A Psychologically Inspired Search Engine

Harinder Aujla[1], Randall K. Jamieson[2], and Matthew T. Cook[2]

[1] University of Winnipeg, Winnipeg MB, Canada
h.aujla@uwinnipeg.ca

[2] University of Manitoba, Winnipeg MB, Canada
randy.jamieson@umanitoba.ca
cookm346@myumanitoba.ca

Abstract. For the past 50+ years, psychologists have worked to deduce formal expressions of how people perceive, learn, remember, think, and know. That effort has led to impressive progress for understanding human cognition. However, the theories are rarely applied outside the aim of theoretical debate. In this paper, we present an example of our work that leverages computational expressions of human cognition to build technologies. In particular, we focus on a psychologically-inspired search engine that finds documents by matching the meaning of a search term to the content of documents in the psychological record.

Keywords: Cognitive computing, Document retrieval, Search engine.

1 Introduction

For the past 50+ years, psychologists have worked to deduce formal expressions of how people perceive, learn, remember, think, and know. That effort has led to impressive theoretical progress, but those theories and expressions are rarely applied outside the ceremonies of academic debate.

A number of psychologists have begun to address this shortcoming. For example, Johns et al. (2017) used a model of human semantic memory to predict mild cognitive impairment from verbal behaviour. Rubin, Koyejo, Jones, and Yarkoni (2016) used semantic models to summarize imaging data from the psychological record (see also Rubin et al., 2017). Kwantes, Derbentseva, Lam, Vartanian, and Marmurek (2016) used semantic models to predict personality profiles from essay data. Bedi et al. (2015) used semantic models to predict mental health from verbal reports. Foltz, Laham, and Landauer (1999) used a semantic model to grade undergraduate essays. Brooks's (1991) work on cognitive subsumption architectures advanced robotics. And, of course, artificial neural networks have long served as an important engine in intelligent systems (e.g., LeCun, Bengio, & Hinton, 2015; Rumelhart, Hinton, & McClelland, 1986; Rosenblatt, 1958).

The work that follows contributes to this emerging effort. In specific, we apply a modern psychological theory of human semantic memory to illustrate a psychologically-inspired search engine for document representation and retrieval.

1.1 The Problem

Scientists and scholars often depend on keyword search engines to retrieve information from the published record. In psychology, PsycINFO is the standard example.

PsycINFO is a database maintained by the *American Psychological Association* that catalogs, organizes, and collates published work from over 2,500 scholarly sources in the behavioural and social sciences. All documents are indexed by experts using a hierarchical classification system that includes seventeen topics broken down into basic level and subordinate categories. For example, the topic *Human Experimental Psychology* has six basic level categories (i.e., *Sensory Perception*, *Motor Processes*, *Cognitive Processes*, *Motivation & Emotion*, *Conscious States*, and *Parapsychology*). Basic level categories are broken down into more specific subcategories (e.g., *Cognitive Processes* is broken down into *Learning & Memory* and *Attention*).

Because PsycINFO is proprietary, the company's search algorithms are unpublished. However, the search interface provides some strong clues on its function. For example, one can search using keywords by abstract, author, document title, location, publication title, subject heading, identifier (keywords), and major subject; as well as more technical metadata (e.g., publisher, email, dissertation number).

To a large extent, the PsycINFO search engine works well. But, keyword matching suffers from rational shortcomings. Primarily, keyword matching assumes a simple relationship between a signifier (i.e., a word) and its signified (i.e., the word's meaning). However, that premise is naïve: polysemous words have multiple meanings (e.g., bank) and a word's meaning can change depending on context (e.g., a rough draft versus a rough ride). As a consequence, keyword matching has difficulty finding documents that use different words to express the same idea and misconstrues the relationship between documents that use the same words to express different ideas. Secondly, different research traditions use different vocabularies. For example, linguists and psycholinguists study overlapping problems and phenomena, but they express their knowledge in different ways. To the extent that meaning overlaps where vocabulary differs, keyword search can be blind to those connections. Thirdly, keyword search assumes the user's vocabulary. But, if a user does not already have the vocabulary needed to conduct a search, they can find themselves in a catch 22.

The research field of information retrieval has tried to solve the problem by developing semantically indexed search methods (Bontcheva, Tablan, & Cunningham, 2014). One dominant method is semantic annotation—a process in which humans or machines add semantically oriented metadata to documents. Once a document is semantically annotated, the user can search the semantic metadata for a match. Although the method takes a meaningful step toward a semantically indexed search tool, it still (in the end) relies on keyword search on the semantic metadata. A second method involves representing words as nodes in a graph. Although graph methods go a step further toward developing a framework for semantically indexed search, they are not motivated by psychological concerns or constraints. Therefore, graph methods are unlikely to produce a network of meaning consistent with the one observed in human language.

The question we ask, is can psychology contribute wisdom and methods to the design of semantic indexing and search?

1.2 What does Psychology have to offer?

Psychologists have worked since the 1940s to derive a quantitative representation of word meaning. By historical convention, that effort is thought to have started with Osgood's (1952) work on the semantic differential and with George Miller's work on *Project Grammarama*. In the late 1960s into the 1970s, focus shifted to a representation of word meaning encoded in hierarchical and propositional networks (e.g., Anderson, 2013; Collins & Loftus, 1975; Collins & Quillian, 1969). In the 1980s, psychologists turned to feature-based methods and quantified word meaning based on peoples' introspective ratings about a word's concreteness, imageability, meaningfulness, emotionality, pleasantness, and so on (Gilhooly & Logie, 1980; Friendly, Franklin, Hoffman, & Rubin, 1982; Rubin & Friendly, 1986; Toglia & Battig, 1978). Although all of those methods advanced the field, the analysis was limited in scope and efficiency because they relied on hand coded meaning derived from introspective free report data.

By the 1990s, psychological theories of semantics leapt forward with the invention of vector-space models for meaning. In contrast to prior methods, the vector-space models used machine learning techniques to extract word meaning based on patterns of word use in sizeable text corpora (e.g., months of articles from the New York Times).

Although vector-space models differ in mechanism, all have the same goal of representing each word's meaning by a vector. Once derived, the semantic vectors can be compared, combined, manipulated, and inspected to measure word meanings. If the semantic vectors encode the same word relationships observed in human judgements (e.g., synonymy), the vectors are interpreted as a psychologically valid representation of semantics. If, however, the semantic relationships encoded to the vectors mismatch human judgements, the theory is rejected as a valid psychological account.

Latent Semantic Analysis (LSA) is the canonical vector-space model (Landauer & Dumais, 1997). LSA extracts word meaning from a text database by storing the corpus in a word-by-document matrix that records word frequencies, transforming the counts in the word-by-document matrix into corresponding measurements of entropy, decomposing the transformed word-by-document matrix using singular value decomposition, and re-constructing the matrix in a reduced dimensionality to obtain a semantic vector for each word. Despite LSA's simplicity, it does a remarkable job of tracking human language behaviour including the rate of language acquisition, human vocabulary judgements, word sorting behaviour, free association behaviour, and categorization.

Bound Encoding of the Aggregate Language Environment (BEAGLE) is a second-generation semantic model that was invented to solve LSA's shortcomings (Jones & Mewhort, 2007). BEAGLE operates by deriving a semantic vector for each word in a text corpus based on principles of distributed memory theory (Murdock, 1982, 1983, 1995, 1997; Plate, 1995). The theory improves over LSA in several ways. Firstly, BEAGLE derives a representation of word meaning conditional on word order; LSA does not. Secondly, BEAGLE encodes information related to syntax and grammar; LSA does not. Thirdly, BEAGLE outperforms LSA in both the scope and precision with which it tracks human language behaviour including semantic typicality, categorization, priming, and semantic constraint in sentence completions; in addition to tracking word prediction behaviour and grammatical sentence completion. Finally,

BEAGLE is grounded in established principles of human memory theory and therefore has a deep link to the history of theoretical advances and the empirical database on human cognition and behaviour.

Although LSA and BEAGLE differ in important ways, they converge on the common goal of deriving a psychologically valid vector-space representation of word meaning. To the extent that they predict human language judgements, both theories offer a sound base representation of semantics. But, is that base representation sufficient to support the construction of a useful and psychologically valid semantic search engine?

The work that follows uses BEAGLE as the base technology for a psychologically valid and semantically indexed search engine. We apply the method to the published record from the field of experimental psychology. To illustrate the tool, we will derive semantic vectors for words from the psychological record, use the semantic vectors to summarize those documents, and, evaluate the system in a series of targeted tests. Following that analysis, we provide a brief illustration of how the tool can be used to derive insights into the database on the whole.

2 The System

2.1 Representation

The Corpus. To derive the semantic word vectors, we obtained a corpus of 27,560 documents from the suite of APA experimental psychology journals: *Canadian Journal of Experimental Psychology* (1947-2015), *Journal of Experimental Psychology: General* (1916-2015), *Journal of Experimental Psychology: Animal Learning and Cognition* (1975-2016), *Journal of Experimental Psychology: Applied* (1995-2016), *Journal of Experimental Psychology: Human Perception, and Performance* (1975-2016), *Journal of Experimental Psychology: Learning, Memory, and Cognition* (1975-2016), and *Psychological Review* (1894-2016). The information we used in the simulations that follow included each document's abstract, author names, and keywords.

Deriving Semantic Vectors With BEAGLE. Next, we applied BEAGLE to derive a semantic memory vector for each word in the database.

BEAGLE is a model of lexical semantics. Broadly, it works by "reading" a text corpus and, en route, encoding a memory vector that represents the meaning of each word in that corpus. Mechanistically, the model is expressed in algebra.

At the outset of a simulation, each of the i unique words in the corpus is represented by an environmental vector, e_i. Each environment vector has dimensionality n and each element in an environment vector takes a value randomly sampled from a normal distribution with mean zero and variance $1/n$ (in the simulations that follow, dimensionality was set to n = 1,024). Environment vectors are stable over a simulation and are meant to serve as unique identifiers for the words in the corpus (i.e., the word's orthographic and phonological identity).

Next, the model "reads" the corpus one sentence at a time to build a semantic memory vector for each word. The memory vector for each word, m_i, is composed of two kinds of information: context information and order information. Context information is computed by summing the environmental vectors for all other words in the same sentence (i.e., excluding the word of interest). For example, after reading the sentence, "A dog bit the mailman," the memory vector for *dog* is updated as $m_{dog} = m_{dog} + e_{bit} + e_{mailman}$, the memory vector for *bit* is updated as $m_{bit} = m_{bit} + e_{dog} + e_{mailman}$, and the memory vector for mailman is updated as $m_{mailman} = m_{mailman} + e_{dog} + e_{bit}$.

As should be obvious, summing the environment vectors in this manner causes the memory vectors for all words in the same sentence to grow more similar to one another. Perhaps less obvious, the method also encodes higher-order associations between words. For example, even if *dog* and *beagle* do not co-occur in the same sentence in the corpus, they will be similar to one another by virtue of having common words summed into their representations (e.g., *loyal*).

Order information is computed by encoding information about which words follow one another in a sentence and updating the memory vector with that information. In particular, the first-order association between words (i.e., immediately adjacent words) is encoded using noncommutative circular convolution; hereafter denoted as circular convolution.

Circular convolution is a vector operation that binds two vectors, **x** and **y**, to produce an associative vector, **z**,

$$z_i = \sum_{j=0}^{n-1} x_{j mod n} \times y_{(i-j) mod n} \qquad \{\text{for } i = 0 \text{ to } n-1\} \tag{1}$$

where, n is the dimensionality of **x** and **y**.

A convenient property of circular convolution is that it produces a vector, **z**, that is the same dimensionality as the inputs, **x** and **y**, thereby allowing the association between **x** and **y** to be summed into a single vector.

Of course, there is higher-order sequential information in a sentence (e.g., sequences of three, four, or more words). Thus, to represent second-, third-, and higher-order order information, BEAGLE applies the operation recursively to update a word's order information,

$$o_i = \sum_{j=1}^{p\lambda-(p^2-p)-1} bind_{ij} \tag{2}$$

where, o_i is the order information for word i, p is the position of word i in the sentence, and $bind_{ij}$ is the jth convolution for the word being coded.

To illustrate the operation, the order information for the word dog, o_{dog}, in the sentence, "a dog bit the mailman," is encoded as a sum of the following,

$$\left.\begin{array}{l} bind_{dog,1} = \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \\ bind_{dog,2} = \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \end{array}\right\}\text{Bigrams}$$

$$\left.\begin{array}{l} bind_{dog,3} = e_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \\ bind_{dog,4} = \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \end{array}\right\}\text{Trigrams}$$

$$\left.\begin{array}{l} bind_{dog,5} = \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \\ bind_{dog,6} = \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \circledast \boldsymbol{e}_{mailman} \end{array}\right\}\text{Quadgrams}$$

$$bind_{dog,7} = \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \circledast \boldsymbol{e}_{mailman}\}\text{Tetragram} \tag{3}$$

where $\circledast$ denotes circular convolution and Φ is a universal placeholder used in the computation of order information for every word in every position in every sentence (i.e., constructed as a random vector in the same way as the environment vectors), such that $m_{dog} = m_{dog} + o_{dog}$.

In summary, BEAGLE uses the environment vectors to construct semantic memory vectors that represent the meaning of each word in the corpus as a combination of both context and order information. As the algebra indicates, the theory predicts that a word's meaning will reflect its history of co-occurrence with, and position relative to, other words in sentences. Thus, BEAGLE implements the wisdom from functional linguistics that, "You shall know a word by the company it keeps" (Firth, 1957).

2.2 Building the document summaries

Once we derived the semantic vectors for all 40,517 words in the 27,560 documents in the journal corpus, we used the word vectors to construct a representation for each of the 27,560 documents.

Each document vector was computed as the sum of the word vectors that corresponded to all w words in the document's title, abstract, and keyword list,

$$d_i = \sum_{j=1}^{w} m_j \tag{4}$$

where d_i is the semantic summary of document i, m_j is the semantic memory vector corresponding to word j in document i, and w is the number of words in document i. Once constructed, the document representation was stored into the database of the 27,560 documents.

2.3 Searching the document space

To search the document space, we constructed a query vector, q, that was equal to the sum of the word vectors corresponding to all w words in the query,

$$q = \sum_{j=1}^{w} m_j \tag{5}$$

where, q is the search query, m_j is the semantic memory vector for word j in the search query, and w is the number of words in the search query.

Once computed, the search query, q, was used to search the database and a ranked list of the documents was retrieved. The ranked list was constructed by, first, computing the cosine similarity between q and each of i = 1…27,560 documents in the database,

$$Sim(q, d_i) = \frac{\sum_{j=1}^{n} q_j \times d_j}{\sqrt{\sum_{j=1}^{n} q_j^2}\sqrt{\sum_{j=1}^{n} d_j^2}} \tag{6}$$

where q is the vector representing the search term, d_i is the semantic summary of document i, and n is the dimensionality of the vectors under comparison. Once the similarity of the query to all documents was computed, a ranked list of the 27,560 documents was returned so that the document with the highest cosine similarity to q was ranked first (i.e., rank = 1) and the document with the lowest cosine similarity to q was listed last (i.e., rank = 27,560).

In summary, we derived vector representations of word meaning using BEAGLE, we used the representations to encode documents, and we used a search query to retrieve a ranked list of all documents. We now turn to the problem of evaluation.

3 Assessing the system

At face value, the system is already a psychologically-inspired semantic search engine: it uses a modern theory for human semantic memory to retrieve documents. But, does it work?

Typically, psychological methods and theories are evaluated by their ability to predict human behaviour (i.e., fit by a descriptive criterion) or making error-free decisions (i.e., fit by a prescriptive criterion). Unfortunately, a search engine evades both methods of evaluation. Firstly, people do not perform document retrieval in the same way and, therefore, there is no behavioural profile to fit. Secondly, it is unclear which documents a search engine *ought to* retrieve and, therefore, it is not possible to assess if the documents retrieved are "correct". Nevertheless, there is no value in a system unless it can be shown to be successful in some objective manner. Therefore, to make the evaluation, we conducted a series of simple rational tests.

3.1 Simulation one: Recovering a target document

In simulation one, we asked if the system can recover a target document. To answer the question, we conducted a Monte Carlo study. In each simulation, we sampled a document from the journal database, randomly sampled words from the document's abstract, title, and keywords, constructed a search query from a set of randomly sampled words, queried the database, and recorded the retrieval rank of the target document. To evaluate the system's tolerance, we conducted 1000 simulations for queries composed of 5%, 10%, 25%, 50%, and 100% of the words from the document's abstract, title, and keywords.

To assess the model against some rational controls, we conducted two additional sets of simulations. The first control method repeated the Monte Carlo simulation already

described, but used random vectors to define word meaning and used those random vectors to construct document summaries and queries. To remain as consistent as possible with the simulation using BEAGLE, the random word vectors were constructed in the same way as the environment vectors that were used in the BEAGLE model to construct the semantic word vectors: as randomly generated vectors of dimensionality 1,024 where each element was assigned a random deviate from a normal distribution with mean zero and variance 1/1,024 (see Jones & Mewhort, 2007). The second control method eschewed the vector-based method altogether and constructed the document list based on the number of times each word in the query occurred in a document so that the document retrieved at rank = 1 had the greatest word overlap with the words sampled from the target document and the document retrieved at rank = 27,560 had the least.

Figure 1 shows the median retrieval rank for the target document depending on the percentage of words included in the query. Results with the semantic vectors are shown in red; results with the nonsemantic vectors are shown in blue; and, results with the word match method are shown in grey.

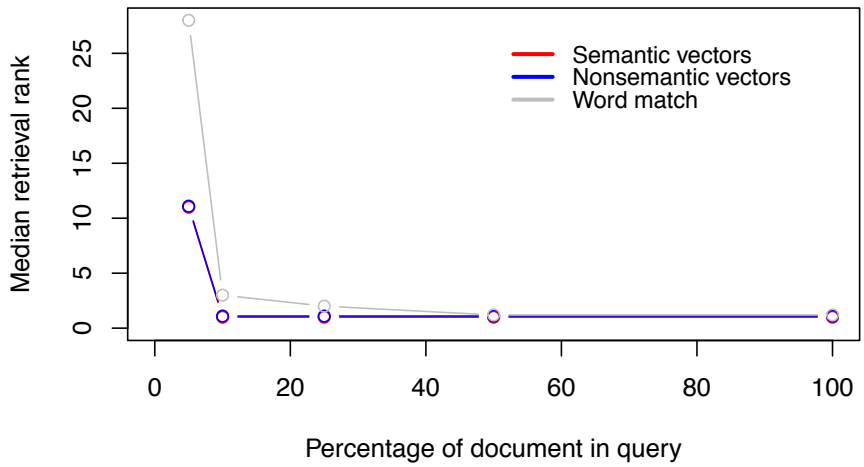

Fig. 1. Simulation one: Target document retrieval using semantic vector, nonsemantic vector, and word match methods. Performance shown as a function of the percentage of words sampled to the search query.

There are three key results in Figure 1. Firstly, all three methods worked very well, performing perfectly in the majority of simulations (i.e., rank = 1) and retrieving the target document no worse than rank 30 out of 27,560 even when only a very small percentage of words was sampled to the query. Secondly, the two vector-based methods (semantic and nonsemantic) worked better than the word match method. Thirdly, the two vector-based methods worked equally well and performed perfectly as long 10% or more of the words in a document were included in the search term.

We conclude that the semantic search method can recover a target document very well and that it is surprisingly tolerant to an incomplete query. However, the equivalent performance using the semantic and nonsemantic methods casts doubt on the value of using semantic vectors at all.

3.2 Simulation two: Comparison of semantic and nonsemantic methods

Simulation one shows that our system can recover a target document. But it shows no advantage of using semantic over nonsemantic vectors.

To test the value of semantics, we re-conducted simulation one. But, we constructed the search query using semantic associates of the words that were randomly sampled from the target document (e.g., the word *memory* was replaced by the word *storage*). In summary, the simulation asks how well a target document is retrieved based on a match to the ideas rather than the words in a query.

Results of the simulation are presented in Figure 2 with the performance using semantic vectors shown in red and performance using nonsemantic vectors shown in blue.

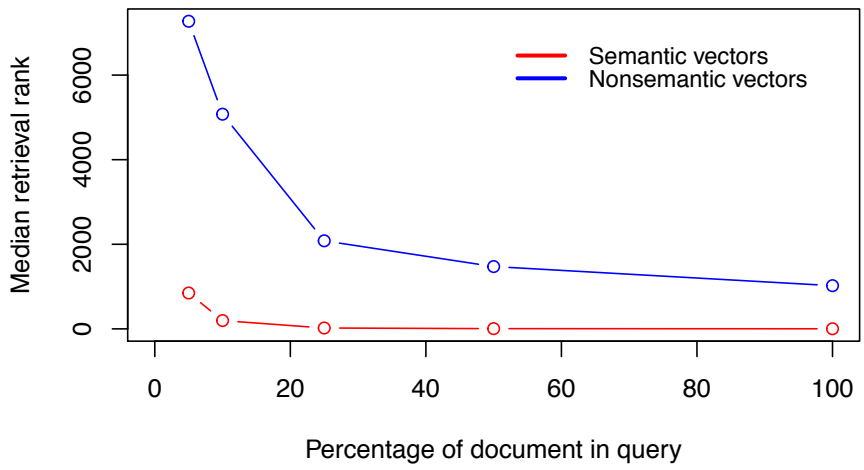

Fig. 2. Simulation two: Target document retrieval for queries constructed with semantic associates using the semantic and nonsemantic vector models. Performance shown as a function of the percentage of words sampled to the search query.

As shown in Figure 2, the semantic method recovers the target document much better than the nonsemantic method. In fact, the nonsemantic method does very poorly, even when all words in the document are used. In contrast, the semantic method does very well unless only a small percentage of words (i.e., less than 10%) is included in the query. However, we offer the excuse that the system still performs well if one considers that the words in the query are sampled at random and, therefore, can include few or no content relevant words.

In summary, when the search query is composed of words from the document abstract, the semantic and nonsemantic vector systems perform equally well. However, when the search query is composed of semantically related words, the semantic method performs much better than the nonsemantic method. Of course, the advantage has very strong practical importance: users should be able to express the intent of their search without the requirement that they use the exact words in the document they are searching for.

3.3 Simulation three: Relationship between semantic and nonsemantic search

Simulation studies one and two provide initial evidence that a document can be recovered better using semantic versus nonsemantic search. However, the results have only expressed the difference in retrieval for a single target document. To address the issue, we conduct a simulation to determine if the document ranking produced by a semantic search matches or mismatches the document ranking produced by nonsemantic and word match retrieval methods.

Simulation three repeated simulation one, but we measured the agreement (i.e., Spearman rank correlation) between the profiles for the semantic, nonsemantic, and word match methods over all 27,560 of the retrieved documents.

The simulation results are presented in Figure 3 (e.g., the agreement between retrieval for the semantic and nonsemantic methods is shown in red). Error bars show one standard error of the mean.

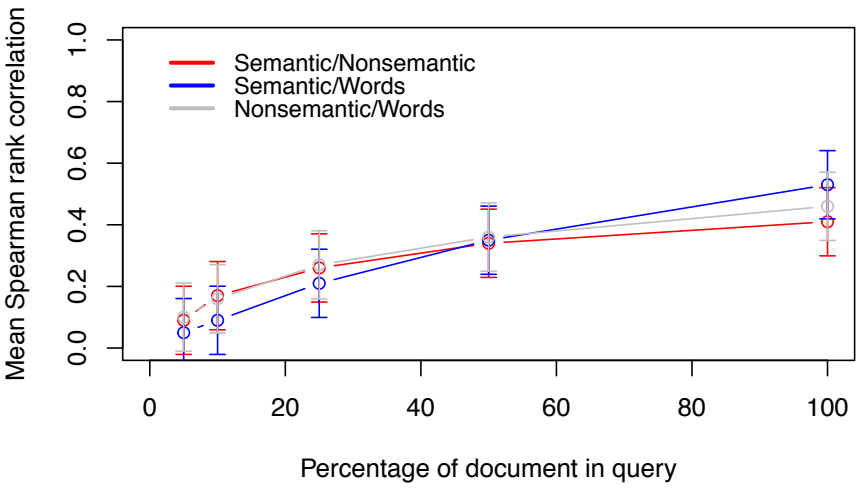

Fig. **3.** Simulation three: Agreement of retrieved lists using the semantic, nonsemantic, and word match methods. Performance shown as a function of the percentage of words sampled to the search query.

There are three key results in Figure 3. Firstly, agreement between the retrieval profiles using the semantic, nonsemantic, and word match methods improved with the

percentage of words included in the search query. Secondly, the strength of agreement was largely consistent between the three methods. Thirdly, there is a curious interaction where the weakest agreement was between the semantic and word match methods when a small percentage of words was included in the query, but the strongest agreement was between those same two methods when a large percentage of words was included in the query.

We conclude that using semantic vectors is not crucial to retrieving a particular target document (see simulation one), but using semantic vectors retrieves a different profile of related documents—even when words are not replaced by semantic associates in the search query. To the extent that the methods disagree on which documents are most and least related to the query (particularly true when the query is short), the semantic method returns different documents than the nonsemantic and word match methods.

3.4 Simulation four: Finding structure in the database

Simulations one, two, and three provide a picture of our system's behavior. However, none of the simulations demonstrates its utility. In simulation four, we show that the theory can find and summarize structure in a database (i.e., in our case, the psychological record).

To illustrate the case, we developed a semantic vector to represent each unique author in the database. An author vector was computed as,

$$a_i = \sum_{j=1}^{d} \sum_{k=1}^{w} m_{jk} \tag{7}$$

where a_i is the semantic summary of all documents written by author i, d is the number of documents published by author i, w is the number of words in document j by author i, and m_{jk} denotes the semantic memory vector for word k in document j published by author i.

Once constructed, we derived a cosine matrix that scored the similarity between all pairs of authors. A full matrix can be obtained by emailing the first author. However, our corpus included 24,215 unique author names and, therefore, it is difficult to show the full matrix of results in any meaningful way.

To compensate, Figure 4 presents a snapshot of the author space including a set of modern Canadian cognitive psychologists. The graph was constructed using multidimensional scaling.

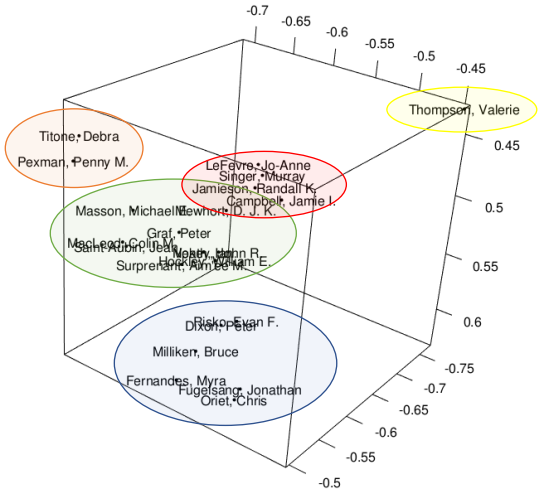

Fig. 4. Simulation four: The author space for a set of current Canadian cognitive psychologists.

The graph shows relationships in three dimensions, where the similarity between authors is represented by their proximity in the space. Given readers are unlikely to know all of the authors, we have overlaid a colour scheme that identifies clusters of researchers who work on related issues. Authors in the blue cloud work on problems associated with attention, authors in the green cloud work on problems related to memory, authors in the orange cloud work on problems related to language, authors in the red cloud work on problems related to reasoning, and the author in the yellow cloud works on decision making. Of course, some of the relationships in the graph reflect co-publication. But, the majority of relationships do not and, therefore, reflect *inferences* drawn by the theory.

Unfortunately, we cannot score the solution. As we have already emphasized, it is not clear how to score the success of the solution given we do not know the prescriptive similarity of authors. However, the illustration makes a more important point.

The method for semantically indexed document retrieval can be applied generally to measure document similarities, author similarities, journal similarities (i.e., the sum of all words published in all papers published in a particular journal), institution similarities (i.e., the sum of all words published in all papers published by all authors at a particular institution), and so on. The method can also be used to compare across the vector categories. For example, one can compute the similarity between a document and the author vectors to retrieve potential manuscript reviewers, between a document and the journal vectors to decide which journal to submit a manuscript to, or between an author and their institution to decide how well they fit into their institutional history. We are currently working to develop and assess these applications.

4 Discussion

By tradition, document retrieval systems are premised on keyword search. However, keyword search is limited in several ways including missed relationships due to shifts in vocabulary as well as the difficulty of finding documents based on those issues.

Researchers working in the discipline of information retrieval have recognized the problem and have focused on solving it. To do so, that field has developed methods for semantic indexing based on techniques like semantic annotation and graph theory.

The work we have presented takes a different approach. We use BEAGLE, a modern psychological theory of human semantic memory, to perform semantic indexing and document retrieval. Because BEAGLE is a theory of semantics, our system is a valid semantically indexed retrieval system. But, the demonstration makes a grander point. To the extent that BEAGLE mimics human language behaviour, our method presents a psychologically-inspired and empirically grounded method for document retrieval that, at least in principle, is more likely to return documents that match the user's intent (see Jones, Kintsch, & Mewhort, 2006; Jones & Mewhort, 2007).

To evaluate the system, we reported a series of simulations. Those simulations showed that the system can recover a target document, that it is tolerant to incomplete search queries, and that it is superior to nonsemantic search. Based on those demonstrations, we are confident that the method works. But, we did not provide an objective measurement about the quality and correctness of the particular documents that were retrieved. The problem is a general one for search engine design: without an objective method to determine which documents *should* be retrieved, there is no way to evaluate if the system was successful in retrieving them.

By tradition, psychological theories are evaluated by one of two general criteria. In a prescriptive analysis, a target is defined for how a system *should* behave. Once defined, the system is evaluated against that criterion: if it matches the prescriptive target, it is a good theory. For example, a theory for classification might be trained on category exemplars. Once it has learned the classification, it is tested on untrained exemplars. Its success is evaluated by the percentage of correct classifications. In a descriptive analysis, measurements are obtained by experimentation and those measurements define how the model system *does* rather than *should* behave. Once defined, the system is evaluated against the descriptive target: if it matches the model system's behaviour, it is a good theory. For example, a theory for classification might be trained on known category exemplars. Once it has learned the classification, it is tested on untrained exemplars and its success is evaluated against the percentage of classifications that are consistent with those given by the model system (e.g., human classifiers trained and tested on the same items).

Although prescriptive and descriptive targets can agree, they often do not. As one illustration, Meyer et al. (2013) reported an experimental study in which 118 physicians correctly diagnosed 55.3% of easy and only 5.8% of difficult cases. The distinction makes our point. Based on Meyer et al.'s example, it makes a good deal of sense to design a diagnostic system that fits the prescriptive rather than descriptive criterion—else, accept a good many diagnostic errors. But, there are cases where a descriptive approach is preferred.

For the field of cognitive computing, the distinction between prescriptive and descriptive approaches is important. In well-defined domains, a prescriptive approach makes good sense—the aim is to produce a behaviour that is consistent with established fact. For example, it is more important in the domain of medical diagnosis to identify a disease correctly. However, in an ill-defined domain where facts are unknown, a descriptive approach makes more sense. For example, it is more important in the domain of document search to retrieve documents that match a user's intent.

But, there is a more fundamental issue that needs excavating. The field of cognitive computing largely focuses on engineering approaches to design cognitive systems. In that tradition, a problem is identified, a goal is defined, and a cognitive system is engineered to meet that goal. The progress in that domain is admirable and the solutions impressive. But, cognition in those systems is often circumscribed to the problem that inspired them and consequently inflexible to uncertainty and new circumstances.

The work we have presented illustrates a different approach to the design of cognitive computing. Our approach uses descriptive theories of intelligence derived from natural systems as a basis for the design of intelligence in artificial systems. Rather than ask how we might design a system to perform semantic indexing, we asked how we could leverage a modern theory of human semantic memory, one already known to mimic human semantic behaviour, to perform semantic indexing and document retrieval.

We see the difference between traditional and psychologically-inspired cognitive computing to be analogous to the difference between traditional and biologically-inspired engineering.

Biological engineers leverage biological principles and examples to solve complex problems. For example, Tero et al. (2010) studied slime molds (*Physarum polycephalum*) and how they develop transportation networks. They used that knowledge to design a computational method to design and optimize human transportation networks (e.g., rail systems). Just as Tero et al. demonstrated that the experimental study of slime molds can produce insights for the design of transportation networks, we argue that the experimental study of human semantic behaviour can produce insights for the design of semantic indexing and document retrieval. In our view, the field of cognitive computing stands to benefit by considering the history of theory and data that the psychological community has produced on the problems of intelligence and cognition.

Although we designed our system in the context of document representation and retrieval. Our true goal is to use that system to design tools for text analysis: our simulation of the author space provides a snap shot of that aim. As we argue in relation to document retrieval, we believe that taking a psychologically-inspired approach to the problem can bring benefits. In particular, when a user wants to understand the structure or find patterns in a database of text, a psychologically-inspired method might be more likely to provide a summary and find patterns that a person is more likely to find meaningful. Although we are only early in the process, our techniques appear to be producing meaningful summaries and output. For instance, the system produces intuitively meaningful clusters of authors and journals. However, the value of the method for producing insights about the global structure of the database remains to be evaluated in an objective and empirical manner.

In summary, the field of document representation and retrieval has typically relied on methods that may or may not represent and evaluate documents in the ways that people do. Yet, all of those systems are intended to serve as cognitive surrogates for the user. To contribute to the ongoing effort, we applied an established and modern theory of human semantic memory to develop what we hope is a psychologically-valid document retrieval system. The system succeeds in a number of objective tests (e.g., retrieves a target document, is tolerant to incomplete queries). However, more work needs to be done to assess the system's function and to evaluate our confidence in the system's descriptive competence.

Acknowledgements. This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada to RKJ and MTC.

References

1. Anderson, J. R. (2013). ACT's propositional network. In *Language, memory, and thought* (pp. 146-181). Psychology Press.
2. Bedi, G., Carrillo, F., Cecchi, G. A., Slezak, D. F., Sigman, M., Mota, N. B., Ribeiro, S., Javitt, D. C., Copelli, M., & Corcoran, C. M. (2015). Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia*, *1*, 15030.
3. Bontcheva, K., Tablan, V., & Cunningham, H. (2014). Semantic search over documents and ontologies. In *Bridging Between Information Retrieval and Databases* (pp. 31-53). Springer Berlin Heidelberg.
4. Brooks, R. (1991). New approaches to robotics. *Science*, *253*, 1227-1232.
5. Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, *82*, 407– 428.
6. Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning & Verbal Behavior*, *8*, 240–247.
7. Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. In Philological Society (Great Britain) (Ed.), *Studies in linguistic analysis*. Oxford, England: Blackwell.
8. Foltz, P. W., Laham, D., & Landauer, T. K. (1999). The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learnin*g, *1*, 939-944.
9. Friendly, M., Franklin, P. E., Hoffman, D., & Rubin, D. C. (1982). The Toronto Word Pool: Norms for imagery, concreteness, orthographic variables, and grammatical usage for 1,080 words. *Behavior Research Methods & Instrumentation*, *14*, 375-399.
10. Gilhooly, K. J., & Logie, R. H. (1980). Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words. *Behavior Research Methods & Instrumentation*, *12*, 395-427.
11. Johns, B. T., Taler, V., Pisoni, D. B., Farlow, M. R., Hake, A. M., Kareken, D. A., ... & Johns, B. (2017). Cognitive Modeling as an Interface Between Brain and Behavior: Measuring the Semantic Decline in Mild Cognitive Impairment. *Canadian journal of experimental psychology*
12. Jones, M. N., & Mewhort, D. J. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, *114*, 1-37.
13. Jones, M. N., Kintsch, W., & Mewhort, D. J. (2006). High-dimensional semantic space accounts of priming. *Journal of memory and language*, *55*, 534-552.

14. Kwantes, P. J., Derbentseva, N., Lam, Q., Vartanian, O., & Marmurek, H. H. (2016). Assessing the Big Five personality traits with latent semantic analysis. *Personality and Individual Differences*, *102*, 229-233.
15. Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*, 211.
16. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436-444.
17. Meyer, A. N., Payne, V. L., Meeks, D. W., Rao, R., & Singh, H. (2013). Physicians' diagnostic accuracy, confidence, and resource requests: a vignette study. *JAMA internal medicine*, *173*, 1952-1958.
18. Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, *89*, 609-626.
19. Murdock, B. B. (1983). A distributed memory model for serial-order information. *Psychological Review*, *90*, 316-338.
20. Murdock, B. B. (1995). Developing TODAM: Three models for serial-order information. *Memory & Cognition*, *23*, 631-645.
21. Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (TODAM2). *Psychological Review*, *104*, 839-862.
22. Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological Review*, *49*, 197–237.
23. Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623-641.
24. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and -organization in the brain. *Psychological review*, *65*, 386-408.
25. Rubin, D. C., & Friendly, M. (1986). Predicting which words get recalled: Measures of free recall, availability, goodness, emotionality, and pronunciability for 925 nouns. Memory & Cognition, 14(1), 79-94.
26. Rubin, T. N., Koyejo, O., Gorgolewski, K. J., Jones, M. N., Poldrack, R. A., & Yarkoni, T. (2016). Decoding brain activity using a large-scale probabilistic functional-anatomical atlas of human cognition. *bioRxiv*, 059618.
27. Rubin, T., Koyejo, O., Jones, M. N., & Yarkoni, T. (2016). Generalized correspondence-LDA models (GC-LDA) for identifying functional regions in the brain. *Advances in Neural Information Processing Systems*.
28. Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, *1*, 45-76.
29. Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, *327*, 439-442.
30. Toglia, M. P., & Battig, W. F. (1978). *Handbook of semantic word norms*. Lawrence Erlbaum.