Word Vectors and Quantum Logic
Experiments with negation and disjunction

DOMINIC WIDDOWS AND STANLEY PETERS
STANFORD UNIVERSITY

Appeared in Mathematics of Language 8, Bloomington, Indiana, June 2003, pages 141-154

A calculus which combined the flexible geometric structure of vector models with the crisp efficiency of Boolean logic would be extremely beneficial for modelling natural language. With this goal in mind, we present a formulation for logical connectives in vector spaces based on standard linear algebra, giving examples of the use of vector negation to discriminate between different senses of ambiguous words. It turns out that the operators developed in this way are precisely the connectives of quantum logic (Birkhoff and von Neumann, 1936), which to our knowledge have not been exploited before in natural language processing. In quantum logic, arbitrary sets are replaced by linear subspaces of a vector space, and set unions, intersections and complements are replaced by vector sum, intersection and orthogonal complements of subspaces. We demonstrate that these logical connectives (particularly the orthogonal complement for negation) are powerful tools for exploring and analysing word meanings and show distinct advantages over Boolean operators in document retrieval experiments.

This paper is organised as follows. In Section 0.1 we describe some of the ways vectors have been used to represent the meanings of terms and documents in natural language processing, and describe the way the WORD-SPACE used in our later experiments is built automatically from text corpora. In Section 0.2 we define the logical connectives on vector spaces, focussing particularly on negation and disjunction. This introduces the basic material needed to understand the worked

Proceedings of Mathematics of Language 8 2003
R. T. Oehrle & J. Rogers (editors).

examples given in Section 0.3, and the document retrieval experiments described in Section 0.3.1. Section 0.4 gives a much fuller outline of the theory of quantum logic, the natural setting for the operators of Section 0.2. Finally, in Section 0.5, we examine the similarities between quantum logic and WORD-SPACE, asking whether quantum logic is an appropriate framework for modelling word-meanings or if the initial successes we have obtained are mainly coincidental.

To some extent, this paper may have been written backwards, in that the implementation and examples are at the beginning and most of the theory is at the end. This is for two reasons. Firstly, we hoped to make the paper as accessible as possible and were afraid that beginning with an introduction to the full machinery of quantum logic would defeat this goal before the reader has a chance to realise that the techniques and equations used in this work are really quite elementary. Secondly, the link with 'quantum logic' was itself only brought to our attention *after* the bulk of the results in this paper had been obtained, and since this research is very much ongoing, we deemed it appropriate to give an honest account of its history and current state.

0.1 Representing word-meaning in vector spaces

A vector space is a collection of points each of which can be specified by a list of co-ordinates (such as the familiar x-y co-ordinates in Cartesian geometry) (Jänich, 1994, Ch 2), where pairs of points can be added together by adding their co-ordinates, and an individual point can be multiplied by a 'scalar' or number (in this paper, these scalars are real numbers, so all of our vector spaces are 'real' vector spaces). The first linguistic examples of vector spaces were developed for information retrieval (Salton and McGill, 1983), where counting the number of times each word occurs in each document gives a *term-document matrix*, where the i, j^{th} matrix entry records the number of times the word w_i occurs in the document D_j. The rows of this matrix can then be thought of as *word-vectors*. The dimension of this vector space (the number of co-ordinates given to each word) is therefore equal to the number of documents in the collection. *Document vectors* are generated by computing a (weighted) sum of the word-vectors of the words appearing in a given document.

Such techniques are used in information retrieval to measure the similarity between words (or more general query statements) and documents, using a similarity measure such as the cosine of the angle between two vectors (Salton and McGill, 1983, p 121),

$$\mathrm{sim}(w, d) = \frac{\sum w_i d_i}{\sqrt{\sum w_i^2 \sum d_i^2}} = \frac{w \cdot d}{||w||\ ||d||},$$

where w_i, d_i are the co-ordinates of the vectors w and d, $w \cdot d$ is the (Euclidean) scalar product of w and d, and $||w||$ is the norm of the vector w (Jänich, 1994, Ch 8). This calculation is simplified further by normalising all vectors to have unit length, so that the 'cosine similarity' is the same as the Euclidean scalar product. This is a standard technique which (for example) avoids giving too much semantic significance to frequent terms or long documents. Normalised vectors were used in all of the models and experiments described in this paper.

A natural advantage of this structure is that it can be used to define a similarity score between pairs of terms in exactly the same way — two terms will have a high similarity score if they often occur in the same documents, and only seldom occur without one another. In general, several terms are combined into a combined query-statement using commutative vector addition (though the fuzzy-set and p-norm operations of (Salton et al., 1983) give more sophisticated models for conjunction and disjunction which also combine some of the benefits of Boolean and vector approaches).

Typically, such term-document matrices are extremely sparse. The information can be concentrated in a smaller number of dimensions using (among other dimension reduction algorithms) singular value decomposition, projecting each word onto the n-dimensional subspace which gives the best least-squares approximation to the original data. This represents each word using the n most significant 'latent variables', and for this reason this process is called *latent semantic analysis* (Landauer and Dumais, 1997). A variant of latent semantic analysis was developed by Schütze (1998) specifically for the purpose of measuring semantic similarity between words. Instead of using the documents as column labels for the matrix, semantically significant *content-bearing words* are used, and other words in the vocabulary are given a score each time they occur within a context window of (*eg.*) 15 words of one of these content-bearing words. Thus the vector of the word *football* is determined by the fact that it frequently appears near the words *sport* and *play*, *etc.* This method has been found to be well-suited for semantic tasks such as word-sense clustering and disambiguation. Such a vector space where points are used to represent words and concepts is sometimes called a WORD-SPACE (Schütze, 1998). The examples and experiments described in this article use exactly this sort of WORD-SPACE, using the Euclidean scalar product on normalised vectors to compute similarity.

Traditional approaches to semantics using set theory and Boolean logic are well-adapted for arranging primitives into composite propositions but have little to say on the meaning of those primitives[1]. The vector models described in this

[1]Typical analyses (eg. (Partee et al., 1993, Ch 13)) give lambda calculus such as $\lambda x \lambda y.loves(x, y)$ for the meaning of the predicate 'loves', but are content to say that the seman-

section, by contrast, have plenty to say about the meaning of the primitive units, but only limited means to infer the meaning of sentences from these units. We would ideally, of course, have the best of both worlds.

0.2 Logical Connectives in WORD-SPACE

In this section we introduce logical connectives which can be used to explore meanings of terms in WORD-SPACE. In particular, we define negation in terms of orthogonality and disjunction in terms of the vector sum of subspaces. A more thorough discussion of the logic behind these operations is given in Section 0.4.

Vector Negation

We want to model the meaning of a statement like 'rock NOT band' in such a way that the system realises we are interested in the geological, not the musical meaning of the word *rock*. This involves finding which aspects of the meaning of *rock* which are different from, and preferably unrelated to, those of *band*. Meanings are unrelated to one another if they have no features in common at all, just as a document is regarded as completely irrelevant to a user if its scalar product with the user's query is zero — precisely when the query vector and the document vector are orthogonal (Jänich, 1994, §8.2) [2]. Our definition of negation for vectors relies on precisely this correspondence between the notions of 'irrelevant' and 'orthogonal in WORD-SPACE'.

Definition 1 *Two words a and b are considered irrelevant to one another if their vectors are orthogonal, i.e. a and b are mutually irrelevant if $a \cdot b = 0$.*

The statement 'a NOT b' is now interpreted as 'those features of a to which b is irrelevant'.

Definition 2 *Let V be a vector space equipped with a scalar product. For a vector subspace $A \subseteq V$, define the orthogonal subspace $A^\perp$ to be the subspace*

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

Let A and B be subspaces of V. By $\mathrm{NOT}\ B$ we mean $B^\perp$ and by $A\ \mathrm{NOT}\ B$ we mean the projection of A onto $B^\perp$.

Let $a, b \in V$. By $a\ \mathrm{NOT}\ b$ we mean the projection of a onto $\langle b\rangle^\perp$, where $\langle b\rangle$ is the subspace $\{\lambda b : \lambda \in \mathbb{R}\}$.

tics of 'John' is given by `john` or j and the semantics of 'Mary' is given by `mary` or m.

[2] This idea of negation as 'otherness' is found in Plato's *Sophist* dialogue (Horn, 2001, p. 1).

We now show how to use these notions to perform simple calculations with individual vectors in WORD-SPACE, using a standard projection mapping technique (Jänich, 1994, §8.2).

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 norm of b.

Proof. Taking scalar product with b, we have that

$$\begin{aligned} (a \text{ NOT } b) \cdot b &= (a - \frac{a \cdot b}{|b|^2} b) \cdot b \\ &= a \cdot b - \frac{(a \cdot b)\,(b \cdot b)}{b \cdot b} \\ &= 0. \end{aligned}$$

This shows that a NOT b and b are orthogonal, so the vector a NOT b is precisely the part of a which is irrelevant to b (in the sense of Definition 1) as desired. $\blacksquare$

For normalised vectors, Theorem 1 takes the particularly simple form

$$a \text{ NOT } b = a - (a \cdot b) b.$$

In practice this vector is then renormalised for consistency. As well as being well-motivated theoretically, this expression for negation is computationally extremely efficient to implement in the 'search phase' of a retrieval system. In order to find terms or documents that are closely related to a NOT b, it is not necessary to compare each candidate with both a and b and then compute some difference. 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.

Vector Disjunction and Conjunction

Modelling disjunctive expressions (such as A OR B) works similarly. Disjunction in set theory is modelled as the union of sets, which corresponds in linear algebra to the vector sum of subspaces, since $A + B$ is the smallest subspace of V containing both A and B.

Definition 3 *Let $b_1 \ldots b_n \in V$. The expression b_1 OR $\ldots$ OR b_n is represented by the subspace*

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

Finding the similarity between an individual term a and a general subspace B is more complicated than finding the similarity between individual terms. It makes sense to define

$$\mathrm{sim}(a, B) = a \cdot P_B(a) \tag{1}$$

that is, the scalar product of a with the *projection* of a onto the subspace B, since this measures the magnitude of the component of a which lies in the subspace B.

To find this similarity in practice, it is not correct simply to compute $\mathrm{sim}(a, b_j)$ for each of the vectors b_j in turn unless the set $\{b_j\}$ is orthonormal *i.e.* the vectors are pairwise orthogonal and of unit length (Jänich, 1994, p 139) (this is very unlikely). Instead, an orthonormal basis for B must first be constructed, a process which can be accomplished in practice by first using the Gram-Schmidt process (Jänich, 1994, p 142) to obtain an orthonormal basis $\{\tilde{b}_j\}$ for the subspace B. Once this is accomplished, it follows that

$$P_B(a) = \sum_j (a \cdot \tilde{b}_j)\tilde{b}_j$$

so that $\mathrm{sim}(a, B) = \sum_j (a \cdot \tilde{b}_j)$. To compute $\mathrm{sim}(a, B)$ we need to take the scalar product of a with *each* of the vectors $\tilde{b}_j$, so this similarity is more expensive to compute than that given by Theorem 1. Thus the gain we get by comparing each document with the query a NOT b using only one scalar product operation is lost for disjunction, though we show later that this desirable property is recovered for *negated* disjunction.

Just as disjunction makes things more general, we would expect conjunction to make them more specific. Since our underlying WORD-SPACE is homogeneous (in the sense that any two non-zero points can be mapped to each other by a linear transformation), no one point is naturally any more or less general than any other. This is one of the noticeable drawbacks for the basic vector model generally: the terms *plant*, *fruit* and *apple* are all represented by single points without any notion of inclusion or inheritance. Ideally, this problem could be solved by having concepts represented not only by points but also by higher dimensional subspaces. Then *plant*, for example, could refer to a space with *fruit* as a subspace thereof, and with *apple* as an even smaller subspace or point in the *fruit* subspace. In theory it should be possible to build such a space using a taxonomy and corpus-data, though to our knowledge this has not been accomplished. Such a structure would present a natural model for conjunction: the conjunction of two subspaces would simply be their intersection. In the meantime, the intersection of distinct one-dimensional subspaces is always zero, so conjunction in this form is not a useful option.

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'

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'

Table 1: Examples of negation

0.3 Using negation to find word-senses

This section presents initial examples of our vector connectives which demonstrate the uses of vector negation, and of vector disjunction and negation together, to find vectors which represent different senses of ambiguous words. We briefly describe document retrieval experiments which show that vector negation has clear benefits over a traditional Boolean method, as shown in (Widdows, 2003b).

A WORD-SPACE model was built as described in Section 0.1 using the New York Times data from the LDC, a corpus consisting of *ca* 173 million words from news articles written between July 1994 and December 1996. As one might expect, news articles consistently prefer some meanings of ambiguous words over others: for example, the word *suit* is used far more often in a legal context than a clothing context. To test the effectiveness of our negation operator, we tried to find some of the less common meanings by removing words belonging to the more predominant meanings.

Table 1 shows 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.

Vector negation and disjunction can be combined to remove several unwanted areas of meaning simultaneously. Suppose we negate not only one argument but several. If a user states that they want documents related to a but not $b_1, b_2, \ldots, b_n$, then (unless otherwise indicated) 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). In

rock		rock NOT band		rock NOT band, arkansas	
rock	1.000000	rock	0.450473	rock	0.412383
band	0.892790	dubious	0.402324	stands	0.389242
band's	0.868856	arkansas	0.400669	celestial	0.387825
bands	0.867765	ark	0.392304	underground	0.381206
punk	0.861354	madison	0.378165	muck	0.376508
pop	0.848222	celestial	0.376519	touches	0.373402
guitar	0.840769	muck	0.367648	pure	0.373129
tunes	0.837099	sheds	0.363119	wind	0.373017
reggae	0.828602	whitewater	0.362743	echoes	0.360734
acoustic	0.820719	gore	0.360440	explosions	0.356637
blues	0.817073	wind	0.357299	beneath	0.355244
rockers	0.807684	majestic	0.355958	planet	0.354783

The word *rock* is most closely associated with pop music in the New York Times corpus. However, removing these meanings by negating the word *band* leaves a set of associations derived from the town *Little Rock, Arkansas*. (The word *little* is not indexed because it is regarded as too common and general to be a useful search term.) Removing *arkansas* as well gives meanings closely associated to rock as a geological material.

Table 2: Senses of *rock* in the New York Times

this way the expression

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

becomes

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

Using Definition 3 to model the disjunction $b_1 \text{ OR } \ldots \text{ OR } b_n$ as the vector subspace $B = \{\lambda_1 b_1 + \ldots + \lambda_n b_n : \lambda_i \in \mathbb{R}\}$, this expression can be assigned a unique vector which is orthogonal to *all* of the unwanted arguments $\{b_j\}$, this vector being $a - P_B(a)$, where P_B is the projection onto the subspace B just as in Equation 1. It follows that to compute the similarity between any vector and the expression $a \text{ NOT } (b_1 \text{ OR } \ldots \text{ OR } b_n)$ is again a *single* scalar product calculation, which gives the same computational efficiency as Theorem 1. This technique can be used to 'home in on' the desired meaning by systematically pruning away unwanted features (see Table 2).

0.3.1 Experiments with document retrieval

The effectiveness of vector negation and disjunction at removing unwanted concepts has been reliably demonstrated in document retrieval experiments, which are reported in much more detail in (Widdows, 2003b). In order to evaluate the effectiveness of different forms of negation, we used the hypothesis that a query for

$$\text{term } a \text{ NOT term } b$$

should retrieve documents containing many occurences of term a and few occurences of term b. This can be accomplished trivially by first retrieving documents using the query 'term a' and then removing any documents that contain term b, in the traditional Boolean manner (Salton and McGill, 1983, p. 26). However, we also measured the occurence of synonyms and neighbours of the term b. Documents retrieved using vector negation contained far fewer of these than the Boolean method, which we believe to be strong evidence that vector negation removes not only unwanted words but unwanted *areas of meaning*.

0.4 Quantum Logic in Vector Spaces

A development that has recently come to our attention is that the logical operators on WORD-SPACE introduced in section 0.2 are precisely the connectives used in quantum logic. Quantum logic was formally introduced by Birkhoff and von Neumann (1936) as a framework in which to account for the observations and predictions of quantum mechanics, which exhibits some distinctly non-classical behaviour. A famous example is given by the two-slit experiment, in which patterns are observed which can not be accounted for by assuming that an electron must have passed through only one of the two slits (Putnam, 1976, p. 180). A much better approach is to model the emerging electron as a linear combination of states, assuming that the final description receives a contribution from each of the electron's possible routes. Classical logic has problems in this situation, because in set-theory if a is an element of the union $A \cup B$, it follows that at least one of the statements $a \in A$, $a \in B$ must hold — for example, where a represents the state of an electron which has passed through the two slits A and B then one of the statements "a passed through A" or "a passed through B" must hold.

Quantum logic solves this problem by describing the outcomes A and B not as arbitrary sets, but as subspaces of a vector space. Their disjunction is then their vector sum $A+B$, which is strictly larger than their set union $A \cup B$ unless $A \subseteq B$ or $B \subseteq A$. [3] Since there are many points $a \in A + B$ which are neither in A nor in B, the question "which slit did the electron a go through?" ceases to apply. Putnam (1976) contends that the differences between quantum logic and classical logic can account for all of the apparent 'difficulties' of quantum mechanics, and

[3] A simple way to envisage the difference between these two forms of disjunction is to consider the possible trajectories of a point which starts in the centre of a map with the instructions that it can travel in a North-South *or* an East-West direction. If this disjunction is interpreted classically, the particle can only travel to one of a 'cross-shape' of points which are either of the same latitude or of the same longitude as the starting point. In the disjunction is interpreted in the quantum framework, the point can travel anywhere that is a linear combination of north-south or east-west journeys, *i.e.* anywhere on the map.

that we should be prepared to change our view of 'logic' accordingly.

Philosophical issues aside, the structure of quantum logic itself is quite simple and is arrived at precisely by replacing the notions of sets and subsets with those of vector spaces and subspaces (Birkhoff and von Neumann, 1936, §6), (Cohen, 1989; Wilce, 2003; Putnam, 1976, p. 177). Events in quantum mechanics are represented by subspaces of a vector space V. [4] This leads us to consider the collection $L(V)$ of subspaces of vector space V, which is a partially ordered set under the inclusion relation, so that an event A implies an event B precisely when $A \leq B$.

The greatest lower bound or *meet* of $A, B \in L(V)$ is the greatest element $C \in L(V)$ such that $C \subseteq A$ and $C \subseteq B$, which is precisely the intersection $A \cap B$. The least upper bound or *join* of A and B is the smallest $D \in L(V)$ such that $A \subseteq D$ and $B \subseteq D$. However, the set union $A \cup B$ is not in general a member of $L(V)$, and the smallest member of $L(V)$ which contains this set is instead the linear span $A + B$. These two operations give the partially ordered set $L(V)$ the structure of a *lattice* (Birkhoff and von Neumann, 1936, §8), (Birkhoff, 1967), (Cohen, 1989, p. 35). Furthermore, because we are working in a space with an inner product, for each $A \in L(V)$ we can define its (unique) orthogonal complement $A^\perp$ just as in Definition 2. We now have three connectives on the lattice $L(V)$, defined as follows (Birkhoff and von Neumann, 1936, §1, §6) (Putnam, 1976, p. 178):

$$\begin{array}{ll} \text{Conjunction} & A \text{ AND } B = A \cap B \\ \text{Disjunction} & A \text{ OR } B = A + B \\ \text{Negation} & \text{NOT } A = A^\perp \end{array} \tag{3}$$

It is simple to show that these connectives on $L(V)$ satisfy the necessary relations (such as, for example, $A + A^\perp = V$, $A \cap A^\perp = \{0 \in V\}$) to define a *logic* on $L(V)$. (Cohen, 1989, p. 36).

Another important equivalence is that each subspace $A \in L(V)$ can be identified (using the scalar product) with a unique projection map $P_A : V \to A$ (as in Theorem 1), and through this bijection the logic of subspaces $L(V)$ is equivalent to the logic of projection mappings on V. This logic plays a key role in quantum mechanics, where the image $P_A(V)$ of a point $v \in V$ under the projection P_A is used to measure the probability that a particle in the state represented by v will be found to have a physical property represented by the subspace A, using the scalar product $P_A(v) \cdot v$ as a probability measure (Wilce, 2003), just as in Equation 1.

[4] More precisely, quantum mechanics is usually modelled within a Hilbert space, which is a complete inner-product space (Cohen, 1989, 2.18). Every finite dimensional Euclidean space (and so every example of a WORD-SPACE with the Euclidean scalar product) is a Hilbert space, and so to avoid overly technical language we shall continue to talk about vector spaces rather than Hilbert spaces.

Quantum logic differs from classical Boolean logic in (at least) two well-known properties: quantum logic is neither distributive nor commutative. The distributive law

$$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$$

is responsible for the question "which slit did the electron pass through", and so (as described above), quantum logic avoids this issue by avoiding the assumption that the electron *must* have passed entirely through either one of the slits. The commutative property fails because two projection mappings P_A and P_B do not in general commute. (An easy example is to consider the projections onto the x-axis and the line $y = x$ in the plane $\mathbb{R}^2$.) This is used to account for the fact that observations interfere with one another in quantum mechanics, which leads to Heisenberg's famous uncertainty principle (Birkhoff and von Neumann, 1936, §1). Measurements made by the projections P_A and P_B are said to be *compatible* if and only if P_A and P_B commute, which imposes particular conditions on the subspaces A and B (Cohen, 1989, p. 37).

0.5 Quantum Logic and WORD-SPACE — a fluke or a goldmine?

The reason for our interest in quantum logic is that we have already been using the quantum connectives on WORD-SPACE in Sections 0.2 and 0.3: the logical operations defined in Equations 2 and 3 are precisely the negation and disjunction connectives in Equation 3. This gives a much clearer account for some of the observations in Section 0.3. For example, the reformulation of the extended conjunction in Equation 2 follows immediately from knowing that the logic $L(V)$ satisfies the de Morgan laws (Cohen, 1989, p. 37), and it is precisely the non-commutativity of projection operators which forced us to first obtain an orthonormal basis for the subspace $(b_1 \text{ OR } \ldots \text{ OR } b_n)$ in order to implement Equation 1.

This raises the question of whether quantum logic is a desirable framework for natural language semantics, or whether the links between quantum logic and concepts in WORD-SPACE are more accidental. The examples in Section 0.3, and in particular the retrieval experiments outlined in Section 0.3.1, demonstrate that the quantum connectives are at least very useful for manipulating word-meanings.

As models for composition of meaning, Boolean and quantum connectives seem to have different spheres of influence. One intuitive prediction, based on the mathematical models underlying the two frameworks, is that Boolean connectives should be more appropriate for describing discrete entities, and quantum connectives should describe concepts which are more continuous. This prediction is borne out in at least some examples. If one's host for a dinner party said *"Please come at 7 or 7.30"*, one would expect 7.15 also to be a perfectly agreeable time to arrive, which would be false under a Boolean interpretation of *"7 or 7.30"* but true under a quantum interpretation. On the other hand, if upon arrival one was asked *"Would you like an apple or a plum?"* and responded positively, one would not really expect to be given a nectarine on the basis that nectarines are on a scale between apples and plums — here we are talking about discrete objects and it appears that a Boolean interpretation is appropriate. (In practice, there are many other factors to take into account — for example, in many day to day contexts (such as train timetables), a continuous variable becomes 'quantised' and interpretations change accordingly.) This discussion at the very least demonstrates that the differences between Boolean and vector connectives have linguistic significance beyond statistical word sense disambiguation and query generation for information retrieval.

One conceptual problem with the quantum disjunction operator is that in a WORD-SPACE of n-dimensions, n fairly similar concepts could be used to generate the whole space, provided they are linearly independent, leaving the possibility that the 'disjunctions' predicted by quantum logic may become far too general. Another problem with the 'linear span of the arguments' approach to disjunction is that it permits interpolation *and* extrapolation, where extrapolation may be inappropriate. For example, in the *"7 or 7.30"* example, we should not predict that 6 o'clock is also an acceptable arrival time. It follows that a better option might be to interpret a disjunction not as a linear subspace but as a *simplex* by adding the conditions $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$ to Definition 3.

There are many apparent similarities between the historical debate over quantum and classical mechanics on the one hand, and the tension between 'symbolic' and 'statistical' approaches to natural language processing on the other. The vector model for information retrieval was first adopted largely because it allowed for a naturally continuous 'relevance score' rather than a simple dichotomy between relevance and irrelevance, in much the same way that quantum mechanics yields a probability that a particular event will be observed. The possible similarity between finding the 'state' of a particle through measurement and finding the 'sense' of an ambiguous word in context is raised in Widdows (2003a). More generally, quantum mechanics is possibly the single most successful scientific theory for making rigourous, testable predictions about systems where it is known that *exceptions are always a possibility*. That natural language bears the hallmarks of such a system is at the least plausible.

Demonstration

An interactive demonstration of word-similarity and negation in WORD-SPACE is publicly available at `http://infomap.stanford.edu/webdemo`.

Acknowledgements

The authors would particularly like to thank Larry Moss, who first suggested that the vector operators developed in this paper might be linked to quantum logic.

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

Bibliography

Birkhoff, G. (1967). *Lattice theory*. American Mathematical Society, 3rd edition. First edition 1940.

Birkhoff, G. and J. von Neumann (1936). The logic of quantum mechanics. *Annals of Mathematics*, **37**:823–843.

Cohen, D. (1989). *An introduction to Hilbert Spaces and Quantum Logic*. Spriger-Verlag.

Horn, L. (2001). *A Natural history of Negation*. CSLI publications, Stanford.

Jänich, K. (1994). *Linear algebra*. Undergraduate Texts in Mathematics. Springer-Verlag.

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

Partee, B. H., A. ter Meulen, and R. E. Wall (1993). *Mathematical Methods in Linguistics*. Kluwer.

Putnam, H. (1976). The logic of quantum maechanics. In *Mathematics, Matter and Method*, pp. 174–197. Cambridge University Press.

Salton, G., E. A. Fox, and H. Wu (1983). Extended boolean information retrieval. *Communications of the ACM*, **26**(11):1022–1036.

Salton, G. and M. McGill (1983). *Introduction to modern information retrieval*. McGraw-Hill, New York, NY.

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

Widdows, D. (2003a). A mathematical model for context and word-meaning. In *(To appear in) Fourth International and Interdisciplinary Conference on Modeling and Using Context*. Stanford, California.

Widdows, D. (2003b). Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *(To appear in) Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*. Sapporo, Japan.

Wilce, A. (2003). Quantum logic and probability theory. In E. N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Spring 2003 Edition)*.