A method for computing lexical semantic distance using linear functionals

Del Jensen\textsuperscript{a}, Christophe Giraud-Carrier\textsuperscript{b,\ast}, Nathan Davis\textsuperscript{b}

\textsuperscript{a} KJ Nova, Inc., Provo, UT, USA
\textsuperscript{b} Brigham Young University, Provo, UT 84602, USA

Abstract

This paper presents a novel, knowledge-based method for measuring semantic similarity in support of applications aimed at organizing and retrieving relevant textual information. We show how a quantitative context may be established for what is essentially qualitative in nature by effecting a topological transformation of the lexicon into a metric space where distance is well-defined. We illustrate the technique with a simple example and report on promising experimental results with a significant word similarity problem.

Keywords: Semantic distance; Knowledge-based semantics; Topological embedding

1. Introduction

In an era when the amount of textual information stored digitally increases continually, the ability to compare pieces of text (e.g., documents, query strings, abstracts) is central to a number of important applications, such as search, text classification, document clustering, query by example, spam filtering, and RSS feeds aggregation. Not surprisingly, researchers, mainly in Information retrieval (IR) and natural language processing (NLP), have spent much effort in designing algorithms and metrics to capture textual similarity.

IR systems have been developed for the purpose of quickly searching for relevant information from large collections of text. These systems have traditionally utilized syntactic and statistical approaches to improve their effectiveness, such as word indexing and frequency counts (e.g., TF-IDF). NLP systems have been developed for the purpose of automatically organizing and analyzing information that is described by natural human languages. These systems generally rely on syntactic information, as well as statistical and probabilistic models to accomplish such tasks as part-of-speech tagging, text classification and document clustering. IR and NLP techniques are clearly not mutually exclusive and many text-based applications borrow from both fields.

Despite much success, both communities have recognized the need to go beyond syntax and keywords. For example, although the two sentences “guys shooting the hoop in the car park” and “boys playing basketball in the gym” are conceptually very similar, they share no keywords. Conversely, one may use the words “sweetheart, restaurant, private conversation” to describe a romantic encounter, while another may use the same words to describe an attempt at bribing one’s political acquaintance; same keywords, different meaning. Finally, a biologist, with access to a text repository of biological research, might be interested in finding all reports relating to a general topic such as cell division. In this case, the biologist would like to provide a query string such as “cell division” and yet receive conceptually related search results about DNA replication, meiosis, and mitosis, where “cell division” may not be mentioned explicitly. Reaching the correct answer in all of these situations requires semantics and concept-based approaches.

There has been significant work in terms of defining semantic similarity in computational linguistics. From a high-level perspective, this work may be split into corpus-based approaches and knowledge-based approaches. Corpus-based approaches derive similarity measures from lexical and statistical information extracted from text corpora (e.g., frequency counts), while knowledge-based approaches derive similarity measures from lexicons and ontologies (e.g., WordNet, UMLS, GermaNet). The effectiveness of the former depends on the richness of the available corpora, while the effectiveness of the latter depends on the richness of the available ontology. It is difficult, based on
current empirical evidence, to decide which of these approaches superior to the other (if any), and it may well be that hybrid systems, that combine the best-in-class from each approach, are the most promising. Recent work in IR seems to be headed in this direction (e.g., see Refs. [18,10,27,13]).

In this paper, we describe a novel, knowledge-based semantic distance measure, with a formal mathematical foundation, which may be implemented efficiently. Although we do not pursue this idea here, our technique may also be naturally extended with corpus-based information. We focus, here, on showing how a notion of semantic distance between noun-concepts might naturally be derived from a graphical model of hyponymy, and present results which compare our approach to lexical semantic distance with other relevant approaches.

2. Related work

A number of corpus-based semantic measures are surveyed in Ref. [17]. Perhaps the most popular approach to corpus-based semantic similarity is Latent Semantic Analysis (LSA) [6,16,15]. LSA relies on co-occurrence counts of words within documents, and applies a singular value decomposition (SVD) feature dimensionality reduction approach, from which a semantic relatedness measure can be derived. LSA makes the assumption that if words occur together within documents, then they are related in some conceptual way, thus measuring semantic relations of an implicit (or latent) nature. Probabilistic extensions of LSA have recently been proposed (e.g., see Refs. [12,3]).

On the other hand, several knowledge-based semantic measures are described and compared in Ref. [4]. All of these are based on some variation of path length in the WordNet graph. Some also use corpus statistics as additional information, as in Ref. [25], where an information-based function inspired by Resnik’s work [24] and a concept-based function inspired by Rada et al.’s work [23] are discussed. The former function uses the notion of information content, defined as the probability of occurrence in a large text corpus of the first class in the noun hierarchy that subsumes the classes of the words being compared. The latter function uses the sum of heuristically computed edge weights along the shortest path connecting the synsets of the words being compared.

The measure we describe here is knowledge-based. However, instead of using the lexicon directly, it embeds the lexicon in a topological space and computes distances in this transformed space. The work in Ref. [22] is closely related to our own. It describes a conceptual vector model wherein the authors take a set of spanning concepts in a taxonomy/hypernymy model, and define a vector for a general concept as the “activation” of the spanning concepts (e.g., V(door) = (opening[0.3], barrier[0.31], limit[0.32],...)). Vector operations, such as sum (union of semantic properties), product (intersection of semantic properties), angular distance (thematic proximity) and contextualization (context-sensitive property amplification) are then proposed to manipulate terms/concepts. We interpret this approach as the first step we take by projecting onto the basis chains, i.e., a kind of covariant representation. They do not, however, take the next step of shifting to the dual space setting and recasting the representation via a metric tensor. Hence, they are basis dependent and miss out on the well-defined algebra of the dual. In our approach, hyponymy is the conceptual basis and an explicit mapping is constructed.

Although less formal, several proposals have also been made recently to extend traditional search with semantic information. Recent techniques, such as associating query words with results (e.g., see Refs. [2,28,21]), can be viewed as capturing some form of meaning implicitly. Of particular relevance here are those papers that advocate the use of preexisting ontological structure rather than inferring structure from statistical processing of representative corpora. For example, recognizing that “with the Semantic Web, we are already given a large-scale, albeit distributed, explicit semantic structure, constructed independently from the text being searched,” the TAP architecture supports queries directly in the Semantic Web [9]. Anytime a query is entered by the user, it is passed both to the traditional search engine and to the semantic search application, which uses heuristics to navigate the semantic graph and retrieve relevant nodes. Results are returned to the user in two different formats (sub-areas of the results page), one for the traditional search results and one for the semantic search results. A similar but more integrated approach is in Ref. [32], where IR techniques are tightly integrated with formal query and reasoning for effective search, using both textual and semantic information, in semantic portals. The motivation is that IR can help formal query when there is a lack of semantic information in the portal and formal query (and reasoning) can help IR by reducing the scope of retrieval. Yet, another technique, based on the idea of spread activation applied to a semantic model of the domain, is presented in Ref. [26]. Provided a semantic model exists, nodes corresponding to terms in the query are activated and their activation spreads through the network, thus recruiting more nodes/concepts along the way. Hence, concepts not explicitly included in the query may be retrieved and used to enhance the search. Lines of activation may be weighted and constraints are imposed on the activation mechanism to guarantee success. The authors specify that the idea is particularly successful when concepts in the ontology have rich textual information.

Finally, we note that many researchers have leveraged WordNet in an attempt to add semantic information to improve such tasks as contextual analysis and query expansion (e.g., see Refs. [31,30]). In the context of search, the work presented here can also be viewed as an alternative to query expansion, where instead of using standard metrics with semantically close words added explicitly to the query, we would implicitly expand the search with our semantic-based metric.

3. Motivation

The main purpose of a computational semantic model is to establish a quantitative context for that which is essentially qualitative in nature. What we want, therefore, is a framework in which we can map lexical elements to mathematical objects, so that the qualitative or subjective notion of semantic similarity among lexical elements may be replaced by a quantitative or objective distance function defined over the corresponding math-
ematical objects. Such a framework would allow us to manipulate lexical elements via operators in the mathematical space.

To be effective, the mapping from lexical elements to mathematical objects should be such that it preserves what de Saussure termed relational identity within the conceptual field [5]. That is, lexical items that are “close” in meaning correspond to objects that are “close” in a mathematically computable sense, and lexical items that are “not close” correspond to objects that are “far” in a computable sense. In mathematical terms, the ideal semantic mapping should be continuous and one-to-one. Such a mapping is called a homeomorphism, and is a concept that figures prominently in such disciplines as topology [11] and functional analysis [7]. Topology uses language and logic in a precise way in order to formalize, refine and extend the human-intuitive notion of space into a useful body of knowledge.

Here, we borrow ideas from topology to define a homeomorphism that maps a conceptual hierarchy of lexical items with a single semantic relation to a (complete) metric space, i.e., a space where the notion of distance is well-defined. We note, at the onset, that this approach is a departure from most standard work in information extraction and natural language processing, which tend to use techniques from probability theory and statistical learning. The lexical items we focus on are nouns, representing concepts, and the semantic relation is hyponymy. A hyponym is a word or phrase whose semantic range (i.e., meaning) is included within that of another, thus more general, word or phrase. For example, breast cancer is a hyponym of cancer, boy is a hyponym of man, and aspen and oak are hyponyms of tree. Intuitively, one may think of hyponymy as capturing a kind of is-a relationship among words or concepts (e.g., a aspen is a tree, breast cancer is a cancer, etc.).

It is easy to see that the hyponymy relation has the following properties, where \( X \), \( Y \) and \( Z \) are arbitrary words:

- Reflexivity: \( X \) is a hyponym of \( X \).
- Anti-symmetry: if \( X \) is a hyponym of \( Y \) and \( Y \) is a hyponym of \( X \), then \( X \) and \( Y \) must be the same (i.e., \( X = Y \)).
- Transitivity: if \( X \) is a hyponym of \( Y \) and \( Y \) is a hyponym of \( Z \), then \( X \) is a hyponym of \( Y \).

A relation satisfying these properties is called a partial order. Given a set of words, it is possible to arrange them in the form of a graph, where there is a link between two words if one is a hyponym of the other. Since hyponymy is a partial order, the resulting graph takes the form of a lattice, where, among other things:

- All links or edges are directed, as a consequence of anti-symmetry (i.e., hyponymy can only be read one way, from the more specific word to the more general one).
- There are no cycles, other than the trivial ones (“around” each word, via reflexivity).
- There is a maximal element or root, either arising naturally from the set of words under consideration or defined explicitly as such (e.g., in a zoological context, all subjects are hyponyms of the most general element or class animal).

Interestingly, the Cognitive Science Laboratory at Princeton University has created, and is maintaining and distributing, a structured model of English language concepts, known as WordNet, which includes a hyponymy graph [20,19,8]. As is typical in linguistics, WordNet uses base forms or stems, rather than actual words. Thus, for example, “child” and “children” are represented by the same base form “child,” while “go” and “went” and “gone” and “going” are grouped under the same base form “go.” Although our approach is not tied to WordNet, WordNet is used here as a convenient and readily available structure on which that approach can be effectively demonstrated. As stated above, we restrict our attention here to nouns only. Hence, the set \( N \) of (stemmed) nouns together with hyponymy, as found in WordNet, is the directed acyclic graph of interest.

Because hyponymy is a partial order, we can associate each noun, or conceptual element, unambiguously with the contiguous (maximal) chains that connect it to the graph’s root. As an illustration, Fig. 1 shows part of the hyponymy chain structure

---

1 These chains are maximal because they are not proper sub-chains of any other chains.
for the (British English) noun *mistress: a female schoolteacher*. The root here is the general concept *entity*.

Maximal chains such as these play a pivotal role in our model. Indeed, the contiguous chains of the graph can be interpreted as functionals on the graph, that map points of the graph into the real numbers. These mappings are continuous under the order topology, so that “close” concepts in the graph are mapped to “close” real numbers. Now consider that the functions form a vector space, and those elements of the original set of lexical concepts can be interpreted as linear functionals on the function space. In other words, the spaces of conceptual elements and chain-functions are dual, as depicted in Fig. 2. We exploit the duality, representing concept-elements in their conjugate form in the (function) vector space. We adopt an inner product on the function space that is consistent with the original order topology, and use the inner product on the conjugate representations of the concept-elements to define a metric. For simplicity, we leave out most of the mathematical details of the transformation here, and only provide an overview through a simple example in the following section.

4. Building the metric space: an example

Let $c$ be some maximal chain and let $q$ be the length of $c$. Define the function $m_c: c \to \mathbb{R}$ by:

$$m_c(c^i) = \frac{i - 1}{q - 1}$$

where $c^i$ is the $i$th element of $c$, ordered from root to leaf. Now, define the function $f_c: N \to [0, 1]$ by:

$$f_c(n) = m_c(c^n)$$

where $c^n$ is the lowest (towards the leaf) point of intersection of $n$, via some chain containing $n$, with $c$.

Consider the simple directed acyclic graph $G$ of Fig. 3. For such a small graph, we may take the set of all maximal chains (from leaf to root). In practice, larger graphs would make the associated computations unfeasible and we would project onto a contextually relevant subset of chains. We return to this issue in Section 7.

Here, the chains are $c_1 = \{ e, c, a \}$, $c_2 = \{ e, d, a \}$, $c_3 = \{ f, b, a \}$, $c_4 = \{ f, d, a \}$, $c_5 = \{ g, b, a \}$, and $c_6 = \{ g, c, a \}$. The value of $f_{c_4}(e)$ is .5, since the earliest point of intersection of any chain containing $e$ with $c_4$ is halfway down $c_4$. Since no chain containing $g$ intersects $c_4$ at any point except the root, the value of $f_{c_4}(g)$ is 0. The complete action of the functions $\{ f_c \}$ on $G$ is summarized in Fig. 4.

The set of functions $\{ f_c \}$ naturally spans a vector space consisting of all linear combinations of the $f_c$’s. Our task is now simply to find an orthonormal set of functions that provides a basis for the that vector space, and produce a linear transformation from $N$ to the corresponding conjugates in the function space with respect to the basis (hence, an inner-product preserving representation of the original elements of $G$).

In the particular case of the model in Fig. 3, we can employ the Gram–Schmidt algorithm to compute an orthonormal basis $\{ u_i \}$ from $\{ f_c \}$. The Gram–Schmidt process takes a finite, linearly independent set of vectors (i.e., a basis) and produces a new set of orthonormal vectors that span the same space. First, note that $f_6$ is in the span of the other functions, so that $\{ f_1, f_2, f_3, f_4, f_5 \}$ is the linearly independent set of vectors we start from. The algorithm goes as follows.

\[
\begin{align*}
  u_1 &= f_1 \\
  u_1 &= u_1 \\
  u_2 &= f_2 - (f_2, u_1) u_1 \\
  u_3 &= f_3 - (f_3, u_1) u_1 - (f_3, u_2) u_2 \\
  u_4 &= f_4 - (f_4, u_1) u_1 - (f_4, u_2) u_2 - (f_4, u_3) u_3 \\
  u_5 &= f_5 - (f_5, u_1) u_1 - (f_5, u_2) u_2 - (f_5, u_3) u_3 - (f_5, u_4) u_4 \end{align*}
\]

Fig. 4. Action of $\{ f_c \}$ on $G$. 

For a small graph, we may take the set of all maximal chains (from leaf to root). In practice, larger graphs would make the associated computations unfeasible and we would project onto a contextually relevant subset of chains. We return to this issue in Section 7. Here, the chains are $c_1 = \{ e, c, a \}$, $c_2 = \{ e, d, a \}$, $c_3 = \{ f, b, a \}$, $c_4 = \{ f, d, a \}$, $c_5 = \{ g, b, a \}$, and $c_6 = \{ g, c, a \}$. The value of $f_{c_4}(e)$ is .5, since the earliest point of intersection of any chain containing $e$ with $c_4$ is halfway down $c_4$. Since no chain containing $g$ intersects $c_4$ at any point except the root, the value of $f_{c_4}(g)$ is 0. The complete action of the functions $\{ f_c \}$ on $G$ is summarized in Fig. 4.

The set of functions $\{ f_c \}$ naturally spans a vector space consisting of all linear combinations of the $f_c$’s. Our task is now simply to find an orthonormal set of functions that provides a basis for the that vector space, and produce a linear transformation from $N$ to the corresponding conjugates in the function space with respect to the basis (hence, an inner-product preserving representation of the original elements of $G$).

In the particular case of the model in Fig. 3, we can employ the Gram–Schmidt algorithm to compute an orthonormal basis $\{ u_i \}$ from $\{ f_c \}$. The Gram–Schmidt process takes a finite, linearly independent set of vectors (i.e., a basis) and produces a new set of orthonormal vectors that span the same space. First, note that $f_6$ is in the span of the other functions, so that $\{ f_1, f_2, f_3, f_4, f_5 \}$ is the linearly independent set of vectors we start from. The algorithm goes as follows.

\[
\begin{align*}
  u_1 &= f_1 \\
  u_1 &= u_1 \\
  u_2 &= f_2 - (f_2, u_1) u_1 \\
  u_3 &= f_3 - (f_3, u_1) u_1 - (f_3, u_2) u_2 \\
  u_4 &= f_4 - (f_4, u_1) u_1 - (f_4, u_2) u_2 - (f_4, u_3) u_3 \\
  u_5 &= f_5 - (f_5, u_1) u_1 - (f_5, u_2) u_2 - (f_5, u_3) u_3 - (f_5, u_4) u_4 \end{align*}
\]
In matrix form, this gives:

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\langle f_1, w_1 \rangle \\
\langle f_2, w_1 \rangle \\
\langle f_3, w_1 \rangle \\
\langle f_4, w_1 \rangle \\
\langle f_5, w_1 \rangle \\
\end{bmatrix}
\]

The steps deriving \( E \) correspond to the elementary operations \( E_{32}, E_{23} \). Thus we obtain an orthonormal basis \( \{w_i\} \) via the matrix \( Q = \Pi_i E_i \). A quick computation gives:

\[
Q^{-1} = \begin{bmatrix}
1.2247 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.8165 & 0.9129 & 0.0000 & 0.0000 & 0.0000 \\
0.2041 & 0.3651 & 1.1511 & 0.0000 & 0.0000 \\
0.4082 & 1.0042 & 0.4778 & 0.3110 & 0.0000 \\
0.4082 & -0.0913 & 1.0425 & -0.2351 & 0.4277 \\
\end{bmatrix}
\]

From \( Q^{-1} \), we can now derive the transformation \( F \), which gives us a representational form for elements of \( \mathcal{N} \) with respect to an orthonormal basis in the function space, as follows:

\[ F(n) = [f_1(n), f_2(n), f_3(n), f_4(n), f_5(n)]Q^{-1} \]

Finally, we define distance between nouns in \( G \) with the following simple procedure. Let \( n_1 \) and \( n_2 \) be two nouns to be compared.

1. Compute the function vectors \( \vec{n}_1 = (f_1(n_1), f_2(n_1), f_3(n_1), f_4(n_1), f_5(n_1)) \) and \( \vec{n}_2 = (f_1(n_2), f_2(n_2), f_3(n_2), f_4(n_2), f_5(n_2)) \).
2. Compute the transformed vectors \( \vec{\tau}_1 = \vec{n}_1 Q^{-1} \) and \( \vec{\tau}_2 = \vec{n}_2 Q^{-1} \).
3. Compute the direction cosine distance dist\((n_1, n_2)\) = arccos((\( \vec{n}_1 \cdot \vec{n}_2 \))/(\( ||\vec{n}_1|| \cdot ||\vec{n}_2|| \))).

It is easy to see that this approach generalizes to any hyponymy graph and any set of basis chains. In the next section, we illustrate its use on a couple of relatively simple examples based on WordNet.

5. Illustration

For our first example we establish a set of basis chains characterizing people, aircraft, household appliances, and animals. The basis, shown in Fig. 5, consists of 12 concepts, indicated by the first several words of the associated glosses.

These 12 concepts, in turn, ground 17 chains which correspond to 17 functionals. We map the following nouns to our mathematical model:

- princess—(a female member of a royal family other than the queen (especially the daughter of a sovereign));
- female child, girl, little girl—(a youthful female person; "the baby was a girl"); "the girls were just learning to ride a tricycle");
- a young dog
- young domestic cat
- young bird especially of domestic fowl
- infant born at a gestational stage between 37...
- a man who idles about in the lounges
- a clothes dryer that uses centrifugal motion to...
- an iron that was heated by placing it...
- a vacuum coffee maker
- a kitchen appliance for disposing of garbage
- an oven or part of a stove used...
- a kitchen appliance (usually electric) for toasting bread
- a large jet plane that carries passengers

Fig. 5. Basis List #1.
Fig. 6. Mapping of selected nouns onto basis List #1.

Fig. 7. Visual representation of distances. (a) Projection on basis List #1; (b) projection on basis List #2.

- male child, boy—(a youthful male person; "the baby was a boy"; "she made the boy brush his teeth every night"; "most soldiers are only boys in uniform");
- milkman—(someone who delivers milk);
- woman, adult female—(an adult female person (as opposed to a man); "the woman kept house while the man hunted");
- stove, kitchen stove, range, kitchen range, cooking stove—(a kitchen appliance used for cooking food; "dinner was already on the stove");
- toaster—(a kitchen appliance (usually electric) for toasting bread);
- wringer—(a clothes dryer consisting of two roles between which the wet clothes are squeezed);
- biplane—(old fashioned airplane; has two wings one above the other);
- blimp, sausage balloon, sausage—(a small nonrigid airship used for observation or as a barrage balloon);
- chicken, Gallus gallus—(a domestic fowl bred for flesh or eggs; believed to have been developed from the red jungle fowl);
- duck—(small wild or domesticated web-footed broad-billed swimming bird usually having a depressed body and short legs).

The distance matrix resulting from this mapping is shown in Fig. 6 ("#UF" represents an underflow condition and can be interpreted as 0) and a visual representation is given in Fig. 7 (a).

It might seem odd that no distinction is made between girl, boy and princess (see the flat area at distance 0 in the left-hand corner of Fig. 7(a)). This situation begs the following question: What if we provide more basis chains concerning people? In Fig. 8 we provide just such a basis, with 14 elements for 23 chains. A compelling visual representation of the new projection is shown in Fig. 7(b).

This is all very encouraging. However, a close inspection of some of the distances, as shown in Fig. 9, presents us with an important feature: the metric is profoundly influenced by (in fact founded upon) the graph topology. Indeed, this result is fundamentally the whole point of the discussion.

How can it be that princess, being female, is conceptually as close to boy as it is to girl (let alone the lack of distinction from

- a young dog
- young domestic cat
- young bird especially of domestic fowl
- infant born at a gestational stage between 37...
- a man who idles about in the lounges
- a clothes dryer that uses centrifugal motion to...
- an iron that was heated by placing it...
- a vacuum coffee maker
- a kitchen appliance for disposing of garbage
- an oven or part of a stove used...
- a kitchen appliance (usually electric) for toasting bread
- a large jet plane that carries passengers
- a boy attending school
- a girl attending school

Fig. 8. Basis List #2.
To answer this question, we need to take a closer look at the WordNet topology. Consider the chain structures for girl and princess, as shown in Fig. 10.

The chain structure for schoolgirl, which is one of our basis elements, is as the chain structure for girl, with schoolgirl appended (subordinate to) girl. Note that girl, sparse as its structure is, is at least subordinate to female, whereas princess is not! And indeed, the chain structure for boy is identical to that for girl, with the exception of the "female" node being replaced by a "male" node, thus no distinction can be made between them relative to princess. We make an attempt at addressing this problem by editing the topology for princess (see dashed lines and shaded node in Fig. 10). This, in turn, yields the (more satisfying) projection result shown in Fig. 11.

In another example using WordNet 1.6 (not shown), we found that girl was closer to the idea of a young pig than to the idea of a male hog. Of course, both piglet and girl are young of the corresponding species, and the female aspect of girl could only serve to distance the noun from the intrinsic maleness boar. But such was not the case with colt and filly? Should not their distances from girl reflect similar underlying relationships? And why would girl map closer to pigs than horses (as it did in this case)? Again, the answer was in the topology. Although the chain structure for filly included the notion of “young female horse,” there was nothing in the graph to indicate that filly was a female animal. Indeed, there was nothing to indicate that a filly was a horse!

In almost every case, we find that any such peculiarities in distances reflect corresponding “errors” or omissions in the topology. Of course, what may reflect an irregularity in the topology for one person may be another’s dearly held prejudice. Whatever the situation, it may be argued that an incomplete knowledge model leads to confused/biased thinking: something one can occasionally see in the actions of people. These examples, however, lend credence to an important aspect of our model we mentioned above. We propose that graphs are relatively easy to change, and appropriate modification of a “semantic” graph, whether by pruning or by extension, is effectively an act of learning. We further conjecture that an effective semantic algebra will be key to whatever mechanism informs a self-modifying process on the graph. In any case, the fact that, under a given projection,
one might expect to find distinctions among conceptual entities that are not otherwise manifest may help inform some scheme for editing the relation, either manually or automatically.

6. Experimental results

To validate the effectiveness of our approach, we now report on its application to Rubenstein and Goodenough’s 65 noun pairs [29]. In their study, each noun pair was given a semantic relatedness score ranging from 0.0 (semantically unrelated) to 4.0 (synonym) by 51 independent human subjects. The scores were then aggregated and the 65 pairs ranked by increasing similarity score. The resulting ranked list provides a kind of gold standard for computational semantic metrics. Indeed, given a semantic similarity metric $m$, one can quantify the value of $m$, as well as compare it to any other metric $m'$, by computing the correlation between the ranking obtained by $m$ (respectively, $m'$) and Rubenstein and Goodenough’s human-generated ranking. The higher the correlation the more accurate the metric.

In order to conduct our experiment, we implemented a word-to-word comparison system in Java (version 1.5.0_09), using WordNet (version 2.1) as our lexical structure. We interface with WordNet via the Java WordNet Library (JWNL). We use a slightly modified version of JWN L 1.3 adapted to WordNet 2.1 (JWN L 1.3 is designed to interface with WordNet 2.0 by default). We deployed the system as a local web-based service, with a simple user interface designed to enable users to compute word-to-word distance measures.

Recall that in order to compute a linear transformation that allows a mapping into our metric space, we must select a spanning set of maximal basis chains. For simplicity, we selected the set of maximal chains generated by the entries in WordNet for the first sense of each of the unique words in the 65 noun pairs. This resulted in 62 maximal basis chains. For each noun pair $(w_1, w_2)$, our system maps $w_1$ and $w_2$ into the metric space and computes the distance between them. For polysemous words, we choose the dominant sense for each word, or, where applicable, we choose mutually triggering senses (e.g., bird and crane).

The distance measure $\text{dist}(w_1, w_2)$ computed by our system (rounded to 2 decimal places), for each of the 65 pairs, alongside the human scores, is shown in Table 1. The correlation coefficient between the human and the semantic distance rankings is $-0.802$.\(^2\)

\(^2\) The negative sign on the correlation coefficient reflects the fact that the human scores measure similarity, while our metric’s scores measure distance.
For comparison purposes, we show how our distance measure fares, on the same task, against 5 other popular knowledge-based semantic measures surveyed and evaluated in Ref. [4]. A ranked list of these measures and ours, from most correlated to least correlated (with the gold standard), is given in Table 2. As an additional element of comparison, we note that for this task, a standard implementation of Latent Semantic Analysis [14], a popular corpus-based semantic measure, with 300 factors and a corpus consisting of general readings (up to first year college level), gives rise to a correlation coefficient of 0.644.

While the performance of our approach comes in third place on this task, our distance computation presents distinct advantages with respect to practical usability. Recall that most other knowledge-based similarity metrics operate directly on the knowledge structure, and generally use variations of the path length between words (or stems). This process can be time consuming in a practical application, and would thus need to be precomputed to be used efficiently. Consider the case of precomputing path lengths. Assume the use of WordNet as the knowledge structure, which has around 155,000 words. Computing the path lengths of all possible combinations of words would require 24,025,000,000 different entries, equating to approximately 48 GB of memory (assuming 2 byte integers are used to represent path lengths). In general, the storage requirement is bounded by $O(n^2)$, where $n$ is the number of words in the structure.

On the other hand, our mapping to a metric space allows us to represent words as single vectors. Having vector representations of two words allows us to quickly compute the distance between the words according to our method as described. Thus, the vector for each word needs to be precomputed, to be used in an efficient, practical manner. Based on the example of 155,000 words, only 155,000 vectors need to be precomputed and stored. This would equate to approximately 47 MB, assuming each vector is able to be represented in less than 300 bytes. In principle, the memory requirement will depend on the size of the basis, but that size is fixed, so that the storage requirement is bounded by a much smaller $O(n)$. Hence, our representation allows us to store word vectors in a reasonable amount of memory, and having these accessible in memory makes it possible to perform very fast distance computations between two words on the fly.

7. Discussion

There is nothing intrinsically compelling about our approach. In some sense, once the technique is understood, one can envision other ways to approach the subject. We have simply produced one “natural” (i.e., derivative of the metric tensor) linear transformation $\Gamma$ that promotes $\mathcal{N}$ to a metric space. Here, we briefly discuss several important aspects of our approach.

7.1. Role and choice of the basis list (or chains)

While all chains are available in principle, in practice one would be dealing with tens of thousands of chains. The little bit of matrix arithmetic used in our simple example would quickly become computationally challenging. Consider instead a relatively small subset of chains $\{c_\phi\}$. The span of such a subset corresponds to a (convex, closed) subspace of the function space spanned by all chains, and a natural projection operator $\Gamma_\phi$ from $\mathcal{N}$ onto $\mathcal{N}_\phi$ may be defined using $Q^{-1}$.

We can think of $\{c_\phi\}$ as establishing a context for some local neighborhood of the textual information of interest, the analog to an attention mechanism or filter for “working” memory, and $\Gamma_\phi$ as an operator that projects a point of view according to the context $\{c_\phi\}$. There are two natural possibilities for the selection of $\{c_\phi\}$. One is to design a context-independent basis list, which could be used indiscriminately across all (English) text-based tasks. Perhaps, the 2000 or so words of the Longman Defining Vocabulary [1] could serve to produce a suitable generic basis. However, such a basis would of necessity be large and hence computationally unfeasible, and one would expect to be better served with a context-sensitive basis list, especially when dealing with streaming text data. In this case, the basis chains could be generated by considering a sample of nouns taken from or associated with the local input stream, in addition to some constant set of chains representing general background knowledge.

In effect, we allow our interpretive context to evolve as we process the input stream, producing a sequence of projection operators. This is evocative of situations where one approximates globally curvilinear coordinates on a manifold with a patchwork of local linear representations. In particular, the piecewise linear “curve” $\{\Gamma_\phi\}$ (interpreted geometrically as a path through an operator space) might function as a kind of signature for a basic story or line of discourse: something not unlike the notion of a sequence of frames, characterizing not only what does happen, but also what could happen at any point along the story line. Context-sensitive basis lists open the way for our approach to be used for such applications as tracking conversations.

7.2. Handling multiple relations

For simplicity, the description of our method focused on the single relation of hyponymy in the lexical semantic graph. The formalism is naturally extended, and indeed the way is clear for handling multiple relations. For example, let $(\mathcal{N}, o_h)$ be the directed graph of nouns with respect to hyponymy (i.e., the graph we have worked with in this paper) and $(\mathcal{N}, o_m)$ be the directed graph of (the same) nouns with respect to...
meronymy. Then, the vector space of bilinear functions on \((N, o_N) \times (N, o_M)\) is the tensor product of their respective dual function spaces. Our method carries seamlessly into the product space.

8. Conclusion

This paper presents a knowledge-based method to quantify the semantics of nouns by embedding the lexicon into an appropriate metric space through well-known topological operations. The method is illustrated on a few simple examples, which show both the system’s ability to recognize semantic distinctions and its inherent ability to be extended through graph modifications. Experimental results on a significant word similarity problem demonstrate that our method compares very favorably with other approaches to semantic similarity.

We note again that the proposed approach is a departure from most standard work in information extraction and natural language processing, which tend to use techniques from probability theory and statistical learning. This paper shows how topology can be effectively used to address the problem of semantic distance computation. Of course, there are other topologies one can investigate, even on a finite partially ordered set (e.g., the open interval topology). More exotic measures might be derived from such topologies, e.g., measures that take into account weighted edges, for example, and that might have interesting consequences with respect to interpreting “semantic distance.” These are the subjects of future work.

Note also that, in this paper, we have focused on computing semantic distance between individual nouns. The very nature of our topological approach offers a natural extension to computing distance between sets of words, or documents, for example, via the Hausdorff distance. Again this is the subject of future work.

References


3 Meronymy denotes the relation “part/whole”, domain and range restricted by part-of-speech (typically restricted to nouns).