENCES

[1] Jing Bai, Dawei Song, Peter Bruza, Jian-Yun Nie, and Guihong Cao. 2005. Query Expansion Using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*. ACM, New York, NY, USA, 688–695.

[2] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. 2001. An Information-theoretic Approach to Automatic Query Expansion. *ACM Trans. Inf. Syst.* 19, 1 (Jan. 2001), 1–27.

[3] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1, Article 1 (Jan. 2012), 50 pages.

[4] Reuben Crimp and Andrew Trotman. 2017. Automatic Term Reweighting for Query Expansion. In *Proceedings of the 22Nd Australasian Document Computing Symposium (ADCS 2017)*. ACM, New York, NY, USA, Article 3, 4 pages.

[5] Tamas E. Doszkocs. 1978. AID, An Associative Interactive Dictionary for Online Searching. *Online Information Review* 2 (12 1978), 163–173.

[6] David S. Johnson, Maria Minkoff, and Steven Phillips. 2000. The Prize Collecting Steiner Tree Problem: Theory and Practice. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 760–769.

[7] Prashanti Manda and Todd Vision. 2018. An analysis and comparison of the statistical sensitivity of semantic similarity metrics. *bioRxiv* (2018).

[8] G. A. Miller. 1995. WordNet: A Lexical Database for English. *CACM* 38, 11 (1995), 39–41.

[9] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. 1996. Okapi at TREC-3. 109–126.

[10] S. E. Robertson. 1991. On Term Selection for Query Expansion. *J. Doc.* 46, 4 (Jan. 1991), 359–364.

[11] S. E. Robertson and Sparck J. K. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science (pre-1986)* 27, 3 (May 1976), 129. Copyright - Copyright Wiley Periodicals Inc. May/Jun 1976; Last updated - 2010-06-09.

[12] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.). Englewood Cliffs, NJ: Prentice-Hall, 313–323.

[13] G. Salton and M. E. Lesk. 1968. Computer Evaluation of Indexing and Text Processing. *J. ACM* 15, 1 (1968), 8–36.

[14] A. Trotman, C. L. A. Clarke, I. Ounis, S. Culpepper, M.-A. Cartright, and S. Geva. 2012. Open Source Information Retrieval: A Report on the SIGIR 2012 Workshop. *SIGIR Forum* 46, 2 (2012), 95–101.

[15] A. Trotman, A. Puurula, and B. Burgess. 2014. Improvements to BM25 and Language Models Examined. In *ADCS '14*. 58:58–58:65.

[16] E. M. Voorhees. 1994. Query Expansion Using Lexical-semantic Relations. In *SIGIR '94*. 61–69.

[17] Y.-C. Wang, J. Vandendorpe, and M. Evens. 1985. Relational thesauri in information retrieval. *JASIS* 36, 1 (1985), 15–27.

[18] Zhibiao Wu and Martha Palmer. 1994. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics (ACL '94)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 133–138.

[19] L. Zhao and J. Callan. 2010. Term Necessity Prediction. In *CIKM 2010*. 259–268.