A Graph-Theoretic Framework for Semantic Distance

Vivian Tsang[*]
University of Toronto

Suzanne Stevenson[**]
University of Toronto

Many NLP applications entail that texts are classified based on their semantic distance (how similar or different the texts are). For example, comparing the text of a new document to that of documents of known topics can help identify the topic of the new text. Typically, a distributional distance is used to capture the implicit semantic distance between two pieces of text. However, such approaches do not take into account the semantic relations between words. In this article, we introduce an alternative method of measuring the semantic distance between texts that integrates distributional information and ontological knowledge within a network flow formalism. We first represent each text as a collection of frequency-weighted concepts within an ontology. We then make use of a network flow method which provides an efficient way of explicitly measuring the frequency-weighted ontological distance between the concepts across two texts. We evaluate our method in a variety of NLP tasks, and find that it performs well on two of three tasks. We develop a new measure of semantic coherence that enables us to account for the performance difference across the three data sets, shedding light on the properties of a data set that lends itself well to our method.

1. Introduction

Many natural language tasks can be cast as a problem of comparing texts in terms of their semantic distance. For example, given a suitable text distance measure, document classification can be performed by comparing the text of a new document to the text of various documents whose topics are known. The new document is then labelled with the topic of the document whose text is most similar to it. In general, the texts to be compared may be full documents, as in this example, or may be portions of documents, or even collections of documents. Using text comparison to perform semantic classification has been adopted in a variety of natural language processing (NLP) tasks, from document classification (Scott and Matwin 1998; Rennie 2001; Al-Mubaid and Umair 2006), to prepositional phrase attachment (Pantel and Lin 2000), to spelling correction (Budanitsky and Hirst 2001).

[*] Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3G4, Canada. E-mail: vyctsang@cs.toronto.edu.

[**] Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3G4, Canada. E-mail: suzanne@cs.toronto.edu.

Submission received: 16 December 2007; revised submission received: 18 June 2008; accepted for publication: 20 August 2008.

Distributional methods for semantic distance are widely used and highly successful in comparing texts that are represented as bags of words with associated frequencies of occurrence (Lee 2001; Weeds, Weir, and McCarthy 2004; Pedersen, Banerjee, and Patwardhan 2005). In document classification, for example, the text of a document may be represented as a word frequency vector, which is compared using a distributional distance measure to each of the word frequency vectors of the texts of the documents of known topics. In this way, distributional distance between word vectors captures the semantic distance between two texts that is implicitly encoded in the set of words used in each.

Semantic distance can also be measured more explicitly, by using the relations in an ontology as the direct encoding of semantic association. However, such approaches have generally been limited to calculating the distance between two individual concepts, rather than capturing the distance between two *sets* of concepts corresponding to two texts. Numerous measures have been proposed, for example, for capturing the distance between two concepts in WordNet, typically relying on the synonymy (synset) and hyponymy (is-a) relations (Wu and Palmer 1994; Resnik 1995; Jiang and Conrath 1997, among others). Using such an ontological measure to compare two texts (collections of words instead of single words) might involve mapping each word of a text to its appropriate concept(s) in the ontology, and then calculating the aggregate distance between the two resulting sets of concepts across the ontological relations. For example, one might calculate the semantic distance between the two texts as the average, minimum, maximum, or summed ontological distance between the individual elements of the two sets of concepts (Corley and Mihalcea 2005).

Observe that each of these approaches to text comparison—distributional and ontological—encodes information not contained in the other. Distributional distance captures important information about frequency of occurrence of the words that constitute the target text, whereas ontological distance captures essential semantic knowledge that has been encoded in the relations of an ontology. In response, previous work has attempted to combine distributional and ontological information in computing semantic distance. For example, researchers have developed measures of semantic distance between texts that apply distributional distances to concept vectors of frequencies rather than to word vectors (McCarthy 2000; Mohammad and Hirst 2006). However, these approaches only make pairwise comparisions between the elements of the concept vectors, and do not take into account the important ontological relations among the concepts. In order to capture such relations, other methods have instead integrated distributional information into an ontological method. However, such approaches have heretofore been limited to measuring distance between two individual concepts. For example, some ontological measures use corpus frequencies of words to yield concept weights that are taken into account in measuring the distance between two concepts (Resnik 1995; Jiang and Conrath 1997). What has been missing is an approach to semantic distance between two texts—two sets of words—that can truly integrate distributional and ontological (relational) information, drawing more fully on their complementary advantages for text comparison.

In this article, we describe a new graph-based distance measure that achieves the desired integration of distributional and ontological factors in measuring semantic distance between two sets of concepts (mapped from two texts). An ontology is treated as a graph in the usual manner, in which the concepts are nodes and the relations are edges. A text is represented as a subgraph of the ontology, by mapping the words in the text into their corresponding concepts, which are weighted according to the word frequencies. We call the resulting set of frequency-weighted concepts a *semantic profile*.

By exploiting the relational structure of the ontology, we can explicitly measure the ontological distance over the paths between two profiles. Using the frequencies on the concept nodes, we weight these paths according to the frequency distribution of words in the two texts. The resulting calculation yields a frequency-weighted ontological distance between the two sets of concepts. Thus, we view a text not as a set of items to be compared individually to those in another set (with those individual distances then somehow combined, e.g., as in Corley and Mihalcea [2005]), but rather as a distribution of "mass" within a graph that encodes the semantic relations across the two sets, and use a weighted graph-based approach that captures the aggregate distance between the two frequency masses.

To our knowledge, this is the first method to integrate ontological and distributional information in the graphical calculation of text distance. This article describes the use of the new measure in several different types of NLP text comparison tasks, in order to explore the situations in which such an approach can be effective. Given the novelty of the approach, the task-based evaluation is not intended as the last word on the usefulness of the method, but rather as a first suite of experiments across different types of text comparison tasks to illuminate some of the strengths and weaknesses of such an approach to text distance. We thus analyze the results in detail to identify future directions for further illuminating when and to what extent the method might be useful.

The analysis reveals that our method is not consistently successful across our sample tasks. We hypothesize that, because ontological relations play an integral role in our semantic distance measure, the measure is less effective when the semantic profile for a text (the set of corresponding concepts) lacks semantic coherence. Other work has explored ways to measure the semantic coherence of a set of concepts in terms of their connectedness within an ontology (Gurevych et al. 2003). Because a semantic profile in our work includes both ontological (relational) and distributional (frequency) knowledge, we require a measure of semantic coherence that takes both into account. We develop a novel measure of semantic coherence called *profile density* that captures both the ontological and distributional coherence of a set of frequency-weighted concepts, and apply it to the data sets used in the different tasks to better understand the performance of our semantic distance measure.

Our distance measure is cast as a graphical text comparison task within a network flow framework as described in Section 2. In Section 3, we give an overview of our exploration of the method on three types of text comparison problems. The following three sections present experimental results and analysis of applying our method to the various tasks: verb alternation detection (Section 4), name disambiguation (Section 5), and document classification (Section 6). In Section 7, we describe our profile density measure and use it to analyze the properties of the data sets that lead to the performance differential across the tasks. We conclude the paper with a description of related work in text comparison and graph-theoretic NLP approaches (Section 8) and a discussion of some future directions for our research (Section 9).

2. The Network Flow Method

As noted previously, we treat an ontology as a graph and represent a text as a semantic profile—a collection of nodes in the graph (concepts in the ontology), each having a weight (its frequency). For example, in Figure 1, a small text consisting of the words {*cheese*, *wheat*}, with frequencies of 4 and 10, respectively, is represented as a small weighted subgraph in an ontology by uniformly distributing the word frequencies among the associated concepts. In this way, a text is a weighted subgraph within a

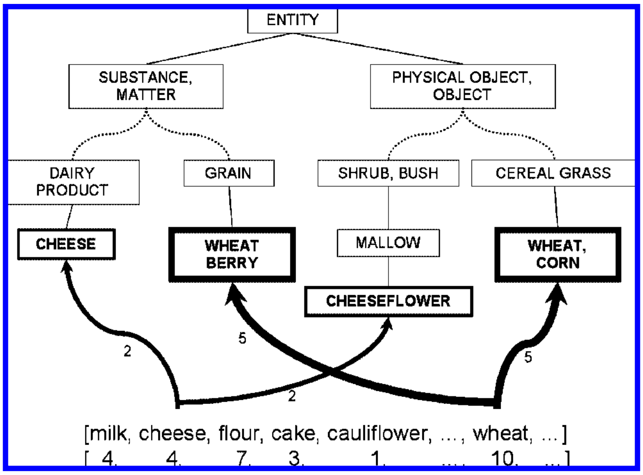

Figure 1
A small text represented as a collection of weighted nodes in a fragment of WordNet.

larger graph (with the thickness of the boxes in the figure indicating weight), and two such weighted subgraphs are connected via a set of paths in the graph.

Our goal is to measure the distance between two subgraphs (representing two texts to be compared), taking into account both the ontological distance between the component concepts and their frequency distributions. To achieve this, we measure the amount of "effort" required to transform one profile to match the other graphically: The more similar they are, the less effort it takes to transform one into the other. (This view is similar to that motivating the use of "earth mover's distance" in computer vision [Levina and Bickel 2001].) In Section 2.1, we first give the intuitive motivation for the approach in terms of the properties of semantic distance that we want to capture by considering transport effort. We then present the mathematical formulation of our graph-based method as a minimum cost flow (MCF) problem in Section 2.2, and describe the formulation of our task within this network flow framework in Section 2.3. In Section 2.4, we return to the properties we identify in Section 2.1 to explain how they are reflected in the MCF formulation.

2.1 An Intuitive Overview

In Figure 2(a), we show a diagrammatic representation of an ontology (the large open triangle) with two profiles, one indicated with filled squares and the other with filled triangles. The location of a filled shape indicates the location of a profile concept in the ontology, and its size indicates its frequency within the profile. We omit edges between the nodes to simplify the diagram, but note that we assume we have a hierarchical, connected ontology; hyponymy links are sufficient. Our goal is to calculate the similarity between the two profiles by determining how much effort is required to transport, along the ontological links, the frequency mass from all of the squares to "fill" the available space in the triangles. The amount of mass to move and the amount of space available are indicated by the sizes of the squares and triangles, respectively. The degree of effort required to transport one to the other indicates the degree of semantic distance.

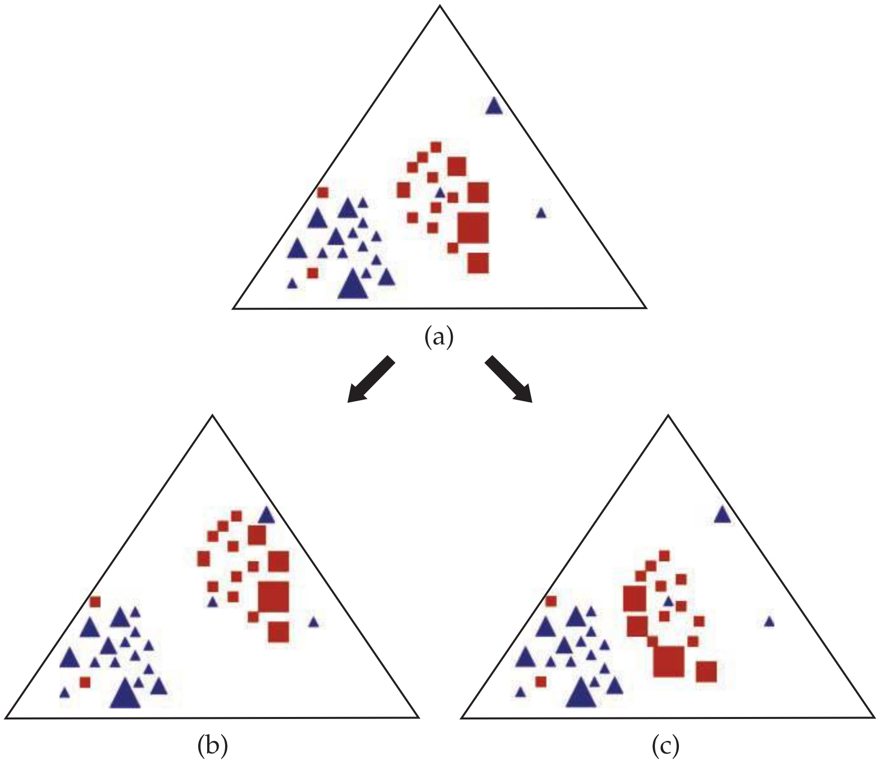

Figure 2
Two subgraphs (one represented by squares, the other, triangles) with varying degrees of overlap and, therefore, similarity within an ontology. Figure (b) differs from Figure (a) in terms of the ontological distance between the square and the triangle clusters. Figure (c) differs from Figure (a) in terms of the size of the individual squares.

The transport effort is determined by both the amount of mass to move and the graphical distance over which it must travel. First consider graphical (ontological) distance between the profiles. Assume the calculated distance between the two profiles in Figure 2(a) is d. In Figure 2(b), the triangle profile is exactly the same. By contrast, although the square profile has the same internal properties (same frequency distribution and graphical structure), its location is further from the triangles. Because the two profiles occupy more distant portions of the ontological space, they are less semantically similar than in Figure 2(a). As desired, the extra ontological distance over which the square frequency mass must be transported to the triangles will cause the calculated distance in Figure 2(b) to be larger than d.

Next consider the effect of varying the frequency distribution over the profile nodes. Again, in Figure 2(c), the triangle profile is exactly the same as in Figure 2(a). However, whereas the nodes of the square profile in Figure 2(c) are in the same locations as in Figure 2(a), their distributional properties are different. The bulk of the frequency distribution is now shifted closer to the nodes of the triangle profile. Because the two profiles have more distributional weight located closer within the ontology, this indicates that the semantic space they occupy is more similar than in Figure 2(a). Correspondingly, because much of the mass of the square profile needs to travel less far to fill the space of the triangle nodes, the calculated distance in Figure 2(c) will be less than d.

It is worth noting explicitly that this notion of semantic distance as transport effort of concept frequency over the relations (edges) of an ontology differs significantly from an approach to semantic distance that utilizes concept vectors of frequency. By crucially utilizing the relations between concepts in calculating semantic distance, our approach can determine the distance between texts that use related but non-equivalent concepts. For example, our measure will find greater similarity between a text that discusses *milk* and one that discusses *cheese* than between one that discusses *milk* and one that discusses *bread*. A vector distance would find each of these equally dissimilar, because there are no concepts in common, and there is no way to relate *milk* to *cheese*.[1]

The intuitive examples in Figure 2 show that calculating semantic distance as transport effort captures in a well-motivated way both the ontological distance between the profiles and their weighting by the distributional amounts of the concept nodes. In the next subsection, we describe a mathematical formulation that captures the relevant properties of our problem in a network flow framework. Network flow methods are often used in computer science for modelling such transport effort, for example, in communication or transportation networks.

2.2 Minimum Cost Flow

Our intuitive transport effort examples above can be viewed as a supply–demand problem, in which we find the minimum cost flow (MCF) from the supply profile to the demand profile to meet the requirements of the latter. Mathematically, let $G = (N, E)$ be a connected graph representing an ontology, where N is the set of nodes representing the individual concepts, and E is the set of edges representing the relations between the concepts. (Most ontologies are connected; in the case of a forest, adding an arbitrary root node yields a connected graph.) Each edge has a cost $c : E \rightarrow \mathbb{R}$, which is the ontological distance of the edge. Each node $i \in N$ is associated with a value $b(i)$ such that $b : N \rightarrow \mathbb{R}$ indicates its available supply ($b(i) > 0$), its demand ($b(i) < 0$), or neither ($b(i) = 0$). The goal is to find a flow from supply nodes to demand nodes that satisfies the supply/demand constraints of each node and minimizes the overall "transport cost."

First, we have to define a function to describe the flow entering i via an incoming edge (h, i) and exiting i via an outgoing edge (i, j). Let IN_i be the set of edges (h, i) with a flow entering node i; similarly, let OUT_i be the set of edges (i, j) with a flow exiting node i. Then, the flow entering and exiting node i is captured by $x : E \rightarrow \mathbb{R}$ such that we can observe the combined incoming flow, $\sum_{(h,i) \in IN_i} x(h, i)$, from the entering edges IN_i, as well as the combined outgoing flow, $\sum_{(i,j) \in OUT_i} x(i, j)$, via the exiting edges OUT_i (see Figure 3). A valid flow, x, must be found such that the net flow at each node—the difference between its exiting flow and its entering flow—equals its specified supply or demand constraints. For example, in Figure 2 where the squares represent the supply and the triangles represent the demand, a solution for x would allow us to transport all the weight at the squares to fill the triangles, via a set of routes connecting them.

[1] Techniques such as SVD or LSA could be applied to the concept vectors, as with word vectors, yielding potential relations through unnamed concepts (e.g., Landauer and Dumias 1997). Note, however, that such methods are dependent on the usages of the concepts implicitly encoding such connections, whereas an ontology-based method draws on a knowledge base that explicitly encodes the relations regardless of the particular usages of the concepts.

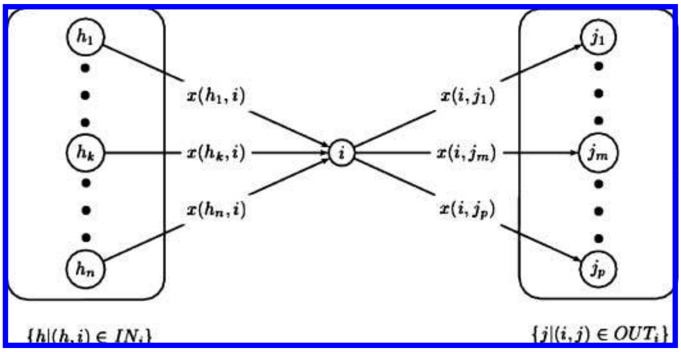

Figure 3
An illustration of flow entering and exiting node *i*.

Formally, the MCF problem can be stated as follows (from Chvátal 1983):

$$\text{Minimize} \quad z(\vec{x}) = \sum_{(i,j)\in E} c(i,j) \cdot x(i,j) \tag{1}$$

$$\text{subject to} \quad \sum_{(i,j)\in OUT_i} x(i,j) - \sum_{(h,i)\in IN_i} x(h,i) = b(i), \forall i \in N \tag{2}$$

$$\text{and} \quad x(i,j) \geq 0, \forall (i,j) \in E \tag{3}$$

The constraint specified by Equation (2) ensures that the difference between the flow entering and exiting each node i matches its supply or demand $b(i)$ exactly. The next constraint, Equation (3), ensures that the flow is transported from the supply to the demand but not in the opposite direction. The calculation of z in Equation (1) (which is subject to these constraints) multiplies the amount of flow travelling along each edge, $x(i,j)$, by the transportation cost of using that edge, $c(i,j)$. Taking the summation over all edges of the product $c(i,j) \cdot x(i,j)$ yields the desired transport effort of using the supply to fill the demand.[2]

2.3 Semantic Distance as MCF

To cast our text comparison task into this framework, we first represent each text as a semantic profile in an ontology. The profile of one text is chosen as the supply (S) and the other as the demand (D); our distance measure is symmetric, so this choice is arbitrary. In our examples in Section 2.1, the square profile was seen as the supply and the triangle

[2] We cast our text comparison problem as an uncapacitated minimum-cost flow problem, i.e., there is no upperbound constraint placed on the amount of flow along each edge (see Equation (3)). Unlike a capacitated version of MCF, which is NP-complete (Garey and Johnson 1979), our problem is tractable and can be solved in polynomial time.

profile as the demand. The concept frequencies of the profiles are normalized, so that the total supply equals the total demand.

The cost of the routes between nodes is determined by a semantic distance measure defined over the nodes in the ontology—that is, a measure of individual concept-to-concept distance. A relation (such as hyponymy) between two concepts i and j is represented by an edge (i, j), and the cost c on the edge (i, j) can be defined as the concept-to-concept distance between i and j. For simplicity in this article, we use edge distance as our concept-to-concept distance measure c; that is, each edge (i, j) has a cost of 1, and the distance between any two concepts is the number of edges separating them.[3]

Next, we must determine the value of $b(i)$ at each concept node i. In the simple case, i occurs in only one profile or the other. If $i \in S$, $b(i)$ is set to the normalized supply frequency, $f_S(i)$. If $i \in D$, $b(i)$ is set to the negative of the normalized demand frequency, $-f_D(i)$, since demand is indicated by a value less than zero. However, i may be part of both the supply and demand profiles, and then $b(i)$ must be set to the net supply/demand at node i. Thus we have:

$$b(i) = f_S(i) - f_D(i) \tag{4}$$

For example, if the supply profile contains a node *car* with frequency of 0.25, and the same node in the demand profile has a frequency of 0.7, then $b(car)$ is -0.45. In other words, the node *car* has a net demand of 0.45.

Recall that our goal is to transport all the supply to meet the demand; the key step is to determine the optimal routes between S and D such that the constraints in Equation (2) and Equation (3) are satisfied. The total distance of the routes, or the MCF—$z(\vec{x})$ in Equation (1)—is the distance between the two semantic profiles.

2.4 Ontological and Distributional Factors in MCF

To see how the factors of ontological distance and frequency distribution play out in the MCF formulation, let's return to our square and triangle profile example. Consider a hypothetical zoomed-in area of the earlier diagram in Figure 2(a), shown in Figure 4. Here we assume that the square nodes have a net supply ($b(i) > 0$) and the triangle nodes have a net demand ($b(i) < 0$).[4] The size of the square and triangle nodes in the figure indicates $|b(i)|$—i.e., the relative supply/demand, respectively. The circles indicate nodes with neither supply nor demand constraints—i.e., $b(i) = 0$. Each arrow from node i to node j indicates the source and destination for transported flow from a square node to a triangle. The length of an arrow represents the ontological distance, $c(i, j)$, and the width indicates the amount of flow, $x(i, j)$. Note that the mass at the rightmost square in the figure has to be distributed over the two triangles, and the mass at the leftmost square is transported over a path with one edge (as indicated by

[3] Some semantic distances, such as those of Lin (1998) and Resnik (1995), do not directly use the underlying graph structure of the ontology in calculating the distance between two concepts. Using this type of distance in our MCF framework requires an extra graph transformation step; see Tsang and Stevenson (2006) for more details.

[4] Earlier we made the simplifying assumption that square nodes were the supply profile and triangle nodes the demand profile. We have now seen that a node can belong to both profiles, and its characterization more accurately is stated in terms of *net* supply/demand. Thus, for example, a square node may belong to just the supply profile or to both the supply and demand profile; the defining factor is that it has a net supply.

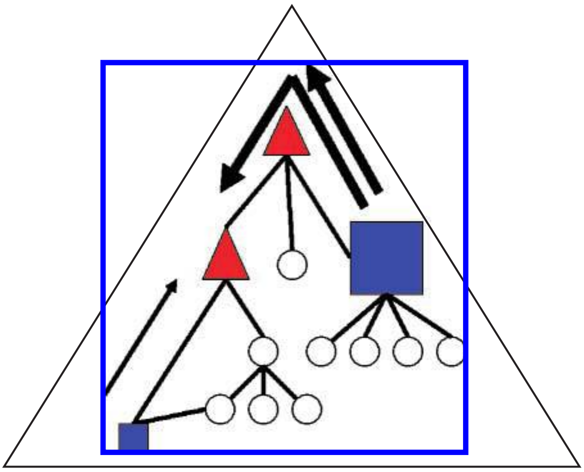

Figure 4
An example of transporting the weights at the square nodes (supply nodes) to the triangle nodes (demand nodes). The circle nodes have zero supply/demand requirement.

the arrow nearby) instead of a path with three edges (with two circle nodes on the path). The aggregated length and width of the three arrows corresponds to the minimum cost flow, i.e., the semantic distance between the profiles represented by the squares and triangles.

Both the ontological distance between nodes and the node weights are important in determining the minimum cost flow. The role of ontological information in the MCF formulation is clear. If the squares were further away from the triangles in the ontology in Figure 4—that is, if more edges separated the squares and the triangles—the sets of concepts they represent would be less semantically similar. The length of the arrows (representing $c(i, j)$) would be greater, and the resulting MCF would be larger, reflecting the greater semantic distance between the profiles. Distributional information in this method is equally critical to the distance calculation, because it determines the amount of supply/demand at each node. If the squares in Figure 4 were more uniformly sized, the two profiles would be more semantically similar because the weight would be distributed more similarly across the ontological space. In this case, less flow would have to travel from the rightmost square to the leftmost triangle (i.e., the corresponding arrow would be thinner, representing $x(i, j)$), and the resulting MCF would therefore be smaller. In short, our MCF method captures the desired property that both ontological distance between profile nodes and their frequency distributions determine the overall semantic distance between two profiles.

3. Evaluation: Experimental Tasks and Methodology

We select three different NLP tasks that can be formulated as text classification problems based on semantic distance between the texts. In each case, the texts to be compared are treated as bags of words with associated frequencies. The tasks are chosen to reflect different types of relations used to extract the relevant words, to see if a varying amount of constraint on the words comprising a text influences the performance of our method.

In verb alternation detection (Section 4), we identify which verbs, out of a set of target and filler verbs, allow a certain variation in the syntactic expression of their underlying argument structure. The task is achieved by comparing the set of head words that occur with the verb in each of two different syntactic positions (e.g., subject of intransitive and object of transitive). In this task, the words that make up the texts to be compared have a particular syntactic relation to the verb under consideration. In proper name disambiguation (Section 5), we classify the sense of an ambiguous name according to its local context. This task is similar to word sense disambiguation (WSD), in picking the intended sense of a term, but also has similarities to topic identification, since the proper name delineates a particular domain of discourse. In this task, we compare the text constituting the ambiguous instance to texts representing each of the known referents of the name. Here, the words of a text are extracted from a small window of occurrence around the target name token (25 words on each side), regardless of the syntactic relations among the words. For the known referents, the words from these windows are aggregated across a small set of labelled instances. In document classification (Section 6), a text is classified into one of a restricted number of topic categories. The text to be classified consists of all the words in a document; for each topic, it is compared to a set of words corresponding to a small set of known documents for that topic. The extracted words are not constrained by syntactic relation (as in verb alternation) or even by distance to a target element (as in name disambiguation).

In each case, the resulting bag of words for a text must be mapped into a semantic profile—a frequency-weighted set of concepts in an ontology. Because all three of our tasks involve general domain text, we use WordNet as our ontology (Fellbaum 1998).[5] (A domain-restricted task may motivate the use of a domain-specific ontology, such as UMLS for comparing medical texts as in Bodenreider [2004].) Because the noun hierarchy of the WordNet ontology is most developed, we restrict our semantic profiles to use only the nouns from the bag of words corresponding to a text: Any word in the text that appears in the noun hierarchy of WordNet is included in the bag of nouns.

The bag of nouns with their associated frequencies must be mapped to the appropriate concepts in WordNet. Given the current state of unsupervised WSD, there is generally no attempt to disambiguate the words of a text when performing this kind of mapping—that is, there is no selection of the most appropriate concept or set of concepts to map the words to, given the context of their use. The simplest method is to distribute the frequency of each word uniformly to its corresponding concepts. For example, Ribas (1995) maps the word frequency to the most specific concept(s) for the word, including all of the possible synsets for the word, but not their hypernyms. Resnik (1993) also distributes the word frequency uniformly, but does so across the most specific concept(s) and all of their hypernyms. Other approaches, although still avoiding the difficulties of WSD, do try to capture the overall semantic "tendencies" of the set of words. Such methods estimate the appropriate probability distribution over a set of concepts to represent a given bag of nouns as a whole (Li and Abe 1998; Clark and Weir 2002). However, such techniques still start with a mapping of each word to all of its immediate concepts.

[5] There is disagreement over the suitability of treating WordNet as an ontology, rather than as a lexical network (Gangemi, Guarino, and Oltramari 2001; Hirst 2009). However, the intention of the creators of WordNet is apparently that its synsets correspond to concepts, and the relations between them include both "conceptual-semantic and lexical relations" (`http://wordnet.princeton.edu/`), qualifying it, under some views, as a general domain ontology. Although recognizing the limitations and difficulties of using a primarily lexical resource as an ontology, we note that WordNet is standardly used as such in computational linguistics, and so we adopt this use here.

For all three of our tasks, we take the simple approach of mapping each noun individually to its most specific concepts (not their hypernyms), uniformly dividing the word frequency among them. In verb alternation, we also experiment with the possibility of finding the best set of frequency-weighted concepts for the full bag of nouns (using the techniques of Li and Abe [1998] and Clark and Weir [2002]), to see if this affects the performance of our method.

The precise classification experiment performed using these semantic profiles is described in detail subsequently in the section for each task. In each case, we compare the performance of our MCF method on the semantic profiles to one or more purely distributional methods using the original word frequency vectors.

4. Task 1: Verb Alternation Detection

Verb alternation refers to variations in the syntactic expression of verbal arguments. If a verb participates in an alternation, the same underlying semantic argument may appear in varying positions (slots) of the verb's subcategorization frames. For example, the following sentences show that the argument undergoing the melting action can appear as the subject of an intransitive use of *melt* (1a) or as the object of a transitive use (1b).

1a. The chocolate melted.

1b. The cook melted the chocolate.

This type of intranstive/transitive pairing is known as the causative alternation because of the explicit expression of the causer (*the cook*) in the transitive alternant.

It has long been hypothesized that the semantics of a verb and its relations to its arguments at least partially determine the syntactic expression of those arguments (see Pinker [1989], among others). Influential work by Levin (1993) showed that this relationship could be exploited "in reverse" by using alternation behavior as an indicator of the underlying semantics of a verb—specifically, that verbs undergoing the same sets of alternations form classes with similar semantics. Computational linguists have built on this work by demonstrating that statistical cues to alternation behavior can be used to automatically place verbs into semantic classes (Merlo and Stevenson 2001; Schulte im Walde 2006).

Detection of verb alternation behavior can be cast as a text comparison problem (McCarthy 2000; Merlo and Stevenson 2001). Consider an alternation such as the causative illustrated in Example (1). The set of nouns appearing as the subject of the intransitive (such as *chocolate*) have the same relation to the verb as the set of nouns appearing as the object of the transitive. Because the verb places constraints on what kinds of entities can be in that relation (here, things that are meltable), the two sets of nouns should be similar. Hence, to identify a particular alternation for a verb, the set of nouns in a certain slot of one of its subcategorization frames is compared to the set of nouns in the alternating slot for that semantic argument in another subcategorization frame.

For example, Merlo and Stevenson (2001) devise a simple lemma overlap score that counts the number of tokens appearing in *both* of the relevant syntactic slots. McCarthy (2000) instead compares two semantic profiles in WordNet that contain the concepts corresponding to the nouns from the two argument positions. In McCarthy's method, the profiles are first generalized to a set of higher level nodes in the hierarchy (starting with the method of Li and Abe [1998]); next, skew divergence is used to find the distance between the resulting vectors of concepts. Here we use our network flow method to directly compare the semantic profiles corresponding to the noun sets. Our method allows us to compare sets of weighted concepts as in McCarthy's, but using a distance method that applies within the ontology graph, rather than simply using a distributional distance measure over concept vectors.

4.1 Experimental Set-up

We adopt the data set from an investigation of a semantic distance measure that was a precursor to our network flow method (Tsang and Stevenson 2004). The selection of these verbs and extraction of their arguments are discussed in the following two sections; we then describe our evaluation methodology.

4.1.1 Experimental Verbs. We evaluate our method on the causative alternation. As noted previously, in this alternation the target syntactic slots for comparison are the subject of the intransitive (Subj-Intrans) and the object of the transitive (Obj-Trans). (These are the positions of *the chocolate* in Examples (1a) and (1b), respectively.) To identify verbs undergoing this alternation, we randomly selected verbs from among Levin classes that are indicated to allow the causative alternation. This allows us to test the ability of a distance measure to detect alternation behavior among verbs from a range of semantic classes which may differ in other respects.

We refer to the verbs that are expected to undergo the causative alternation as **causative verbs**. For comparison, we randomly selected an equal number of **filler verbs**, subject to the constraint that their Levin classes do not allow a causative alternation. (Specifically, none of the classes containing a filler verb allows an alternation in which the same underlying argument appears in the Subj-Intrans slot as well as the Obj-Trans slot.) The full set of potential causative and filler verbs were filtered according to corpus counts, as described next.

4.1.2 Corpus Data and Argument Extraction. We used a randomly selected 35M-word portion of the British National Corpus (BNC; Burnard 2000). The text was parsed using the RASP parser of Briscoe and Carroll (2002), and subcategorization frames were extracted using the system of Briscoe and Carroll (1997). Each subcategorization frame entry for a verb includes a list of the observed argument heads per slot along with their frequencies. For each verb/slot pair, we thus extracted the set of nouns used in that slot along with their frequency of occurrence.

Verbs were filtered from the potential list of experimental items if they occurred less than 10 times in our corpus in either the transitive or intransitive frame. The verbs were then divided into multiple frequency bands: high (at least 450 instances), medium (between 150 and 400 instances), and low (between 10 and 100 instances). An equal number of verbs of each type (causative and filler) were randomly selected within each band, yielding a total of 120 experimental verbs in balanced data sets of 60 items for development and 60 items for testing. The development data was used in our earlier work to select a profile-generation method for the test data (Tsang and Stevenson 2004). In our current work, we did not make any adjustments to our method based on results on the development set (i.e., it was not used to set any parameters or select a particular implementation approach). Hence, we report the evaluation of our method on the full set of 60 verbs in each of the data sets, as well as individually on the three frequency bands of 20 verbs each. We refer here to the original "development" and "test" data sets as "dataset1" and "dataset2".

4.1.3 Evaluation Methodology. For each verb, we create a semantic profile for each of the Subj-Intrans and Obj-Trans slots. We first take the argument heads with their frequencies from the appropriate slots in the extracted subcategorization frame for the verb. We then map these words with their frequencies to the corresponding nodes in WordNet, as described in Section 3. (We also consider here different profile generation methods, discussed later in Section 4.2.2.) We then calculate the network flow distance between the two semantic profiles for each verb, yielding a distance calculation for that verb. Recall that we expect verbs that participate in the alternation to have more similar semantic profiles corresponding to the Subj-Intrans and Obj-Trans nouns. For example, a causative verb like *melt*, as in Examples (1a) and (1b), may have words like *chocolate*, *sherbet*, and *glacier* in the Subj-Intrans slot, and words like *chocolate*, *butter*, and *bronze* in the Obj-Trans slot. In contrast, a non-causative verb like *fry* will typically have more dissimilar sets of words that contribute to the two profiles (e.g., *cook*, *wife*, and *chef* in the Subj-Intrans slot, and *egg*, *noodle*, and *onion* in the Obj-Trans slot). We thus rank all the verbs by the distance calculation, and (as in McCarthy 2000) set a threshold to divide the verbs into causative (smaller distance values) and non-causative (larger distance values). Following McCarthy, we experimented with both the mean and median values as the threshold, but found little difference. We report the results using the median distance as the threshold, because this provided more consistent results with our method.

Because we label all verbs in our experiments as causative or non-causative, we use accuracy as the performance measure. Since we have balanced data sets, the random baseline is 50%. We compare our results as well to a number of distributional methods (as enumerated in the next section). Given the small size of our data sets, a simple statistical test on the resulting accuracies is not powerful enough to reveal differences when the accuracies are close. However, because the difference in methods is due to variation in how they rank the experimental items, we perform a Wilcoxon signed rank test (Wilcoxon 1945) to determine when the rankings between two methods are significantly different, using a p value of .05.

4.2 Results and Analysis

As noted herein, we present results on two sets of data, and also examine the effect of using alternative profile generation methods. We compare our network flow distance (NF) to a number of other distance measures including probability distributional distances given by Jensen-Shannon divergence (JS) and skew divergence (skew div) (Lee 2001), as well as the general vector distances of cosine, Manhattan distance, and Euclidean distance.

4.2.1 Experimental Results. On dataset1, our network flow distance performs better than or as well as all other measures on the individual frequency bands, as shown in Table 1. On all verbs combined (the "All" column) the performance of our method is not the best, although the Wilcoxon test shows no significant difference between the rankings of NF and the best measure (Manhattan). (The difference in rankings between NF and all other measures is significant.)

Interestingly, we find that the "All Verbs" performance of NF (and that of several other methods) is indeed worse than the performance on the individual frequency bands. We examined the distance values across the frequency bands to determine the cause for this pattern. We found that low frequency verbs tend to have smaller distances

Table 1
Accuracies on dataset1 by the network flow method (NF), cosine, Manhattan distance, Euclidean distance, skew divergence (skew div), and Jensen-Shannon divergence (JS). Best accuracies in each condition are shown in **boldface**.

	All	Frequency Bands			Avg of
	Verbs	High	Medium	Low	Bands
NF	0.60	**0.70**	**0.70**	**0.70**	**0.70**
cosine	0.57	0.60	0.60	0.60	0.60
Manhattan	**0.63**	**0.70**	**0.70**	**0.70**	**0.70**
Euclidean	0.47	0.40	0.50	0.40	0.43
skew div	0.57	0.60	0.60	0.50	0.57
JS	0.60	**0.70**	0.60	**0.70**	0.67

Table 2
Accuracies on dataset2 by the network flow method (NF), cosine, Manhattan distance, Euclidean distance, skew divergence (skew div), and Jensen-Shannon divergence (JS). Best accuracies in each condition are shown in **boldface**.

	All	Frequency Bands			Avg of
	Verbs	High	Medium	Low	Bands
NF	0.67	**0.60**	**0.80**	**0.60**	**0.67**
cosine	0.50	**0.60**	0.50	0.50	0.53
Manhattan	0.63	**0.60**	**0.80**	**0.60**	**0.67**
Euclidean	0.60	0.50	0.70	0.50	0.57
skew div	0.63	**0.60**	**0.80**	**0.60**	**0.67**
JS	**0.70**	**0.60**	**0.80**	**0.60**	**0.67**

between the two slots and high frequency verbs tend to have larger distances. This is due to the fact that higher frequency verbs typically occur with a wider range of nouns, leading to a more dispersed semantic profile (i.e., a larger number of concepts). As a result, the best threshold for separating the alternating and non-alternating verbs differs across the frequency bands, and the threshold for all verbs together lies in between the thresholds for the high and low frequency bands. When classifying all verbs, the frequency effect may result in more false positives for low frequency verbs (which have generally smaller distance values), and more false negatives for high frequency verbs (which have generally larger distance values). The column labelled “Avg” in Table 1 shows the performance when averaging the results across the individual frequency bands. For most methods, including ours, the “Avg” results are much better than when considering all verbs together (the “All” column).

Table 2 reports the performance on dataset2, which is similar to that on dataset1. Again, we find that our method is tied for the best performance in every condition except for all verbs combined. (Here we find that all four methods over .60 accuracy in the “All” condition have statistically indistinguishable rankings of the experimental items.) On this data set, taking the average of the frequency bands does not help performance of our method compared to “All,” but neither does it hurt (and for most methods “Avg” does better or the same as “All”). We conclude that separating items by frequency may be required to achieve robust results in this type of task.

Although our method is tied for best in every condition except “All,” neither is our method distinguished from several of the other distance measures. Given the

Table 3
Average accuracies by the network flow method (NF), Manhattan distance (Man), skew divergence (skew div), and Jensen-Shannon divergence (JS) on different profiles: original ("raw"), Li and Abe, and Clark and Weir profiles. Best accuracies in each condition are shown in **boldface**.

	raw		Li and Abe		Clark and Weir	
	Dataset1	Dataset2	Dataset1	Dataset2	Dataset1	Dataset2
NF	**0.70**	**0.67**	0.50	**0.67**	**0.73**	**0.70**
Manhattan	**0.70**	**0.67**	0.57	**0.67**	0.60	0.57
skew div	0.57	**0.67**	0.53	**0.67**	0.68	0.60
JS	0.67	**0.67**	**0.63**	**0.67**	0.63	0.53

relatively small amounts of data per verb (with profiles averaging about 900 nodes in size), it is possible that the raw profiles suffer from a sparse data problem and are not sufficiently capturing the conceptual similarities among alternating slots. McCarthy (2000) addressed this issue by using a technique for generalizing concept nodes prior to comparing profiles. We explore this issue next.

4.2.2 Comparing Different Profile Generation Methods. Our experiments use semantic profiles created directly from the word frequencies, as described earlier. However, research has explored the possibility of generalizing this kind of "raw" data to a semantic profile that more appropriately reflects the coherent concepts expressed in the original set of weighted concept nodes. This can be especially useful when creating semantic profiles from small amounts of data, given the noise introduced in the mapping of words to concepts.[6] To explore the effect of different profile generation methods on this task, we consider here two approaches, that of Li and Abe (1998) and Clark and Weir (2002). Both these methods start with a semantic profile generated as described in Section 3 and attempt to find the set of nodes in the ontology that appropriately generalize the concepts in the "raw" profile.

Table 3 compares the performance of the network flow distance with that of several other measures on the original ("raw") profiles, the Li and Abe profiles, and the Clark and Weir profiles. Results are reported for the average of the individual frequency bands, since that produced the best results overall in our earlier experiments. The results for cosine and Euclidean distance are omitted, because they perform worse overall than the other measures.[7]

The best results across both data sets are achieved by our network flow method on the Clark and Weir profiles. Considering the results across all profile types, the network flow approach is most consistent, achieving the best (or tied for best) performance in but one condition (dataset1 with Li and Abe profiles). The distributional methods

[6] Because we divide the frequency of a word uniformly among all the word's concepts, with no attempt at disambiguation or informed weighting, much noise is introduced. Given the small amounts of data, the noise may be sufficient to mislead our network flow method.

[7] Because these results use the approach of averaging results across the frequency bands, we cannot apply the Wilcoxon signed rank test to the rankings. (The individual frequency bands have too few items for the test to detect differences.) On All Verbs combined (results not reported in this table), the rankings of NF are different from all other methods on each combination of data set and profile generation approach, except in the single case of Manhattan and JS on dataset2 using Li and Abe to create the profiles.

(Manhattan, skew div, JS) in almost all cases perform worse on the generalized profiles than on the "raw" profiles. (The one exception is that skew divergence does better on dataset1 on the Clark and Weir profiles.)

Overall, then, it seems that raw data is likely best for a purely distributional method, but the Clark and Weir profiles enable the network flow method to outperform them by exploiting the graph structure of the ontology. Indeed, when comparing our method to the others on the Clark and Weir profiles for the individual frequency bands (not shown in the table), we find that much of our performance advantage comes on the low frequency verbs. This indicates that the combination of our method with a suitable generalization technique is especially important when dealing with sparse data.

We examine the data further to discover why the Li and Abe profiles yield poorer performance in most cases on dataset1. We find that Li and Abe's (1998) method tends to generate profiles with more general concepts. For example, when given an original set of concepts such as *Edam*, *Brie*, *Sockeye*, and *Chinook*, the method may produce a single general concept such as *food* instead of the two concepts *cheese* and *salmon* that capture the two kinds of food that are indicated. The loss of semantic information from using overly general concepts may produce the decrease in performance.

For comparison, we also apply McCarthy's (2000) method to our dataset2, and find that it achieves only 0.60 on all verbs and 0.53 averaged over the three frequency bands. Her method is especially poor on low frequency verbs (below chance at 0.40). We hypothesize that her method is less robust to low frequency counts because it may overgeneralize the data by first applying Li and Abe's (1998) method, and then generalizing the nodes even further.

We see that although some amount of generalization of the semantic profiles is useful in this task, overgeneralization may be harmful. We leave it to future work to explore the interaction of our network flow method with different types of profile generation across various tasks. Because the next two tasks we consider use larger amounts of data, we only experiment with raw profiles in these cases.

5. Task 2: Name Disambiguation

Interest in the NLP problem of name disambiguation has increased as the growth of the World Wide Web has led to large numbers of ambiguous name references in online text. For example, Web sites or documents containing the name *John Edwards* may refer to the U.S. presidential candidate for 2008, an NBA basketball player, or a British medical geneticist. An ambiguous name may be resolved by comparing its local textual context—the set of words it co-occurs with—with the local textual contexts of the name when its reference is known. For example, the text surrounding the name *John Edwards* in its various uses are very likely to include distinguishing words such as *politician* vs. *game* vs. *research*. Many approaches have been proposed for resolving name ambiguity by using distributional methods over contextual information (Xu, Liu, and Gong 2003; Han, Zha, and Giles 2005; Pedersen, Purandare, and Kulkarni 2005).

In this section, we present the application of our network flow distance measure to a name disambiguation task, and demonstrate the benefits of combining ontological and distributional knowledge in this task. The particular task we examine is one of "pseudo name disambiguation," in which the texts containing matched pairs of *different* names are extracted, and then the two different names are replaced by a single symbol, leading to an ambiguous "name" across the two sets of texts. The goal is to recover the correct target name in each instance. For example, the names of two soccer players (Ronaldo and David Beckham) form one disambiguation task, and the names of an ethnic group and a diplomat (Tajik and Rolf Ekeus) form another. This task was established by Pedersen, Purandare, and Kulkarni (2005) to provide "annotated" experimental data (with each text indicating the correct name) without the need for expensive manual annotation.

In Pedersen, Purandare, and Kulkarni (2005), an unsupervised method of name discrimination through text clustering was used to address this task. This is infeasible for a method like ours, in which each distance calculation requires access to an ontology. (The worst-case complexity of clustering with our method is quadratic in the size of the ontology used; a detailed discussion can be found in Tsang and Stevenson [2006].) Instead, we use a supervised methodology, but experiment with varying small amounts of data in a minimally supervised approach. Although our method requires extra manual effort in the form of data annotation for training, we find that the amount of annotated data required is modest.

5.1 Experimental Methodology

5.1.1 Corpus Data. We use Pedersen, Purandare, and Kulkarni's (2005) data set, which was taken from the Agence France-Press English Service portion of the GigaWord English corpus distributed by the Linguistic Data Consortium. They extracted the local context of six pairs of names of varying confusability, including: the names of two soccer players (Ronaldo and David Beckham); an ethnic group and a diplomat (Tajik and Rolf Ekeus); two companies (Microsoft and IBM); two politicians (Shimon Peres and Slobodan Milošević); a nation and a nationality (Jordan and Egyptian); and two countries (France and Japan). For each name instance, the extracted text consists of 50 words (25 words to the left and to the right of the target name), with the target name obfuscated. For example, for the task of distinguishing *David Beckham* and *Ronaldo*, the target name in each instance becomes *David_BeckhamRonaldo*. The original name in each instance is retained only for evaluating the results (and for training, in the case of our method, as described subsequently). (Note that this approach to data creation avoids the use of manually annotated data for this experimental task, but in an actual application, manual annotation of truly ambiguous names would be necessary.) Each pair of names thus serves as one of six name disambiguation tasks. Table 4 shows the number of instances per task (name pair). The "Majority" column also indicates the relative frequency of the majority name in each pair, which we adopt as the baseline accuracy.

5.1.2 Classification Using the Network Flow Method. As mentioned previously, we take a supervised approach, in which name instances are classified with the use of training

Table 4
The pairs to be identified, the raw frequencies, and the relative frequency of the majority name.

Name 1	Count	Name 2	Count	Total	Majority
Ronaldo	1,700	David Beckham	752	2,452	0.69
Tajik	3,002	Rolf Ekeus	1,071	4,073	0.74
Microsoft	3,401	IBM	2,406	5,807	0.59
Shimon Peres	7,686	Slobodan Milošević	6,048	13,734	0.56
Jordan	25,039	Egyptian	21,392	46,431	0.54
Japan	116,379	France	110,435	226,814	0.51

data annotated by the original name in the instance. To generate our training data, we randomly select a portion of the instances for each of the 12 names. All the training instances for a name are used to form a single aggregate semantic profile, which serves as the gold-standard for that name. The remaining instances serve as test data; for each of these, we build an individual semantic profile. All profiles are generated as described in Section 3, namely, each frequency count for a word is distributed uniformly among the corresponding concepts in WordNet. A gold-standard profile is constructed in exactly the same way except that its word frequency vector is created by aggregating the word counts from all the relevant training instances. Note that there is nothing special about such a profile or how it is formed; it simply aggregates counts from multiple contexts.[8]

To classify a name instance, we measure the network-flow distance between the individual profile of the ambiguous instance and each of the two gold-standard profiles for that task. The name whose gold-standard profile has the shortest distance to the instance profile is the name assigned to the ambiguous instance. For example, assume we have a "David_BeckhamRonaldo" instance to be classified. We compare its profile to each of the gold standard profiles for "David Beckham" and "Ronaldo" by measuring the distance between each of the two pairs of profiles. If the instance profile has a shorter distance to the profile for "David Beckham" than to that of "Ronaldo," then it is classified as "David Beckham," otherwise as "Ronaldo."

5.1.3 Evaluation Methodology. We use the accuracy of labelling all instances as our evaluation measure. To compare to prior results using F-measure, we report that in some tables. Because we label all instances, accuracy and F-measure are equivalent in our method, using $2rp/(r+p)$ as the definition of F-measure.

The random baseline for our task is the accuracy of labelling all instances with the predominant name, as shown in the "Majority" column of Table 4. Because we use the data set of Pedersen, Purandare, and Kulkarni (2005), we compare our performance to their distributional method (reporting their best results both with and without singular value decomposition). Because their method is an unsupervised one, we also train and test a supervised learner using distributional data (LIBSVM by Chang and Lin [2001]). For each set of training data, we remove stopwords and use the remaining words (with their frequencies) as input features for the SVM. We then obtain the optimal parameters (i.e., optimal values for cost and gamma in LIBSVM) by using 10-fold cross-validation over the training data. Finally, we perform classification on the test data using those parameters. This enables us to compare our results to a purely distributional method with access to the same training data.

Because our method is supervised, it is important to minimize the amount of annotated data required to build the gold-standard profiles. (Lengthy training time can also be an issue for a supervised method, but here "training" is the straightforward task of building an aggregate semantic profile.) Because it is unclear a priori what amount of training data is sufficient, we experiment with several quantities. We initially select 200 random instances per pair of names, respecting the relative proportions of the two names overall. (Two hundred instances constitute about 0.1–10% of the data per pair of names.) Subsequently, we decrease the quantity further, to one-half and one-quarter the original amount (100 and 50 instances, respectively) to observe how the

[8] In our later experiment in document classification, on a subset of our data, we tried a nearest neighbor approach to all training instances rather than aggregating them, but this did not perform as well.

Table 5
Network flow results using 200 training instances on the random samples and their average performance.

Name Pair	Random Samples					Average of
	1	2	3	4	5	Samples
Ronaldo/Beckham	0.78	0.83	0.76	0.79	0.84	0.80
Tajik/Ekeus	0.98	0.98	0.97	0.96	0.98	0.97
Microsoft/IBM	0.73	0.72	0.73	0.74	0.73	0.73
Peres/Milošević	0.96	0.96	0.97	0.96	0.97	0.96
Jordan/Egyptian	0.79	0.78	0.78	0.77	0.76	0.77
Japan/France	0.79	0.73	0.77	0.70	0.73	0.75

Table 6
Average results for the network flow (NF) results using 200 instances per gold-standard profile, SVM using 200 training vectors, and Ped05 and Ped05$_{SVD}$ (the best results without and with SVD, respectively). All results are F-measure (the same as accuracy for our method and SVM). The weighted average is calculated based on the number of instances in each pair of names. The best result for each name pair is indicated in **boldface**.

Name Pair	Majority	Ped05	Ped05$_{SVD}$	SVM$_{200}$	NF$_{200}$
Ronaldo/Beckham	0.69	0.73	0.65	**0.85**	0.80
Tajik/Ekeus	0.74	0.96	0.89	0.90	**0.97**
Microsoft/IBM	0.59	0.51	0.59	0.62	**0.73**
Peres/Milošević	0.56	**0.97**	0.94	0.90	0.96
Jordan/Egyptian	0.54	0.59	0.62	0.72	**0.77**
Japan/France	0.51	0.51	0.50	0.48	**0.75**
Unweighted Average	0.61	0.71	0.70	0.75	**0.84**
Weighted Average	0.53	0.55	0.55	0.55	**0.77**

performance is influenced by the amount of data used to construct the gold standard profiles.[9] To reduce the impact of possible skewed sampling of training data, we repeat the random sampling five times, with no overlap between the random samples. We report the performance of each sample set as well as the average over the five samples.

5.2 Results and Analysis

5.2.1 Initial Experiments. Table 5 shows the performance of our method over five random samples of 200 training instances per task. Observe that the performance over the five rounds varies very little (a maximum difference of 0.08, and most are much closer). This shows the robustness of our method to different make-ups of training data. Table 6 shows the average performance of our method, in comparison to the chance (majority) baseline, as well as the results produced by the unsupervised method of Pedersen, Purandare, and Kulkarni (2005) (with singular value decomposition [SVD] reported as

[9] We also experiment with 400 training instances to see whether increasing the amount of training data helps. The performance benefit is minimal: two tasks have the same average performance, three improve by 1%, and one by 2%, with an improvement in the average over all the tasks of 1.25%. A paired t-test between the results on 400 and 200 training instances yields a high p value ($p = 0.73$), indicating that the differences between the two are statistically insignificant.

Ped05$_{\mathrm{SVD}}$, and without SVD as Ped05), and the supervised SVM on the same training data as our method. Observe that our method not only significantly outperforms the random baseline, it is moreover the best performer among all the methods (paired t-test, $p < 0.05$).

There are cases for which Pedersen, Purandare, and Kulkarni's (2005) methods have at best chance performance (*Microsoft/IBM* and *Japan/France*). The authors suggest that these pairs of names arise in the context of news text in which there are "no consistently strong discriminating features" useful in the clustering algorithm. (Interestingly, this is the case even with SVD, where words are grouped into a small number of unnamed concepts.) Even the SVM has difficulty with these pairs, also performing at just around chance. Yet our method performs well above chance for these pairs. In general, SVM produces results that are little better on average than the unsupervised results in Pedersen, Purandare, and Kulkarni (2005) (with some tasks performing better, and some worse). This shows that the performance improvement from the network flow method does not depend solely on access to training data. Instead, it seems that the use of ontological relations in calculating distance can significantly enhance the discriminatory power over simply using words.

Note that there is one difference between the data used in the SVM and the network flow experiments: The SVM is trained using all words as features, while only WordNet noun concepts are used in the network flow experiments. It is possible that using just nouns or a mapping of nouns to WordNet concepts could bring the performance of the SVM into line with our network flow measure. We thus perform two replications of the SVM experiments, one using only nouns as features and one using noun concepts as features (with the relevant frequencies as the feature values in both cases). However, both of these approaches produce little to no improvement over the all-words results reported in Table 6. We conclude that our network flow method is superior to, and more consistent than, the purely distributional methods, and that this difference is attributable to the integration of distributional and ontological (relational) information in our measure.

5.2.2 Reducing the Amount of Training Data. Because, in contrast to Pedersen, Purandare, and Kulkarni (2005), we use a supervised approach, we want to determine whether we can reduce our dependence on training data. Here, we report experiments using one-half (100 instances) and one-quarter (50 instances) of the training data used earlier. As before, we repeat the random sampling of the training instances five times in each case, and report the average performance here.

Table 7 shows the network flow performance for 200, 100, and 50 training instances. Numerically, the results do not differ by much when the training data is reduced from 200 to 100 instances, and a paired t-test finds the difference to be non-significant. The performance drop is more pronounced in the 50-instance experiment, where every pair of names shows some drop in performance compared to 100 instances. Here, a paired t-test shows that the performance drop in the 50-instance experiment is statistically significant ($p = 0.04$). Despite this, we still outperform the other methods: Our results using 50 training instances are much better than those of Pedersen, Purandare, and Kulkarni (2005) in all but one task, and even better overall than the SVM in 200 training instances (compare the SVM column of Table 6).

For comparison, we also train the SVM in 100 training instances, and find a decrease of 3% on average from using 200 training instances. We conclude that our method is more robust to minimal training conditions. To explore the least amount of training data needed for our measure, we further reduce the amount for producing gold-standard

Table 7
Average classification results of the network flow method using 200, 100, and 50 training data per classification task. The weighted average is calculated based on the number of test instances per task.

Name Pair	Number of Training Instances		
	200	100	50
Ronaldo/Beckham	0.80	0.79	0.76
Tajik/Ekeus	0.97	0.98	0.96
Microsoft/IBM	0.73	0.73	0.72
Peres/Milošević	0.96	0.97	0.94
Jordan/Egyptian	0.77	0.74	0.70
Japan/France	0.75	0.75	0.70
Unweighted Average	0.83	0.83	0.80
Weighted Average	0.77	0.76	0.72

profiles to 20 and 5 instances per task, and observe a continual drop in performance. The performance of one task (*Ronaldo/David Beckham*) drops below chance with 20 training instances and another (*Microsoft/IBM*) drops below chance with 5. For this set of data, we conclude that 50 instances per task are required to provide enough discriminatory power for our method.

Although unsupervised methods have the advantage of requiring no training data, in our case, 50 to 100 training instances constitute only a very small portion of the data, as well as a small amount of annotation effort in absolute terms. We conclude that the (small) labelling effort is justified by the performance gain achieved using our minimally supervised approach.

6. Task 3: Document Classification

Document classification is an NLP task in which a previously unseen document is given a topic label (or a set of such labels) based on its subject matter. For example, a financial document discussing the fluctuation of crude oil prices may be labelled "commerce" or "crude oil" in the Reuters Corpus (Lewis et al. 2004). In our version of the task, each document has a single topic label. Document classification is typically performed by comparing the text of an unlabelled document to the text of documents whose topics (labels) are known, and assigning the label of the closest such document (Joachims 2002; Iwayama et al. 2003; Esuli, Fagni, and Sebastiani 2006; Nigam, McCallum, and Mitchell 2006). This task is thus similar to the name disambiguation task in the previous section, and our approach is similar as well: Here again, we form gold-standard profiles from a small collection of texts of known classes, and then compare each test instance to each of the gold-standard profiles. As in name disambiguation, we experiment with different amounts of training data for creating the gold-standard profiles.

There are two differences of note in comparison to name disambiguation. First, in document classification we use the entire set of words constituting the document to create a semantic profile, rather than a smaller window around a target word. Second, whereas each ambiguous name instance in the earlier task had exactly two potential labels (and thus there were two gold-standard profiles for comparison), the number of labels in the document classification task is much larger, leading to more ambiguity in the task.

6.1 Experimental Set-up

6.1.1 Corpus Data. Our data is a corpus of articles from 20 different Usenet newsgroups released by Mitchell (1999). Because each newsgroup corresponds to a topic, the articles can be classified using the (single) newsgroup label. We use the collection maintained by Rennie (2001), in which all the duplicates (cross-posts) are removed, resulting in 18,828 articles. The articles are approximately evenly distributed among the 20 newsgroups. Stopwords and article headers are removed before processing each text.

Work that relies on word frequency vectors to represent the texts in document classification has revealed the importance of preprocessing the word frequency data to emphasize those terms that are likely to be most meaningful. For example, word frequencies have typically been weighted by inverse document frequencies ($tf \cdot idf$) to lessen the impact of very common but less distinguishing words. According to Rennie (2001), their best system on the same corpus uses the $\frac{\log tf+1}{\log idf}$ weighting scheme. In order to compare our system to theirs, we use this same word weighting scheme in the creation of the word vectors that are used to produce our semantic profiles. (We have experimented with using raw word frequencies as well as $tf \cdot idf$ to produce profiles. Both methods yield approximately the same results as the $\frac{\log tf+1}{\log idf}$ frequency weighting scheme.)

6.1.2 Training and Evaluation. As mentioned before, we treat the classification task similarly to name disambiguation, taking a minimally supervised approach. We randomly select a small number of documents as training data for creating the gold-standard semantic profiles. We use 10 or 30 documents per newsgroup, or approximately 1–3% of the documents. The remaining documents are used as testing data. Again, we use a random sample of documents for each gold-standard profile, repeated five times to minimize the impact of a possible skewed sampling. We report the average accuracy over the five samples.

Because there are 20 possible topic labels, the random baseline is very low, at 5%. (Using the predominant label raises this only slightly.) A more informative evaluation of our method is to compare to a state-of-the-art approach that is purely distributional. A comparison to Rennie (2001) is natural, since we use the same data set. However, they trained an SVM on 30 documents per class and tested on 10% of the documents, repeated 10 times. Because our training approach differs somewhat (training on 10 or 30 documents per class, testing on all remaining documents, repeated 5 times), we also replicate their SVM experiment using our training and test sets. As in the name disambiguation task, we use the LIBSVM software package (Chang and Lin 2001) and tune the classifier in the training phase for the best SVM parameters prior to the testing. Also as in our name disambiguation task, we additionally train and test the SVM on just the nouns in a document (rather than all words), and also on the nouns mapped to concepts (with the relevant frequencies as the feature values in both cases). Thus we report results of the SVM on three different types of input frequency vectors: all words, nouns, and concepts.

6.2 Results and Analysis

6.2.1 Initial Results. Table 8 presents the classification results using 10 and 30 training documents per class for our network flow and SVM methods. Our network flow method performs well above the random baseline, but is far from achieving state-of-the-art results. The SVM experiments using all words in the document perform much better than our network flow method, and are consistent with the accuracy of 68.7% achieved

Table 8
Average classification results using 10 and 30 training documents per newsgroup.

Training Size / Class	SVM All words	SVM Nouns (Words)	SVM Noun Concepts	NF
10	47.1	47.8	42.7	31.2
30	66.2	66.4	61.4	32.0

by Rennie (2001) using an SVM. One possible reason is that the SVM is trained on all words (minus stopwords and article headers), whereas our network flow method applies to noun concepts only. The SVM performance on noun-only data is similar to that of all words. Although there is a marked decrease in performance on the concept frequency vectors, SVM still outperforms our method.

The poorer SVM performance on concept frequencies suggests that concept frequency vectors are less easily distinguishable than word frequency vectors. Recall, however, that we found no difference with these various training approaches for SVM in name disambiguation. It is possible that the mapping from words to concepts is a problem here because the full text is used, rather than a relatively small window around a target word. Because each word can map to multiple (potentially unrelated) concepts, the use of a larger, unconstrained bag of words may lead to a high degree of ambiguity, introducing more noise in the semantic profile than our method can handle. This may also explain why the network flow method does not improve with additional training data, showing virtually no improvement between 10 and 30 training instances (0.8% difference). We speculate that the amount of noise in a semantic profile based on the larger amount of text may increase along with the increase in the training size, offsetting any potential gain from having additional data.

If this hypothesis is correct, it is natural to ask why the SVM result using concepts shows a substantial increase in accuracy from 10 to 30 training documents. If larger texts yield nosier semantic profiles, why does this not negatively affect the SVM as well? This highlights a fundamental distinction of our approach: our method is novel because it finds the distance between concepts *as embedded in a graph (the ontology)*, not just between concept *vectors*. Generally, our thesis is that this is an advantage of our model: It entails that all concepts generated from a text play a role in determining the distance of that text from another. As we noted earlier, this allows us to find similarity between texts that use related but not equivalent concepts. However, the performance of our method in this document classification task reveals a potential drawback of this property of our method. Because it takes all concepts into account in determining distance, it is more susceptible to noise. Figure 5 illustrates the problem. We see that the square and triangle profiles are noisy—that is, they each have a number of nodes that are not part of their coherent semantic content. These noisy aspects of the two profiles are less separated in ontological space, making the two profiles more similar according to our measure than their "true" semantic content would indicate. Because a vector representation of concepts does not form connections between differing concepts, it is not led astray in the way our method is.

6.2.2 Removing Noise from the Profiles. Our conjecture is that the poor performance of our network flow method is due to noise caused by ambiguity in the mapping of each word to all of its concepts (i.e., not just the relevant ones to the topic). This effect could also be

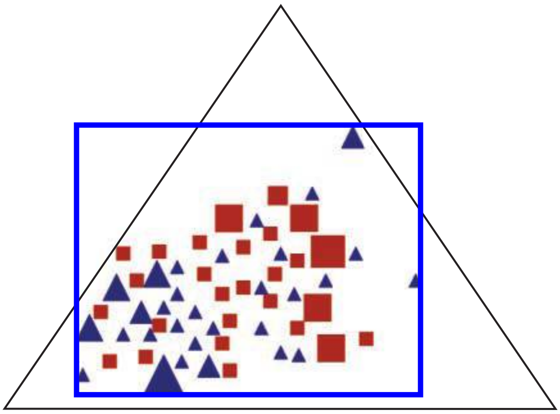

Figure 5
Two noisy profiles, one represented by squares, the other by triangles.

exacerbated by the fact that, in using the full document, we may have a higher number of less relevant words than when a profile is formed from a more constrained set of words (as in verb alternation detection and name disambiguation). If this hypothesis is true, then the noisy (irrelevant) concepts should be distributed within each profile according to some prior probability distribution. If we knew that distribution, then we could "subtract out" the noise and form more semantically coherent profiles. Referring to Figure 5, the idea is that we would like to remove the small, dispersed squares and triangles, leaving only the larger ones that form a semantically more coherent set.

We test this idea, experimenting with two possible noise distributions. The first is simply the uniform distribution, and the second is a distribution determined empirically using frequency counts from a domain-general corpus. For the latter, we determine a distribution over concepts based on the nouns in the BNC. Because the BNC is a balanced corpus, the distribution of its nouns can be considered a prior that is treatable as noise compared to the distribution in a newsgroup posting that is specific to a particular topic. In each case, we create a semantic profile representing the expected noise, and then "subtract" the resulting noise profile from each of our gold-standard semantic profiles in the document classification task. The "subtraction" is actually a process of setting to zero all of the semantic profile frequencies that are less than the noise value for that concept. Any node with a value higher than the noise value for that node is expected to be a potentially relevant concept. We leave such nodes at their original value so that they are more distinguished from the remaining values (now set to zero). Figure 6 illustrates the result of applying this kind of noise reduction to the profiles in Figure 5. We can see that low-frequency concept nodes are zeroed out, with higher frequency nodes maintaining their concept weight.

Table 9 presents the network flow results on the noise-subtracted data, showing a 3–5% increase in the performance using 30 training documents per class. The performance decreases with noise-subtraction when we have only 10 training documents per class, suggesting that there may not be enough data in this case to use this simplistic subtractive method.

Interestingly, subtracting the uniform noise distribution from the profiles has a more favorable effect than subtracting the BNC noise distribution. The BNC distribution is perhaps inappropriate for our data. Newsgroup data includes a variety of subjects

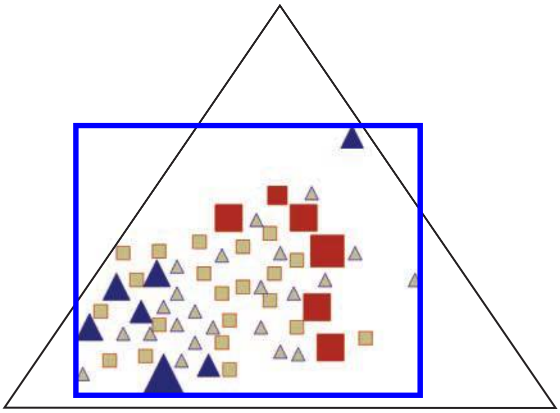

Figure 6
The same two profiles in Figure 5. The profile masses that are "subtracted" are shaded in gray.

which may make it more similar to a balanced corpus than we have originally anticipated, thus what we are treating as a "noise" distribution in this case may not actually represent noise. That said, there is a small but notable increase even using the BNC noise distribution when we have sufficient training data. The idea of subtracting out noise seems promising, but we leave the appropriate representation of noise, and the mechanism for removing it effectively, as an area of future research.

7. Profile Density: A Measure of Coherence of Semantic Profiles

We have seen a performance difference across the three tasks we used in evaluation: the network flow method outperforms purely distributional measures on verb alternation detection and name disambiguation, but does poorly on document classification compared to a distributional approach. (See Table 10 for a summary of the results.) We use the same ontology (WordNet) and the same concept distance (number of edges) in our network flow measure across all three tasks, hence there must be some difference in the three data sets themselves that impacts the ability of our method to distinguish the semantic profiles corresponding to one class of data (one usage of an ambiguous name, for example) from the profiles of a different class of data (the other usage of the name). In this section, we develop a measure that can capture this property and explain the performance differential we have observed for our method.

Table 9
Average classification results using 30 and 10 training documents per newsgroup, using the original profiles (NF), and using profiles after the "noise subtraction" process described in the text ("NF − Uniform" and "NF − BNC" are results subtracting the uniform distribution and the BNC noun frequency distribution, respectively).

Training Size / Class	NF	NF − Uniform	NF − BNC
10	31.2	28.2	27.4
30	32.0	37.2	35.6

Table 10
Summary of task-based results. The numbers in parentheses indicate the number of training instances used. The best result for each task is shown in **bold**.

Verb Alternation Detection	random	Manhattan	skew div	JS	NF
Dataset1 Avg	0.50	**0.70**	0.57	0.67	**0.70**
Dataset2 Avg	0.50	**0.67**	**0.67**	**0.67**	**0.67**

Name Disambiguation	random	SVM (100)	SVM (200)	NF (100)	NF (200)
Unweighted Avg	0.61	0.72	0.75	**0.83**	**0.83**
Weighted Avg	0.53	0.52	0.55	0.76	**0.77**

Document Classification	random	SVM (10)	SVM (30)	NF (10)	NF (30)
20 newsgroups	0.05	0.43	**0.61**	0.31	0.32

7.1 Profile Coherence

Our goal is to find a property of individual semantic profiles that, when averaged across the profiles in a data set, indicates whether our method will be able to distinguish profiles of different classes in that data set. That is, we aim to learn about the overall separability of the classes in a data set by investigating the properties of individual profiles that constitute the data set. Our hypothesis is that the important factor for our method is what we refer to as **profile coherence**: the degree to which profile mass is concentrated within a constrained space (or set of constrained spaces) of the ontology. The more spatially coherent the sets of weighted concepts are for the profiles in a data set, the more likely it is that our method will be able to distinguish contrasting profiles. Conversely, less coherent profiles, whose frequency mass is more distributed across a wider area of the ontology, will be more difficult to separate into classes. (Note that profile coherence is not a sufficient condition for data separability, but we hypothesize that it can be a useful indicator.)

For example, consider the square and triangle profiles in Figure 7. Coherent profiles have their profile mass (the concept weights) focused within small, distinct regions of the ontology, as in Figure 7(a). These types of profiles tend to be highly distinguishable from each other. Less coherent profiles, whose mass is more dispersed through the ontology, such as those in Figure 7(b), are likely to be less distinguishable. Note, however, that it is not simply occupying greater or fewer nodes in the hierarchy that determines profile coherence (and distinguishability). The profiles in Figure 7(c) are "spread out" as in (b), but are more coherent (and distinguishable) due to having areas of high mass.

The considerations illustrated in Figure 7 suggest that both distributional and ontological factors contribute to the coherence of a semantic profile, and that we must determine a suitable measure of coherence that captures both factors. A simpler, alternative hypothesis is that either purely distributional or purely ontological factors may sufficiently capture the coherence of a semantic profile. To explore these ideas, we examine different ways to assess the coherence of the semantic profiles in our example data sets. We develop various measures of coherence, and then evaluate whether the degree of coherence as determined by each measure indeed corresponds to the performance of

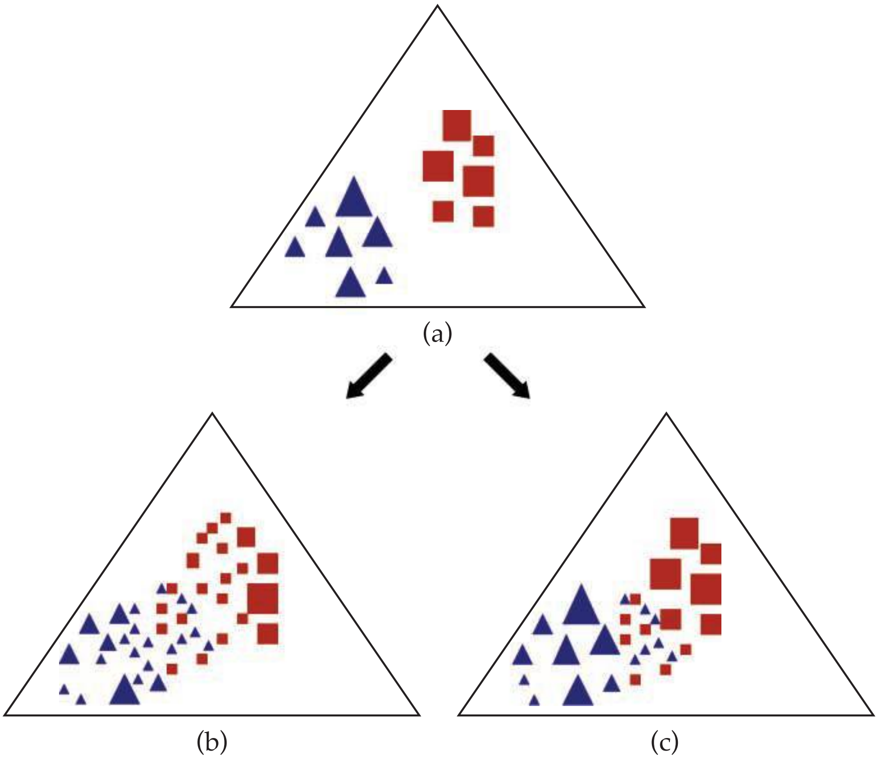

Figure 7
Examples of two profiles (indicated by squares and triangles) of varying coherence. The profiles in (a) are more distinguishable than those in (b) and (c); the profile in (c) is in turn more distinguishable than that in (b). The degree of distinguishability of these profiles is reflected in their degree of **coherence**.

our network flow method on the data sets in our three tasks. We expect a useful measure of profile coherence to have a high average value across the data sets on which we perform well (verb alternation and name disambiguation), and a low average value across the data set on which we perform poorly (document classification).

In Section 7.2, we briefly review several measures intended to separately capture the distributional or ontological coherence of a semantic profile. We show that such measures are insufficient for accounting for the performance differences of our method across the data sets. In Section 7.3, we develop a novel measure to capture the coherence of our profiles in terms of both distributional and ontological information. This measure, called **profile density**, expresses the degree to which a semantic profile forms a coherent clustering of weighted concepts in an ontology. We demonstrate that our profile density measure can account for the performance differential across our data sets.

7.2 Separate Distributional and Ontological Approaches

We explored several (unsuccessful) means for capturing profile coherence with a purely distributional or purely ontological measure. Although we could not exhaustively investigate all possible measures of this kind, the underlying reasons for the lack of success of these measures in explaining the differing performance of our method across the data sets convinced us of the need for a measure that integrates distributional and ontological factors (which we present in the following section). We mention the single-factor measures here for completeness.

7.2.1 Potential Distributional Coherence. Recall that Section 6.2.2 shows that removing the "noise" distribution from each profile improves the document classification performance of our method. In other words, subtracting the noise distribution from a profile can make it distributionally more distinct from other profiles. Based on this observation, we hypothesize that the less a profile resembles a noise distribution over the ontology, the more coherent it is—that is, the more likely the frequency mass is situated in meaningful clusters of concepts. To test this hypothesis, we calculate the average distance (using KL-divergence [Kullback and Leibler 1951]) of the profiles in a data set from a profile created from a noise distribution (the uniform distribution of words, or their distribution in the BNC, as in Section 6.2.2). Higher values of this measure indicate further distance from the noise distribution.

7.2.2 Potential Ontological Coherence. Here we consider two observations. First, we hypothesize that profiles with fewer concepts are more coherent, because a smaller number of concepts is more likely to be less dispersed in the ontology. We simply use average profile size to capture this property (here, smaller values of profile size indicate greater coherence). Second, we hypothesize that profiles whose concepts have greater specificity are more coherent, because use of less specific concepts is indicative of vagueness and potential lack of coherence. Because specificity corresponds well to depth in WordNet, we use a simple measure of average profile depth to indicate the specificity of the set of concepts in a profile (here, greater values of depth should indicate greater coherence).

7.2.3 Analysis of the Single-Factor Measures. For each task, we calculate the average of each of the hypothesized distributional and ontological coherence measures over the profiles in the data set, and find that there is no consistent correspondence with the performance of our network flow method across the tasks. Despite the intuitions and observations presented herein, these results are not surprising. For example, the profiles of a data set may all be distributionally very similar overall to the noise profile, supposedly indicating low coherence, but they may be quite coherent in the actual ontological space they occupy. Similarly, the profiles in a data set may all have a small average depth in the ontology or large size (again supposedly indicating low coherence), but their distributional properties (the weights on the concepts that are occupied) may yield coherent clusters of mass in the profile. This analysis then confirms our hypothesis that, because distributional and ontological information are intertwined in the representation of a semantic profile, a useful measure of profile coherence must take into account an integration of these two information sources.

7.3 Integrating Distributional and Ontological Factors in a Coherence Measure

As noted earlier, and tentatively confirmed by the results herein, we assume that the interaction of distributional and ontological factors determines the coherence of profiles (i.e., a coherent profile has its frequency mass concentrated within a reasonably constrained space [or set of constrained spaces] of the ontology). We observe that this is similar to the geographical notion of population density, which is determined by the population mass divided by the area occupied. Here we extend the geographical

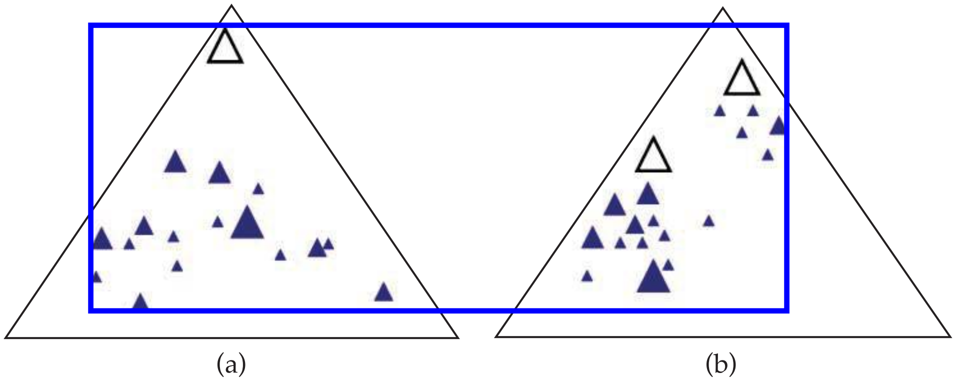

Figure 8
Two examples of profile density within an ontology. The hollow triangles are the common ancestors of the filled triangles, which are concept nodes in the profile. The profile in (a) is fairly dispersed, requiring a single but distant ancestor node. The profile in (b) is more clustered; two ancestor nodes are required but each is close to its descendants.

definition of density within our network framework by relating population mass to distributional weights on concepts, and occupied area to the spread of the weighted concepts in the ontology. We call the resulting measure of profile coherence **profile density**.

7.3.1 The Profile Density Measure. To adapt the definition of geographical density to our problem, we first need to determine the analogs of population mass and occupied area in a semantic profile. The profile mass at each concept node is directly analogous to the population mass. Defining the occupied area within an ontology is not as straightforward, as there is no simple definition of area within a graph. For example, Agirre and Rigau (1996) use the number of nodes within a subgraph as its area, but this fails to take into account how dispersed the nodes are throughout the ontology. We instead develop a definition of area that captures the actual spatial spread of the profile mass through the ontology.

To begin, we note that any subgraph of the WordNet hypernym hierarchy is hierarchical itself. Thus, any region of the ontology that contains some profile mass is a hierarchy rooted at some common ancestor of those profile nodes.[10] As shown in Figure 8, the more dispersed (less closely clustered together) a set of nodes is, the further away their common ancestor is. That is, a highly related (and spatially constrained) set of concept nodes can be generalized to a more specific ancestor concept (i.e., near the descendants, as in Figure 8(b)), whereas a semantically distant set of concepts will be generalized to a semantically general ancestor concept (i.e., far from the descendants, as in Figure 8(a)). The ontological distance between a set of nodes and their common ancestor thus indicates how closely clustered the descendant nodes are.

Next note that any semantic profile can be represented by a set of ancestor nodes, and these ancestor nodes capture the spatial clusterings of the profile mass. For example,

[10] Although WordNet contains instances of multiple inheritance, the rate is low. As a result, the likelihood of a set of profile nodes sharing multiple ancestors is low as well.

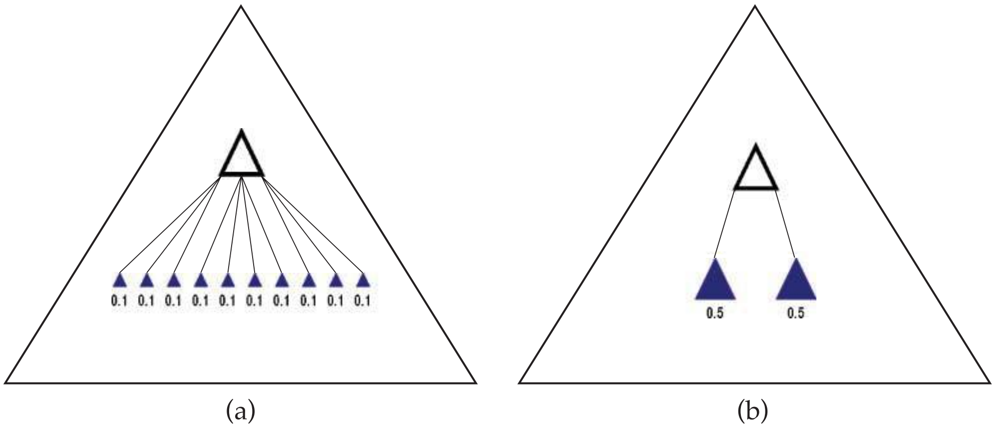

Figure 9
These two profiles have equal density value given our original *profile_density* formula in Equation (5), but are suitable distinguished (with the profile in (b) having higher density than that in (a)) by the *norm_density* formula in Equation (6). See the text for discussion.

the profile in Figure 8(a) is represented by one ancestor node, and that in Figure 8(b) by two such nodes. Combining these observations, we see that given a suitable manner for identifying ancestor nodes to represent a profile, we can use the combined ontological distance between each of those nodes and their descendants as an indication of how closely clustered the concepts of the profile are. We can now complete our definition of profile density by using the total distance between each identified ancestor and its descendants as an indication of the occupied area of the ontology.

Formally, let P be a profile and A be a set of ancestor concept nodes such that each profile node $d \in P$ is guaranteed to have an ancestor $a \in A$. (We will explain in Section 7.3.2 how to find the set A.) The profile density of P is then defined as follows:

$$\mathit{profile\_density}(P) = \sum_{a \in A} \sum_{\substack{d \in P,\\ d \in \mathit{descendant}(a)}} \frac{\mathit{mass}(d)}{\mathit{distance}(d,a)} \tag{5}$$

where $\mathit{mass}(d)$ is the profile mass (concept frequency) at node d, and $\mathit{distance}(d,a)$ is the distance in the ontology between node d and node a, as given by a suitable concept-to-concept distance measure (such as the edge distance that we have used in our task-based evaluations).

There is one more subtle detail we must address. Consider the two examples in Figure 9, where the distance between each ancestor and all its descendants is the same (here, say, a distance of 1), but the distribution of the profile mass differs. The first diagram has ten equally weighted profile nodes, and the second has two. Our current formulation in Equation (5) yields a density of 1 for both diagrams (i.e., $(0.1/1) \times 10 = 1 = (0.5/1) \times 2$). However, the profile mass in diagram (a) is distributed among more nodes than that in diagram (b). Intuitively, the second profile is more densely clustered and should have a higher density value.

Looking more closely at our density formula in Equation (5), observe that the number of profile nodes has an impact on the calculation—that is, density increases as the number of profile nodes increases due to the inner summation in the formula. To achieve an appropriate density measure, then, we normalize the original density value by the number of profile nodes, resulting in a normalized density for a profile:

$$
\begin{aligned}
\mathit{norm\_density}(P) &= \frac{\mathit{density}(P)}{\mathit{sizeof}(P)} \\
&= \frac{1}{\mathit{sizeof}(P)} \sum_{a \in A} \sum_{\substack{d \in P, \\ d \in \mathit{descendant}(a)}} \frac{\mathit{mass}(d)}{\mathit{distance}(d,a)} \qquad (6)
\end{aligned}
$$

Returning to our example in Figure 9, Equation (6) assigns the first profile a normalized density of 0.1, and the second profile a normalized density of 0.5. The modified measure now appropriately distinguishes the two profile densities, indicating that the profile in Figure 9(a) is less tightly clustered than the profile in Figure 9(b).

7.3.2 Finding the Ancestor Set for Profile Density. As noted earlier, our definition of profile density depends on identifying a suitable set of ancestor nodes of the concept nodes in the profile: the aggregate distance of the ancestors to the profile nodes indirectly indicates the degree to which the profile nodes are spatially clustered close together. Thus, given a profile P, we need to find A, the set of nodes that are ancestors of the profile nodes $d \in P$. (The nodes $a \in A$ correspond to the hollow triangles indicated in Figure 8 and Figure 9.) Recall that these ancestor nodes are intended to be a set of concepts that serve as an appropriate generalization of the nodes in the profile—each ancestor in a sense represents a coherent cluster of profile nodes. However, we do not know a priori what the appropriate level of generalization is—we simply want a level that gives a useful assessment of how clustered together the profile nodes are.

For this purpose, we make use of Clark and Weir's (2002) method for generalizing a set of weighted concept nodes in an ontology. As we noted in Section 4, given a frequency distribution over all concept nodes, Clark and Weir use a statistical method to search for the set of nodes (i.e., our node set A) that best generalize the original weighted concepts. This method is particularly appropriate for our purposes because it includes a parameter, $\alpha \in (0, 1)$, that controls the level of generalization. We vary α over five values (0.05, 0.25, 0.5, 0.75, and 0.95) to obtain five different (more to less generalized) sets of ancestors. In our analysis, we calculate the density using each ancestor set in order to evaluate the impact of the precise choice of ancestor nodes on our measure.

7.3.3 Results and Analysis. For each of the three tasks in our earlier task-based evaluation, we calculate the profile density of the corresponding data set. We define the profile density of a data set to be the average of the normalized density values over its profiles. For the verb alternation detection task, we perform the analysis on all 240 profiles used in the task (120 verbs, with 2 profiles per verb, one for the subject slot, one for the object slot). In the remaining two tasks, because each instance profile is compared to a gold-standard profile, we believe that the performance depends primarily on the coherence of the gold-standard profiles. We thus perform our analysis on the gold-standard profiles only. For name disambiguation, we have 60 profiles (5 samplings with 12 gold-standard profiles each); for document classification, we have 100 profiles (5 samplings with 20 gold-standard profiles each). For each profile, we calculate the normalized density using each of five ancestor sets (based on the α value, as noted above). For the concept-to-concept distance measure, $\mathit{distance}(d, a)$ in Equation (6), we use edge distance, the same measure used in the tasks in earlier sections of the paper.

Table 11
The profile density scores for each data set at five different values of α, as well as the average scores across the α values.

α value	0.05	0.25	0.5	0.75	0.95	Avg
Verb Alternation	5.59e-4	5.90e-4	6.32e-4	7.14e-4	8.87e-4	6.76e-4
Name Disambiguation (200)	8.93e-5	9.89e-5	1.08e-4	1.18e-4	1.35e-4	1.10e-4
Name Disambiguation (100)	1.11e-4	1.26e-4	1.38e-4	1.52e-4	1.78e-4	1.41e-4
Doc Classification (30)	5.25e-5	5.94e-5	6.59e-5	7.43e-5	8.78e-5	6.80e-5
Doc Classification (10)	8.03e-5	8.85e-5	9.87e-5	1.11e-5	1.33e-5	5.84e-5

We expect that, if our profile density measure does indeed reflect the coherence of a data set, then we will see a correspondence between the density values and the performance of our network flow method. Higher density values indicate a profile whose weighted concepts form more coherent clusters in the ontology. Specifically, then, we expect higher density values for the data sets from our verb alternation detection and name disambiguation tasks (on which our method had better performance than distributional methods), and lower density values for the document classification data set (on which our method had worse performance than a purely distributional method).[11]

Table 11 shows the profile densities of each data set. First note that the density values are relatively stable across all values of α, indicating that the precise level of generalization is not critical to the usefulness of our density measure. Next, observe that, as predicted, the document classification data set is shown to have the lowest density for both training set sizes. This observation is in accord with our hypothesis that the profile density measure indicates the coherence of the profiles in a data set and is therefore informative about the network flow performance on that data set.

Interestingly, we also observe that, across all values of α and training set sizes, the verb alternation data set has the largest densities, followed by the name disambiguation data set, then the document classification data. (The differences between all three data sets are statistically significant, $p \ll 0.05$.) This result might stem from the fact that there are varying degrees of constraint placed upon the data in the three tasks. In verb alternation, the nouns used to generate a profile appear either all in the subject or all in the object position of the target verb. In name disambiguation, we loosen the restriction to include all nouns in a small window surrounding the target word. Lastly, in document classification, the only restriction on the nouns used to generate a profile is that they appear in the same document. This suggests that the syntactic and semantic constraints placed upon a set of nouns can have an impact on the coherence of the profile created from them.

This latter observation suggests that our profile density measure may be useful not only in indicating the ability of our network flow method to distinguish relevant profiles. More generally, it may also reflect the varying degrees of syntactic and semantic

[11] Note that because our method in each task is compared to different kinds of alternative distributional methods, we do not expect to find a mathematical correlation between the performance improvement and the density values; rather, good performance should be reflected in higher density values and poor performance in lower density values.

constraints placed upon the set of words that generate a profile. Our profile density measure may indeed be generally useful as a measure of the semantic coherence of a set of concepts in an ontology (Gurevych et al. 2003), a matter we plan to explore in future work.

In summary, our analysis in this section has shown that both distributional and ontological properties contribute to the coherence of a profile, but neither alone is indicative of the network flow performance in a particular task. Our new measure of profile density serves as a tool for analyzing profiles that integrates their distributional and ontological coherence, and provides a post hoc means for explaining the performance differential of our method across the different tasks we performed here. The results also point to the possibility of devising a diagnostic tool for the suitability of the network flow method on novel data. An analysis of the data and results across a larger set of tasks will allow us to investigate the possibility of determining a density threshold that would be indicative of expected positive results with our method.

8. Related Work

To the best of our knowledge, our method is the only work that measures text distance by combining ontological knowledge and distributional information together via a graph-based algorithm. Although there are existing methods that use either or both types of information in measuring the semantic distance of texts (Corley and Mihalcea 2005; Mohammad and Hirst 2006), our work is unique in that it integrates the ontological distance between individual words across two texts as well as the distributional differences between the texts. Here we review existing work on both text comparison and graph-based approaches in CL, given the relevance of these two areas to our research.

8.1 Text Comparison

Our work stems from the studies on measuring the semantic distance between two words or concepts using an ontological resource (which is extensively covered in Pedersen, Banerjee, and Patwardhan [2005] and Budanitsky and Hirst [2006]). To extend these methods for the comparison of two texts, we incorporate ontological distance between concepts and distributional information in a systematic and efficient manner. Other research that attempts to include the two takes a more modular approach. For example, Corley and Mihalcea (2005) consider the ontological distance between the concepts representing the texts but ignore their distributional information. On the other hand, Scott and Matwin (1998), McCarthy (2000), and Mohammad and Hirst (2006) take the distributional distance between concept vectors representing the texts but do not consider the ontological relations among the concepts.

Most recent work on text comparison tends to be word-based and distributional (Lee 2001; Weeds, Weir, and McCarthy 2004; Pedersen, Purandare, and Kulkarni 2005; Al-Mubaid and Umair 2006). In the case of high dimensionality and data sparseness, words are grouped into a smaller number of (unnamed) concepts using some matrix factorization technique (e.g., SVD) or some clustering method (Pereira, Tishby, and Lee 1993). In other words, words are grouped together based on their distributional properties instead of their explicit semantic/ontological properties. Furthermore, unlike in our method, once the words are collapsed into unnamed concepts, the individual elements (i.e., the unnamed concepts) across data points cannot be compared. As shown in our experiments, taking into account this extra piece of information is beneficial for some applications.

8.2 Graph Approaches

In recent years, we have seen an increasing use of graph-based methods in NLP (Pang and Lee 2004; Mihalcea 2005; Navigli and Velardi 2005). The graph-theoretic approach is popular due to the elegance of representing appropriate NLP problems and the availability of a number of efficient algorithms. One of the most straightforward NLP examples is the use of WordNet. Besides our work here, much prior research has taken advantage of the graphical structure of WordNet. For example, Agirre and Rigau's (1996) conceptual density uses WordNet as a graph and calculates the density within a subgraph (the number of relevant concepts within a subgraph), which was found to be useful for WSD.

Graphs in general are the obvious mathematical formalism to encode the relationships (represented as edges) between either words or longer units of text (represented as nodes). (The reverse is possible, using nodes to represent relations and edges for semantic entities. The choice of representation clearly depends on the NLP task itself.) Once we formulate a problem into a standard graph problem, there are existing efficient graph-based algorithms that we can use to find an optimal or near-optimal solution. For example, both Pang and Lee (2004) and Barzilay and Lapata (2005) use a minimum-cut algorithm for two vastly different applications, document polarity classification and content selection, respectively. In these approaches, the sentences are represented as nodes in a graph, and the edge connecting each pair of nodes is weighted with an association score between the sentences, reflecting, for example, the distance (number of sentences) between a pair of sentences. The minimum-cut method allows them to classify the nodes, and thus the sentences, into different categories.

Another popular graph method is the random walk algorithm, which is successfully employed by the PageRank approach for ranking Web pages (Brin and Page 1998). Similar to the minimum-cut algorithm, here, nodes represent semantic entities (e.g., words), and edges represent associations between the nodes (e.g., word co-occurrence). The random walk algorithm allows for the classification of each node based on the relevance of its neighbors. For example, Mihalcea (2006) uses random walk for WSD by constructing a graph in the following way. Each node represents an ambiguous (test) word, or a (training) word labelled with one of its senses. Each edge indicates that the corresponding two words co-occur in some context. The sense of an ambiguous word is determined by the sense of its most relevant neighbor(s), by randomly traversing the graph until an equilibrium state has been reached. Hughes and Ramage (2007) also use a random walk method, with the goal of determining semantic relatedness between individual words (not sets of words, as in our work). In their work, the random walk method computes a probability distribution over WordNet concepts. Note that the probability distributions resulting from random walks centered at different concepts in WordNet are distinct. One can then measure the semantic relatedness between two concepts by calculating the divergence between their probability distributions over WordNet concepts as a result of the two random walks centered at them.

In comparison to other graph approaches to NLP, we choose to use a minimum-cost flow algorithm based on our graph formulation. Because a profile is a collection of frequency-weighted concepts, some concept nodes are weighted more heavily than others, therefore the routes between such nodes across the two profiles are also weighted more heavily. An algorithm solving a minimum-cost flow problem provides an efficient mechanism to find these weighted routes as our solution, making MCF, rather than the shortest paths or maximum-flow-minimum-cut, the best choice for formalizing the constraints we define in the text comparison problem.

9. Conclusion

We have presented a graph-theoretic approach to calculating semantic distance between two texts, which encompasses both ontological knowledge and distributional information. We have developed a network flow method that takes advantage of the graphical structure of an ontology. Given a suitable ontology, a word frequency vector for a text can be transformed into a frequency distribution over concept nodes. Hence, we treat texts as weighted subgraphs within a larger graph (the ontology). By incorporating the semantic distance between individual concepts, the graphical structure representing the ontology becomes a metric space in which we can measure the distance between subgraphs, weighted by their frequencies.

In this article, we use edge distance exclusively for the individual distance between concepts. Given that the distance between concepts is an integral part of our formulation, and that other sophisticated concept-to-concept distances have been shown to outperform edge distance for comparing concepts (Jarmasz and Szpakowicz 2003; Weeds 2003), we also investigate the use of such distances. However, incorporating them can lead to a quadratic growth in complexity. To remedy this, a pre-processing step is required to reduce the complexity to reasonable computation time. In Tsang and Stevenson (2006), we introduce one such method by performing a graph transformation on the original network prior to the network flow calculation. The transformed network is more efficient to process without compromising the performance accuracy. We refer the reader to that paper for further information.

In the task-based evaluation presented here, our method has been shown to provide superior performance on verb alternation detection and name disambiguation, in comparison to alternative distributional approaches—even in cases where the alternative methods have attempted to incorporate additional semantic knowledge (McCarthy 2000; Pedersen, Purandare, and Kulkarni 2005). Unlike existing distributional distances and clustering techniques, the use of our text representation as well as the integration of ontological distance allows a systematic way of capturing appropriate semantic distinctions between the texts in these tasks.

In contrast, our method does not perform as well on document classification as a state-of-the-art machine learning algorithm using a purely distributional approach. In order to examine the performance discrepancy across tasks, we explore measures of the coherence of the profiles in a data set, as potential indicators of how easily semantic profiles of different classes can be distinguished. The purely distributional and purely ontological indicators we consider are not useful in explaining the relatively poor performance of our method on document classification. In response, we develop a measure of profile coherence, called **profile density**, that integrates these factors by determining the degree to which a profile forms distributionally and ontologically coherent clusters of concepts. As a result, we are able to explain the performance of our method on the data sets in terms of their density values.

Recall also that we saw a performance difference in the verb alternation task depending on the different method used to generate the semantic profiles from the bag of words of the text (i.e., using "raw" data, versus a method to generalize to the best set of concepts for the bag of words). Given also that we found that the profiles in document classification have a low density (i.e., their concepts are overly dispersed), one focus for future work will be to explore further means for generating profiles that best capture the intended senses of the words within the text. One option may be to use Mohammad's (2008) unsupervised method for building concept vectors from word frequency data, which focuses the frequencies onto the most likely senses of the words according to coarse ontological categories.

Another strand of future work relates to our profile density measure. We suggest that not only is our profile density useful in predicting the performance of our network flow method on unseen data, it may also be useful for measuring the semantic coherence of a text. Note that a text that is semantically coherent tends to form profiles with highly frequent and highly related concepts within an ontology. Coincidentally, our profile density formulation measures the overall relatedness, and thus coherence, of a collection of concepts by taking into account the distance between the concepts as well as the frequency distribution. For example, if we relax the notion of a text to include verbal arguments, semantic coherence of a text can be thought of as the selectional preference strength a verb imposes on its arguments. As future work, we intend to investigate profile density as an indicator of selectional preference strength. Generally, we believe profile density may offer a quantitative measure for semantic coherence and other related NLP applications.

Acknowledgments

We would like to thank our colleagues in Toronto, in particular Afsaneh Fazly and the CL research group at the University of Toronto, for helpful discussions. We would also thank the anonymous reviewers for their detailed comments. We gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), Ontario Graduate Scholarship (OGS), and the University of Toronto.

References

Agirre, Eneko and German Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 12th International Conference of Computational Linguistics (COLING-1996)*, pages 16–22, Copenhagen.

Al-Mubaid, Hisham and Syed A. Umair. 2006. A new text categorization technique using distributional clustering and learning logic. *IEEE Transaction on Knowledge and Data Engineering*, 18(9):1156–1165.

Barzilay, Regina and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the Joint Conference on Human Language Technology / Empirial Methods in Natural Language Processing (HLT/EMNLP)*, pages 331–338, Vancouver, Canada.

Bodenreider, Olivier. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Brin, Sergey and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Briscoe, Ted and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th Applied Natural Language Processing Conference (ANLP)*, pages 356–363, Washington, DC, USA.

Briscoe, Ted and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1499–1504, Las Palmas, Spain.

Budanitsky, Alex and Graeme Hirst. 2001. Semantic distance in Wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, pages 29–34, Pittsburgh, PA, USA.

Budanitsky, Alex and Graeme Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

Burnard, Lou. 2000. *The British National Corpus Users Reference Guide*. Oxford University Computing Services, Oxford, UK.

Chang, Chih-Chung and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at www.csie.ntu.edu.tw/∼cjlin/libsvm.

Chvátal, Vašek. 1983. *Linear Programming*. W.H. Freeman and Company, New York.

Clark, Stephen and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Corley, Courtney and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, USA.

Esuli, Andrea, Tiziano Fagni, and Fabrizio Sebastiani. 2006. TreeBoost.MH: A boosting algorithm for multi-label hierarchical text categorization. In *Proceedings of the 13th International Symposium on String Processing and Information Retrieval (SPIRE'06)*, pages 13–24, Glasgow.

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

Gangemi, Aldo, Nicola Guarino, and Alessandro Oltramari. 2001. Conceptual analysis of lexical taxonomies: The case of WordNet top-level. In Chris Welty and Barry Smith, editors, *Formal Ontology in Information Systems: Collected papers from the Second International Conference*. ACM Press, pages 285–296, New York, USA.

Garey, Michael R. and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., New York.

Gurevych, Iryna, Rainer Malaka, Robert Porzel, and Hans-Peter Zorn. 2003. Semantic coherence scoring using an ontology. In *Proceedings of the Joint Human Language Technology and Northern Chapter of the Association for Computational Linguistics Conference (HLT-NAACL)*, pages 88–95, Edmonton.

Han, Hui, Hongyuan Zha, and C. Lee Giles. 2005. Name disambiguation in author citations using a K-way spectral clustering method. In *Joint Conference on Digital Libraries (JCDL'05)*, pages 334–343, Denver, CO, USA.

Hirst, Graeme. 2009. Ontology and the lexicon. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, New York, pages 269–292.

Hughes, Thad and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 581–589, Prague.

Iwayama, Makoto, Atsushi Fujii, Noriko Kando, and Yuzo Marukawa. 2003. An empirical study on retrieval models for different document genres: Patents and newspaper articles. In *Proceedings of the 26th ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 251–258, Toronto, Canada.

Jarmasz, Mario and Stan Szpakowicz. 2003. Roget's thesaurus and semantic similarity. In *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 212–219, Borovets.

Jiang, Jay and David Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on the International Conference on Research in Computational Linguistics*, pages 19–33, Taipei, Taiwan.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines—Methods, Theory, and Algorithms*. Kluwer/Springer, New York.

Kullback, S. and R. A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, (104):211–240.

Lee, Lillian. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, pages 65–72.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

Levina, Elizaveta and Peter Bickel. 2001. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 2, pages 251–256, Vancouver, Canada.

Lewis, David D., Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, (5):361–397.

Li, Hang and Naoki Abe. 1998. Word clustering and disambiguation based on co-occurrence data. In *Proceedings of COLING-ACL 1998*, pages 749–755, Montreal, Canada.

Lin, Dekang. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, pages 296–304, Madison, Wisconsin, USA.

McCarthy, Diana. 2000. Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of Applied Natural Language Processing and North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, pages 256–263, Seattle, Washington, USA.

Merlo, Paola and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):393–408.

Mihalcea, Rada. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Joint Conference on Human Language Technology / Empirial Methods in Natural Language Processing (HLT/EMNLP)*, pages 411–418, Vancouver, Canada.

Mihalcea, Rada. 2006. Random walks on text structures. In *Proceedings of Computational Linguistics and Intelligent Text Processing (CICLing) 2006*, pages 249–262, Mexico City, Mexico.

Mitchell, Tom. 1999. 20 newsgroups usenet articles. `http://kdd.ics.uci.edu//databases/20newsgroups/20newsgroups.data.html`.

Mohammad, Saif. 2008. *Measuring Semantic Distance using Distributional Profiles of Concepts*. Ph.D. thesis, University of Toronto, Canada.

Mohammad, Saif and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 35–43, Sydney.

Navigli, Roberto and Paola Velardi. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7), pages 1075–1086.

Nigam, Kamal, Andrew McCallum, and Tom Mitchell. 2006. Semi-Supervised text classification using EM. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*. MIT Press, Cambridge, MA, pages 33–56.

Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd ACL*, pages 271–278, Barcelona, Spain.

Pantel, Patrick and Dekang Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of Association for Computational Linguistics (ACL-00)*, pages 101–108, Hong Kong.

Pedersen, Ted, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. Technical Report UMSI 2005/25, University of Minnesota, Duluth.

Pedersen, Ted, Amruta Purandare, and Anagha Kulkarni. 2005. Name discrimination by clustering similar context. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 226–237, Mexico City, Mexico.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio, USA.

Pinker, Steven. 1989. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA.

Rennie, Jason. 2001. *Improving Multi-class Text Classification with Naive Bayes*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Resnik, Philip. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Resnik, Philip. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada.

Ribas, Francesc. 1995. On learning more appropriate selectional restrictions. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 112–118, Dublin.

Schulte im Walde, Sabine. 2006. Experiments on the automatic induction of German semantic verb classes. *Computational Linguistics*, 32(2):159–194.

Scott, Sam and Stan Matwin. 1998. Text classification using WordNet hypernyms. In *Proceedings of the COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 45–51, Montreal, Canada.

Tsang, Vivian and Suzanne Stevenson. 2004. Calculating semantic distance between word sense probability distributions. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 81–88, Boston, MA, USA.

Tsang, Vivian and Suzanne Stevenson. 2006. Context comparison as a minimum cost flow problem. In *Proceedings of HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, pages 97–104, New York, NY.

Weeds, Julie. 2003. *Measures and Applications of Lexical Distributional Similarity*. Ph.D. thesis, University of Sussex, Sussex, UK.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference of Computational Linguistics (COLING-2004)*, pages 1015–1021, Geneva, Switzerland.

Wilcoxon, Frank. 1945. Individual comparisons by ranking methods. *Biometrics*, 1:80–83.

Wu, Zhibiao and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA.

Xu, Wei, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 267–273, Toronto, Canada.