Orthogonal Negation in Vector Spaces for Modelling Word-Meanings and Document Retrieval

Appeared in 41st Annual Meeting of the Association for Computational Linguistics
Sapporo, Japan, July 2003, pages 136–143

Dominic Widdows, Stanford University [*]
dwiddows@csli.stanford.edu

Abstract

Standard IR systems can process queries such as "web NOT internet", enabling users who are interested in arachnids to avoid documents about computing. The documents retrieved for such a query should be irrelevant to the negated query term. Most systems implement this by reprocessing results after retrieval to remove documents containing the unwanted string of letters.

This paper describes and evaluates a theoretically motivated method for removing unwanted meanings directly from the original query in vector models, with the same vector negation operator as used in quantum logic. Irrelevance in vector spaces is modelled using orthogonality, so query vectors are made orthogonal to the negated term or terms.

As well as removing unwanted terms, this form of vector negation reduces the occurrence of synonyms and neighbours of the negated terms by as much as 76% compared with standard Boolean methods. By altering the query vector itself, vector negation removes not only unwanted strings but unwanted meanings.

1 Introduction

Vector spaces enjoy widespread use in information retrieval (Salton and McGill, 1983; Baeza-Yates and Ribiero-Neto, 1999), and from this original application vector models have been applied to semantic tasks such as word-sense acquisition (Landauer and Dumais, 1997; Widdows, 2003) and disambiguation (Schütze, 1998). One benefit of these models is that the similarity between pairs of terms or between queries and documents is a continuous function, automatically ranking results rather than giving just a YES/NO judgment. In addition, vector models can be freely built from unlabelled text and so are both entirely unsupervised, and an accurate reflection of the way words are used in practice.

In vector models, terms are usually combined to form more complicated query statements by (weighted) vector addition. Because vector addition is commutative, terms are combined in a "bag of words" fashion. While this has proved to be effective, it certainly leaves room for improvement: any genuine natural language understanding of query statements cannot rely solely on commutative addition for building more complicated expressions out of primitives.

Other algebraic systems such as Boolean logic and set theory have well-known operations for building composite expressions out of more basic ones. Set-theoretic models for the logical connectives 'AND', 'NOT' and 'OR' are completely understood by most researchers, and used by Boolean IR systems for assembling the results to complicated queries. It is clearly desirable to develop a calculus which combines the flexible ranking of results in a vector model with the crisp efficiency of Boolean logic, a goal which has long been recognised (Salton et al., 1983) and attempted mainly for conjunction and disjunction. This paper proposes such a scheme for negation, based upon well-known linear algebra, and which also implies a vector form of disjunction. It turns out that these vector connectives are precisely

[*]This research was supported in part by the Research Collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University, and by EC/NSF grant IST-1999-11438 for the MUCHMORE project.

those used in quantum logic (Birkhoff and von Neumann, 1936), a development which is discussed in much more detail in (Widdows and Peters, 2003). Because of its simplicity, our model is easy to understand and to implement.

Vector negation is based on the intuition that unrelated meanings should be orthogonal to one another, which is to say that they should have no features in common at all. Thus vector negation generates a 'meaning vector' which is completely orthogonal to the negated term. Document retrieval experiments demonstrate that vector negation is not only effective at removing unwanted terms: it is also more effective than other methods at removing their synonyms and related terms. This justifies the claim that, by producing a single query vector for "a NOT b", we remove not only unwanted strings but also unwanted meanings.

We describe the underlying motivation behind this model and define the vector negation and disjunction operations in Section 2. In Section 3 we review other ways negation is implemented in Information Retrieval, comparing and contrasting with vector negation. In Section 4 we describe experiments demonstrating the benefits and drawbacks of vector negation compared with two other methods for negation.

2 Negation and Disjunction in Vector Spaces

In this section we use well-known linear algebra to define vector negation in terms of orthogonality and disjunction as the linear sum of subspaces. The mathematical apparatus is covered in greater detail in (Widdows and Peters, 2003). If A is a set (in some universe of discourse U), then 'NOT A' corresponds to the complement $A^\perp$ of the set A in U (by definition). By a simple analogy, let A be a vector subspace of a vector space V (equipped with a scalar product). Then the concept 'NOT A' should correspond to the *orthogonal* complement $A^\perp$ of A under the scalar product (Birkhoff and von Neumann, 1936, §6). If we think of a basis for V as a set of features, this says that 'NOT A' refers to the subspace of V which has no features in common with A.

We make the following definitions. Let V be a (real) vector space equipped with a scalar product. We will use the notation $A \leq V$ to mean "A is a vector subspace of V." For $A \leq V$, define the orthogonal subspace $A^\perp$ to be the subspace

$$A^\perp \equiv \{v \in V : \forall a \in A, a \cdot v = 0\}.$$

For the purposes of modelling word-meanings, we might think of 'orthogonal' as a model for 'completely unrelated' (having similarity score zero). This makes perfect sense for information retrieval, where we assume (for example) that if two words never occur in the same document then they have no features in common.

Definition 1 *Let $a, b \in V$ and $A, B \leq V$. By* NOT A *we mean $A^\perp$ and by* NOT a, *we mean $\langle a \rangle^\perp$, where $\langle a \rangle = \{\lambda a : \lambda \in \mathbb{R}\}$ is the 1-dimensional subspace subspace generated by a. By a* NOT B *we mean the projection of a onto $B^\perp$ and by a* NOT b *we mean the projection of a onto $\langle b \rangle^\perp$.*

We now show how to use these notions to perform calculations with individual term or query vectors in a form which is simple to program and efficient to run.

Theorem 1 *Let $a, b \in V$. Then a* NOT b *is represented by the vector*

$$a \text{ NOT } b \equiv a - \frac{a \cdot b}{|b|^2} b.$$

where $|b|^2 = b \cdot b$ is the modulus of b.

Proof. A simple proof is given in (Widdows and Peters, 2003). ∎

For normalised vectors, Theorem 1 takes the particularly simple form

$$a \text{ NOT } b = a - (a \cdot b)b, \tag{1}$$

which in practice is then renormalised for consistency. One computational benefit is that Theorem 1 gives a single vector for a NOT b, so finding the similarity between any other vector and a NOT b is just a single scalar product computation.

Disjunction is also simple to envisage, the expression b_1 OR ... OR b_n being modelled by the subspace

$$B = \{\lambda_1 b_1 + \ldots + \lambda_n b_n : \lambda_i \in \mathbb{R}\}.$$

Theoretical motivation for this formulation can be found in (Birkhoff and von Neumann, 1936, §1,§6) and (Widdows and Peters, 2003): for example, B is the smallest *subspace* of V which contains the set $\{b_j\}$.

Computing the similarity between a vector a and this subspace B is computationally more expensive than for the negation of Theorem 1, because the scalar product of a with (up to) n vectors in an orthogonal basis for B must be computed. Thus the gain we get by comparing each document with the query a NOT b using only one scalar product operation is absent for disjunction.

However, this benefit is regained in the case of *negated* disjunction. Suppose we negate not only one argument but several. If a user specifies that they want documents related to a but not $b_1, b_2, \ldots, b_n$, then (unless otherwise stated) it is clear that they only want documents related to *none* of the unwanted terms b_i (rather than, say, the average of these terms).

This motivates a process which can be thought of as a vector formulation of the classical de Morgan equivalence $\sim a \wedge \sim b \equiv \sim (a \vee b)$, by which the expression

$$a \text{ AND NOT } b_1 \text{ AND NOT } b_2 \ldots \text{ AND NOT } b_n$$

is translated to

$$a \text{ NOT } (b_1 \text{ OR } \ldots \text{ OR } b_n). \qquad (2)$$

Using Definition 1, this expression can be modelled with a unique vector which is orthogonal to *all* of the unwanted arguments $\{b_1\}$. However, unless the vectors $b_1, \ldots, b_n$ are orthogonal (or identical), we need to obtain an orthogonal basis for the subspace b_1 OR $\ldots$ OR b_n before we can implement a higher-dimensional version of Theorem 1. This is because the projection operators involved are in general non-commutative, one of the hallmark differences between Boolean and quantum logic.

In this way vector negation generates a meaning-vector which takes into account the similarities and differences between the negative terms. A query for

chip NOT computer, silicon

is treated differently from a query for

chip NOT computer, potato.

Vector negation is capable of realising that for the first query, the two negative terms are referring to the same general topic area, but in the second case the task is to remove radically different meanings from the query. This technique has been used to remove several meanings from a query iteratively, allowing a user to 'home in on' the desired meaning by systematically pruning away unwanted features.

2.1 Initial experiments modelling word-senses

Our first experiments with vector negation were to determine whether the negation operator could find different senses of ambiguous words by negating a word closely related to one of the meanings. A vector space model was built using Latent Semantic Analysis, similar to the systems of (Landauer and Dumais, 1997; Schütze, 1998). The effect of LSA is to increase linear dependency between terms, and for this reason it is likely that LSA is a crucial step in our approach. Terms were indexed depending on their co-occurrence with 1000 frequent "content-bearing words" in a 15 word context-window, giving each term 1000 coordinates. This was reduced to 100 dimensions using singular value decomposition. Later on, document vectors were assigned in the usual manner by summation of term vectors using *tf-idf* weighting (Salton and McGill, 1983, p. 121). Vectors were normalised, so that the standard (Euclidean) scalar product and cosine similarity coincided. This scalar product was used as a measure of term-term and term-document similarity throughout our experiments. This method was used because it has been found to be effective at producing good term-term similarities for word-sense disambiguation (Schütze, 1998) and automatic lexical acquisition (Widdows, 2003), and these similarities were used to generate interesting queries and to judge the effectiveness of different forms of negation. More details on the building of this vector space model can be found in (Widdows, 2003; Widdows and Peters, 2003).

suit		suit NOT lawsuit	
suit	1.000000	pants	0.810573
lawsuit	0.868791	shirt	0.807780
suits	0.807798	jacket	0.795674
plaintiff	0.717156	silk	0.781623
sued	0.706158	dress	0.778841
plaintiffs	0.697506	trousers	0.771312
suing	0.674661	sweater	0.765677
lawsuits	0.664649	wearing	0.764283
damages	0.660513	satin	0.761530
filed	0.655072	plaid	0.755880
behalf	0.650374	lace	0.755510
appeal	0.608732	worn	0.755260

Terms related to 'suit NOT lawsuit' (NYT data)

play		play NOT game	
play	1.000000	play	0.779183
playing	0.773676	playing	0.658680
plays	0.699858	role	0.594148
played	0.684860	plays	0.581623
game	0.626796	versatility	0.485053
offensively	0.597609	played	0.479669
defensively	0.546795	roles	0.470640
preseason	0.544166	solos	0.448625
midfield	0.540720	lalas	0.442326
role	0.535318	onstage	0.438302
tempo	0.504522	piano	0.438175
score	0.475698	tyrone	0.437917

Terms related to 'play NOT game' (NYT data)

Table 1: First experiments with negation and word-senses

Two early results using negation to find senses of ambiguous words are given in Table 1, showing that vector negation is very effective for removing the 'legal' meaning from the word *suit* and the 'sporting' meaning from the word *play*, leaving respectively the 'clothing' and 'performance' meanings. Note that removing a particular word also removes concepts related to the negated word. This gives credence to the claim that our mathematical model is removing the *meaning* of a word, rather than just a string of characters. This encouraged us to set up a larger scale experiment to test this hypothesis, which is described in Section 4.

3 Other forms of Negation in IR

There have been rigourous studies of Boolean operators for information retrieval, including the p-norms of Salton et al. (1983) and the matrix forms of Turtle and Croft (1989), which have focussed particularly on mathematical expressions for conjunction and disjunction. However, typical forms of negation (such as NOT $p = 1-p$) have not taken into account the relationship between the negated argument and the rest of the query.

Negation has been used in two main forms in IR systems: for the removal of unwanted documents after retrieval and for negative relevance feedback. We describe these methods and compare them with vector negation.

3.1 Negation by filtering results after retrieval

A traditional Boolean search for documents related to the query a NOT b would return simply those documents which contain the term a and do not contain the term b. More formally, let D be the document collection and let $D_i \subset D$ be the subset of documents containing the term i. Then the results to the Boolean query for a NOT b would be the set $D_a \cap D'_b$, where D'_b is the complement of D_b in D. Variants of this are used within a vector model, by using vector retrieval to retrieve a (ranked) set of relevant documents and then 'throwing away' documents containing the unwanted terms (Salton and McGill, 1983, p. 26). This paper will refer to such methods under the general heading of 'post-retrieval filtering'.

There are at least three reasons for preferring vector negation to post-retrieval filtering. Firstly, post-retrieval filtering is not very principled and is subject to error: for example, it would remove a long document containing only one instance of the unwanted term.

One might argue here that if a document containing unwanted terms is given a 'negative-score' rather than just disqualified, this problem is avoided. This would leaves us considering a combined score,

$$\mathrm{sim}(d, a \text{ NOT } b) = d \cdot a - \lambda d \cdot b$$

for some parameter λ. However, since this is the same as $d \cdot (a - \lambda b)$, it is computationally more efficient to treat $a - \lambda b$ as a single vector. This is exactly what vector negation accomplishes, and also determines a suitable value of λ from a and b. Thus a second benefit for vector negation is that it produces a combined vector for a NOT b which enables the relevance score of each document to be computed using just one scalar product operation.

The third gain is that vector retrieval proves to be better at removing not only an unwanted term but also its synonyms and related words (see Section 4), which is clearly desirable if we wish to remove not only a string of characters but the meaning represented by this string.

3.2 Negative relevance feedback

Relevance feedback has been shown to improve retrieval (Salton and Buckley, 1990). In this process, documents judged to be relevant have (some multiple of) their document vector added to the query: documents judged to be non-relevant have (some multiple of) their document vector subtracted from the query, producing a new query according to the formula

$$Q_{i+1} = \alpha Q_i + \beta \sum_{rel} \frac{D_i}{|D_i|} - \gamma \sum_{nonrel} \frac{D_i}{|D_i|},$$

where Q_i is the i^{th} query vector, D_i is the set of documents returned by Q_i which has been partitioned into relevant and non-relevant subsets, and $\alpha, \beta, \gamma \in \mathbb{R}$ are constants. Salton and Buckley (1990) report best results using $\beta = 0.75$ and $\gamma = 0.25$.

The positive feedback part of this process has become standard in many search engines with options such as "More documents like this" or "Similar pages". The subtraction option (called 'negative relevance feedback') is much rarer. A widely held opinion is that that negative feedback is liable to *harm* retrieval, because it may move the query away from relevant as well as non-relevant documents (Kowalski, 1997, p. 160).

The concepts behind negative relevance feedback are discussed instructively by Dunlop (1997). Negative relevance feedback introduces the idea of subtracting an unwanted vector from a query, but gives no general method for deciding "how much to subtract". We shall refer to such methods as 'Constant Subtraction'. Dunlop (1997, p. 139) gives an analysis which leads to a very intuitive reason for preferring vector negation over constant subtraction. If a user removes an unwanted term which the model deems to be closely related to the desired term, this should have a strong effect, because there is a significant 'difference of opinion' between the user and the model. (From an even more informal point of view, why would anyone take the trouble to remove a meaning that isn't there anyway?). With any kind of constant subtraction, however, the removal of distant points has a greater effect on the final query-statement than the removal of nearby points.

Vector negation corrects this intuitive mismatch. Recall from Equation 1 that (using normalised vectors for simplicity) the vector a NOT b is given by $a - (a \cdot b)b$. The similarity of a with a NOT b is therefore

$$a \cdot (a - (a \cdot b)b) = 1 - (a \cdot b)^2.$$

The closer a and b are, the greater the $(a \cdot b)^2$ factor becomes, so the similarity of a with a NOT b becomes *smaller* the closer a is to b. This coincides exactly with Dunlop's intuitive view: removing a concept which in the model is very close to the original query has a large effect on the outcome. Negative relevance feedback introduces the idea of subtracting an unwanted vector from a query, but gives no general method for deciding 'how much to subtract'. We shall refer to such methods as 'Constant Subtraction'.

4 Evaluation and Results

This section describes experiments which compare the three methods of negation described above (post-retrieval filtering, constant subtraction and vector negation) with the baseline alternative of no negation at all. The experiments were carried out using the vector space model described in Section 2.1.

To judge the effectiveness of different methods at removing unwanted meanings, with a large number of queries, we made the following assumptions. A document which is relevant to the meaning of 'term a NOT term b' should contain as many references to term a and as few references to term b as possible. Close neighbours and synonyms of term b are undesirable as well, since if they occur the document in question is likely to be related to the negated term even if the negated term itself does not appear.

4.1 Queries and results for negating single and multiple terms

1200 queries of the form 'term a NOT term b' were generated for 3 different document collections. The terms chosen were the 100 most frequently occurring (non-stop) words in the collection, 100 mid-frequency words (the 1001^{st} to 1100^{th} most frequent), and 100 low-frequency words (the 5001^{st} to 5100^{th} most frequent). The nearest neighbour (word with highest cosine similarity) to each positive term was taken to be the negated term. (This assumes that a user is most likely to want to remove a meaning closely related to the positive term: there is no point in removing unrelated information which would not be retrieved anyway.) In addition, for the 100 most frequent words, an extra retrieval task was performed with the roles of the positive term and the negated term reversed, so that in this case the system was being asked to remove the very most common words in the collection from a query generated by their nearest neighbour. We anticipated that this would be an especially difficult task, and a particularly realistic one, simulating a user who is swamped with information about a 'popular topic' in which they are not interested.[1] The document collections used were from the British National Corpus (published by Oxford University, the textual data consisting of *ca* 90M words, 85K documents), the New York Times News Syndicate (1994-96, from the North American News Text Corpus published by the Linguistic Data Consortium, *ca* 143M words, 370K documents) and the Ohsumed corpus of medical documents (Hersh et al., 1994) (*ca* 40M words, 230K documents).

The 20 documents most relevant to each query were obtained using each of the following four techniques.

- No negation. The query was just the positive term and the negated term was ignored.
- Post-retrieval filtering. After vector retrieval using only the positive term as the query term, documents containing the negated term were eliminated.
- Constant subtraction. Experiments were performed with a variety of subtraction constants. The query a NOT b was thus given the vector $a - \lambda b$ for some $\lambda \in [0, 1]$. The results recorded in this paper were obtained using $\lambda = 0.75$, which gives a direct comparison with vector negation.
- Vector negation, as described in this paper.

For each set of retrieved documents, the following results were counted.

- The relative frequency of the positive term.
- The relative frequency of the negated term.
- The relative frequency of the ten nearest neighbours of the negative term. One slight subtlety here is that the positive term was itself a close

[1]For reasons of space we do not show the retrieval performance on query terms of different frequencies in this paper, though more detailed results are available from the author on request.

neighbour of the negated term: to avoid inconsistency, we took as 'negative neighbours' only those which were closer to the negated term than to the positive term.

- The relative frequency of the synonyms of the negated term, as given by the WordNet database (Fellbaum, 1998). As above, words which were also synonyms of the positive term were discounted. On the whole fewer such synonyms were found in the Ohsumed and NYT documents, which have many medical terms and proper names which are not in WordNet.

Additional experiments were carried out to compare the effectiveness of different forms of negation at removing several unwanted terms. The same 1200 queries were used as above, and the next nearest neighbour was added as a further negative argument. For two negated terms, the post-retrieval filtering process worked by discarding documents containing either of the negative terms. Constant subtraction worked by subtracting a constant multiple of each of the negated terms from the query. Vector negation worked by making the query vector orthogonal to the plane generated by the two negated terms, as in Equation 2.

Results were collected in much the same way as the results for single-argument negation. Occurrences of each of the negated terms were added together, as were occurrences of the neighbours and WordNet synonyms of either of the negated words.

The results of our experiments are collected in Table 2 and summarised in Figure 1. The results for a single negated term demonstrate the following points.

- All forms of negation proved extremely good at removing the unwanted words. This is trivially true for post-retrieval filtering, which works by discarding *any* documents that contain the negated term. It is more interesting that constant subtraction and vector negation performed so well, cutting occurrences of the negated word by 82% and 85% respectively compared with the baseline of no negation.

- On average, using no negation at all retrieved the most positive terms, though not in every case. While this upholds the claim that any form of negation is likely to remove relevant as well as irrelevant results, the damage done was only around 3% for post-retrieval filtering and 25% for constant and vector negation.

- These observations alone would suggest that post-retrieval filtering is the best method for the simple goal of maximising occurrences of the positive term while minimising the occurrences of the negated term. However, vector negation and constant subtraction dramatically outperformed post-retrieval filtering at removing neighbours of the negated terms, and were reliably better at removing WordNet synonyms as well. We believe this to be good evidence that, while post-search filtering is by definition better at removing unwanted strings, the vector methods (either orthogonal or constant subtraction) are much better at removing unwanted meanings. Preliminary observations suggest that in the cases where vector negation retrieves fewer occurrences of the positive term than other methods, the other methods are often retrieving documents that are still related in meaning to the negated term.

- Constant subtraction can give similar results to vector negation on these queries (though the vector negation results are slightly better). This is with queries where the negated term is the closest neighbour of the positive term, and the assumption that the similarity between these pairs is around 0.75 is a reasonable approximation. However, further experiments with a variety of negated arguments chosen at random from a list of neighbours demonstrated that in this more general setting, the flexibility provided by vector negation produced conclusively better results than constant subtraction for any single fixed constant.

In addition, the results for removing multiple negated terms demonstrate the following points.

- Removing another negated term further reduces the retrieval of the positive term for all forms of negation. Constant subtraction is the worst affected, performing noticeably worse than vector negation.

- All three forms of negation still remove many occurrences of the negated term. Vector negation and (trivially) post-search filtering perform as well as they do with a single negated term. However, constant subtraction performs much worse, retrieving more than twice as many unwanted terms as vector negation.

- Post-retrieval filtering was even less effective at removing neighbours of the negated term than with a single negated term. Constant subtraction also performed much less well. Vector negation was by far the best method for removing negative neighbours. The same observation

		1 negated term			**2 negated terms**		
		BNC	**NYT**	**Ohsumed**	**BNC**	**NYT**	**Ohsumed**
No Negation	Positive term	0.53	1.18	2.57	0.53	1.18	2.57
	Negated term	0.37	0.66	1.26	0.45	0.82	1.51
	Negative neighbours	0.49	0.74	0.45	0.69	1.10	0.71
	Negative synonyms	0.24	0.22	0.10	0.42	0.42	0.20
Post-retrieval filtering	Positive term	0.61	1.03	2.51	0.58	0.91	2.35
	Negated term	0	0	0	0	0	0
	Negative neighbours	0.31	0.46	0.39	0.55	0.80	0.67
	Negative synonyms	0.19	0.22	0.10	0.37	0.39	0.37
Constant Subtraction	Positive term	0.52	0.82	1.88	0.42	0.70	1.38
	Negated term	0.09	0.13	0.20	0.18	0.21	0.35
	Negative neighbours	0.08	0.11	0.14	0.30	0.33	0.18
	Negative synonyms	0.19	0.16	0.07	0.33	0.29	0.12
Vector Negation	Positive term	0.50	0.83	1.85	0.45	0.69	1.51
	Negated term	0.08	0.12	0.16	0.08	0.11	0.15
	Negative neighbours	0.10	0.10	0.10	0.17	0.16	0.16
	Negative synonyms	0.18	0.16	0.07	0.31	0.27	0.12

Table 2: Table of results showing the percentage frequency of different terms in retrieved documents

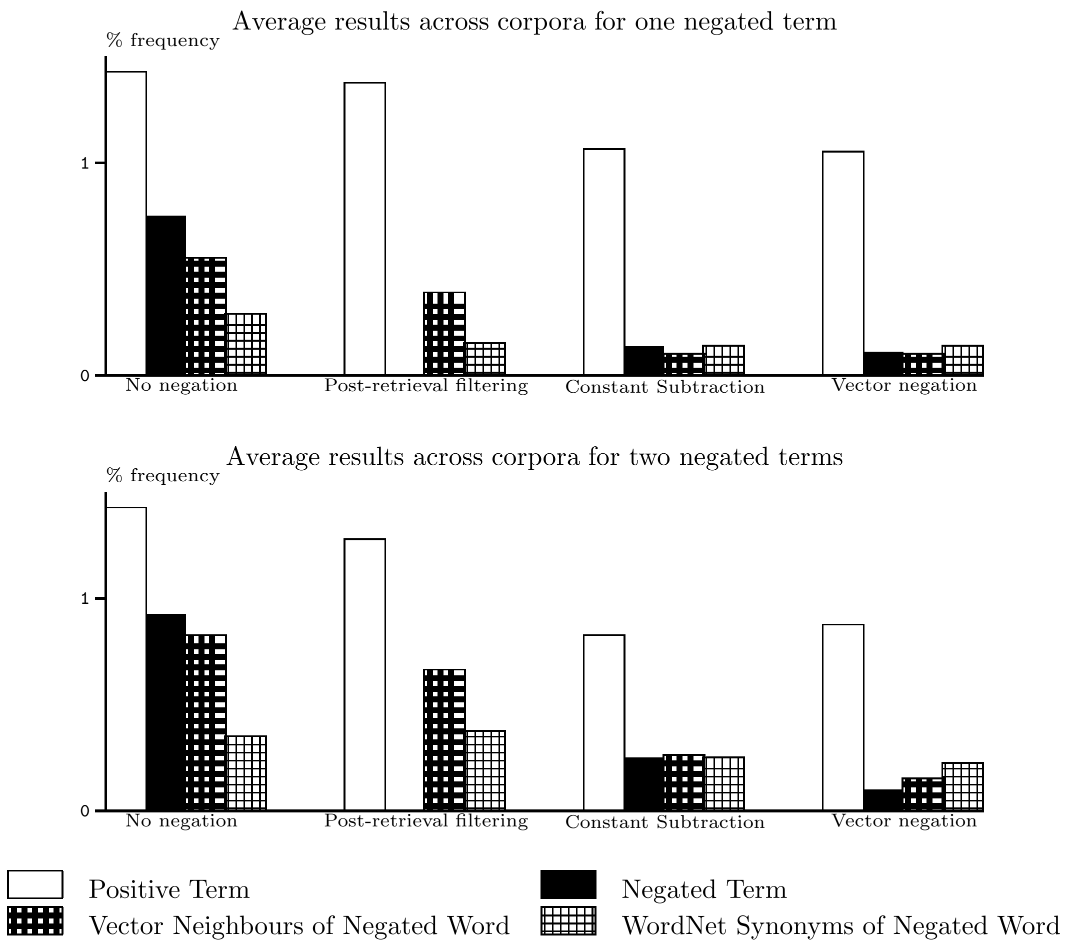

Figure 1: Barcharts summarising results of Table 2

holds for WordNet synonyms, though the results are less pronounced.

This shows that vector negation is capable of removing unwanted terms and their related words from retrieval results, while retaining more occurrences of the original query term than constant subtraction. Vector negation does much better than other methods at removing neighbours and synonyms, and we therefore expect that it is better at removing documents referring to unwanted meanings of ambiguous words. Experiments with sense-tagged data are planned to test this hypothesis.

The goal of these experiments was to evaluate the extent to which the different methods could remove unwanted meanings, which we measured by counting the frequency of unwanted terms and concepts in retrieved documents. This leaves the problems of determining the optimal scope for the negation quantifier for an IR system, and of developing a natural user interface for this process for complex queries. These important challenges are beyond the scope of this paper, but would need to be addressed to incorporate vector negation into a state-of-the-art IR system.

5 Conclusions

Traditional branches of science have exploited the structure inherent in vector spaces and developed rigourous techniques which could contribute to natural language processing. As an example of this potential fertility, we have adapted the negation and disjunction connectives used in quantum logic to the tasks of word-sense discrimination and information retrieval.

Experiments focussing on the use of vector negation to remove individual and multiple terms from queries have shown that this is a powerful and efficient tool for removing both unwanted terms and their related meanings from retrieved documents.

Because it associates a unique vector to each query statement involving negation, the similarity between each document and the query can be calculated using just one scalar product computation, a considerable gain in efficiency over methods which involve some form of post-retrieval filtering.

We hope that these preliminary aspects will be initial gains in developing a concrete and effective system for learning, representing and composing aspects of lexical meaning.

Demonstration

An interactive demonstration of negation for word similarity and document retrieval is publicly available at `http://infomap.stanford.edu/webdemo`.

References

Ricardo Baeza-Yates and Berthier Ribiero-Neto. 1999. *Modern Information Retrieval.* Addison Wesley / ACM Press.

Garrett Birkhoff and John von Neumann. 1936. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843.

Mark Dunlop. 1997. The effect of accessing nonmatching documents on relevance feedback. *ACM Transactions on Information Systems*, 15(2):137–153, April.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge MA.

William Hersh, Chris Buckley, T. J. Leone, and David Hickam. 1994. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual ACM SIGIR Conference*, pages 192–201.

Gerald Kowalski. 1997. *Information retrieval systems: theory and implementation.* Kluwer academic publishers, Norwell, MA.

Thomas Landauer and Susan Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition. *Psychological Review*, 104(2):211–240.

Gerard Salton and Chris Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, 41(4):288–297.

Gerard Salton and Michael McGill. 1983. *Introduction to modern information retrieval.* McGraw-Hill, New York, NY.

Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, November.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Howard Turtle and W. Bruce Croft. 1989. Inference networks for document retrieval. In *Proceedings of the 13th Annual ACM SIGIR Conference*, pages 1–24.

Dominic Widdows and Stanley Peters. 2003. Word vectors and quantum logic. In *Proceedings of the Eighth Mathematics of Language Conference*, Bloomington, Indiana.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. HLT-NAACL, Edmonton, Canada.