Using Semantic Distance to Automatically Suggest Transfer Course Equivalencies

Beibei Yang
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854
byang1@cs.uml.edu

Jesse M. Heines
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854
heines@cs.uml.edu

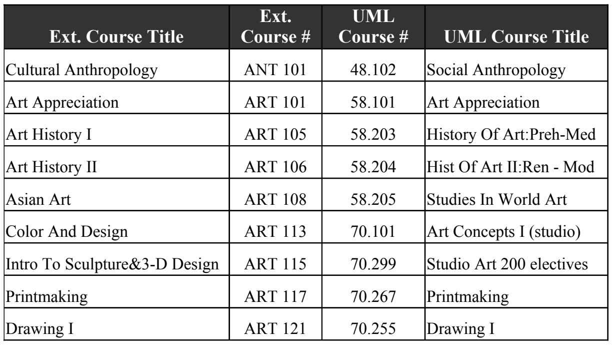
Abstract

Semantic distance is the degree of closeness between two pieces of text determined by their meaning. Semantic distance is typically measured by analyzing a set of documents or a list of terms and assigning a metric based on the likeness of their meaning or the concept they represent. Although related research provides some semantic-based algorithms, few applications exist. This work proposes a semantic-based approach for automatically identifying potential course equivalencies given their catalog descriptions. The method developed by Li et al. (2006) is extended in this paper to take a course description from one university as the input and suggest equivalent courses offered at another university. Results are evaluated and future work is discussed.

1 Introduction

Hundreds of students transfer to University of Massachusetts Lowell (UML) each year. As part of that process, courses taken at students' previous educational institutions must be evaluated by UML for transfer credit. Course descriptions are usually short paragraphs of less than 200 words. To determine whether an incoming course can be transferred, the undergraduate and graduate transfer coordinators from each department must manually compare its course description to the courses offered at UML. This process can be tedious and time-consuming. Although the publicly available *course transfer dictionary* (Figure 1) for students transferring to UML lists equivalent courses from hundreds of institutions, it is not always up to date and the data set is sparse and non-uniformed.

This work proposes an approach to automatically identify course equivalencies by analyzing the course descriptions and comparing their semantic distance. The course descriptions are first pruned and unrelated contexts are removed. Given a course from another university, the algorithm measures word, sentence, and paragraph similarities to suggest a list of potentially equivalent courses offered by UML. This work has two goals: (1) to efficiently and accurately suggest equivalent courses to reduce the workload of transfer coordinators, and (2) to explore new applications using semantic distance to move toward the Semantic Web, i.e., to turn existing resources into knowledge structures.



Ext. Course Title	Ext. Course #	UML Course #	UML Course Title
Cultural Anthropology	ANT 101	48.102	Social Anthropology
Art Appreciation	ART 101	58.101	Art Appreciation
Art History I	ART 105	58.203	History Of Art:Preh-Med
Art History II	ART 106	58.204	Hist Of Art II:Ren - Mod
Asian Art	ART 108	58.205	Studies In World Art
Color And Design	ART 113	70.101	Art Concepts I (studio)
Intro To Sculpture&3-D Design	ART 115	70.299	Studio Art 200 electives
Printmaking	ART 117	70.267	Printmaking
Drawing I	ART 121	70.255	Drawing I

Figure 1. A subset of the transfer dictionary for students transferred from an external institution to UML.

Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pages 142–151,
Portland, Oregon, 24 June 2011.

2 Related Research

Semantic distance measures have been used in applications such as automatic annotation, keyword extraction, and social network extraction (Matsuo et al., 2007). It is important to note that there are two kinds of semantic distance: *semantic similarity* and *semantic relatedness*. Semantic relatedness is more generic than semantic similarity in that it includes all classical and non-classical semantic relations such as holonymy[1], meronymy[2], and antonymy[3], where semantic similarity is limited to relations such as hyponymy[4] and hypernymy[5] (Budanitsky and Hirst, 2006). The terms semantic distance, semantic relatedness, and semantic similarity are sometimes used interchangeably by different authors in the literature related to this topic. The relative generality of the three terms is illustrated in Figure 2.

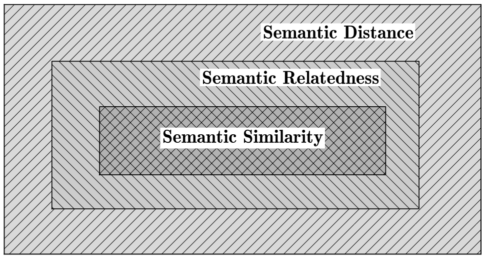


Figure 2. The relations of semantic distance, semantic relatedness, and semantic similarity as described by Budanitsky and Hirst (2006).

Related work in semantic distance measurement can be roughly divided into three categories: (1) lexicographic resource based methods, (2) corpus based methods, and (3) hybrid methods.

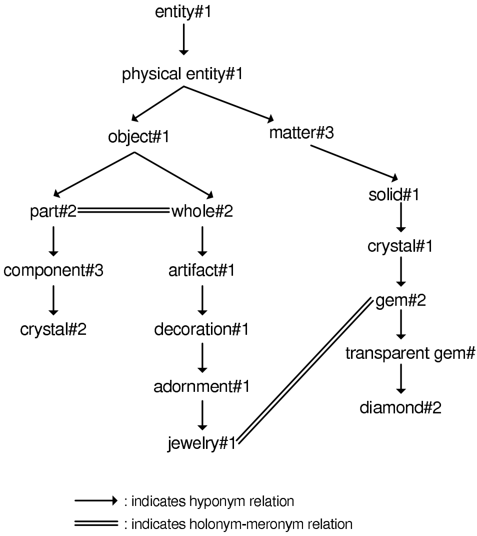


Figure 3. A fragment of WordNet's taxonomy.

Lexicographic resource based methods typically calculate semantic distance based on WordNet[6]. In related work (Rada et al., 1989; Wu and Palmer, 1994; Leacock and Chodorow, 1998; Hirst and St-Onge, 1998; Yang and Powers, 2005), lexicographic resource based methods use one or more edge-counting (also known as shortest-path) techniques in the WordNet taxonomy (Figure 3). In this technique, concept nodes are constructed in a hierarchical network and the minimum number of hops between any two nodes represents their semantic distance (Collins and Quillian, 1969). The measure by Hirst and St-Onge (1998) is based on the fact that the target concepts are likely more distant if the target path consists of edges that belong to many different relations. The approach by Leacock and Chodorow (1998) combines the shortest path with maximum depth so that edges lower down in the is-a hierarchy correspond to smaller semantic distances than the ones higher up. Yang and Powers (2005) further suggest that it is necessary to consider relations such as holonymy and meronymy.

A corpus-based method typically calculates co-occurrence on one or more corpora to deduce semantic closeness (Sahami and Heilman, 2006; Cilibrasi and Vitanyi, 2007; Islam and Inkpen, 2006; Mihalcea et al., 2006). Using this technique, two words

[1] A *holonym* is a word that names the whole of which a given word is a part. For example, “hat” is a holonymy for “brim” and “crown.”

[2] A *meronym* is a word that names a part of a larger whole. For example, “brim” and “crown” are meronyms of “hat.”

[3] A *antonym* is a word that expresses a meaning opposed to the meaning of another word. For example, “big” is an antonym of “small.”

[4] A *hyponym* is a word that is more specific than a given word. For example, “nickel” is a hyponym of “coin.”

[5] A *hypernym* is a word that is more generic than a given word. For example, “coin” is a hypernym of “nickel.”

[6] http://wordnet.princeton.edu/

are likely to have a short semantic distance if they co-occur within similar contexts (Lin, 1998).

Hybrid methods (including distributional measures) combine lexicographic resources with corpus statistics (Jiang and Conrath, 1997; Mohammad and Hirst, 2006; Li et al., 2003; Li et al., 2006). Related work shows that hybrid methods generally outperform lexicographic resource based and corpus based methods (Budanitsky and Hirst, 2006; Curran, 2004; Mohammad and Hirst, 2006; Mohammad, 2008).

Li et al. (2006) proposed a hybrid method based on WordNet and the Brown corpus to incorporate semantic similarity between words, semantic similarity between sentences, and word order similarity to measure overall sentence similarity. The semantic similarity between words is derived from WordNet based on path lengths and depths of lowest common hypernyms. The semantic similarity between two sentences is defined as the cosine coefficient of two vectors that are derived from building two semantic vectors and collecting the information content for each term from the Brown corpus. The word order similarity is then determined by the normalized difference in word order of each sentence. Finally, the overall sentence similarity is defined as the weighted sum of the semantic similarity between sentences and the word order similarity.

3 Proposed Method

This work proposes a variant of the hybrid method by Li et al. (2006) to identify course equivalencies by measuring the semantic distance between course descriptions. Our approach has three modules: (1) semantic distance between words, (2) semantic distance between sentences, and (3) semantic distance between paragraphs. Their word order similarity and overall sentence similarity modules are found to *decrease* the accuracy (See Section 4). Therefore, these methods are not used in our approach. This work modifies the semantic similarity between words and the semantic similarity between sentences modules developed by Li et al. (2006) and adds semantic distance between paragraphs tailored to the domain of identifying equivalent courses. Experiments show that these modifications maximized accuracy.

3.1 Semantic Distance Between Words

Given a concept c_1 of word w_1, and a concept c_2 of word w_2, the semantic distance between the two words (SDBW) is a function of the path length between the two concepts and the depth of their lowest common hypernym.

The path length p from c_1 to c_2 is determined by one of five cases. This work adds holonymy and meronymy relations to the method by Li et al. (2006) to measure the semantic relatedness:

1. c_1 and c_2 are in the same synonym set (synset).
2. c_1 and c_2 are not in the same synset, but the synset of c_1 and the synset of c_2 contain one or more common words.
3. c_1 is either a holonym or a meronym of c_2.
4. c_1 is neither a holonym nor a meronym of c_2, but the synset of c_1 contains one or more words that are either holonyms or meronyms of one or more words in the synset that c_2 belongs to.
5. c_1 and c_2 do not satisfy any of the previous four cases.

If c_1 and c_2 belong to case 1, p is 0. If c_1 and c_2 belong to cases 2, 3, or 4, p is 1. In case 5, p is the number of links between the two words. Therefore, the semantic distance of c_1 and c_2 is an exponential decaying function of p, where α is a constant (Li et al., 2006):

$$f_1(p) = e^{\alpha p} \quad (\alpha \in [-1, 0]). \tag{1}$$

Let h be the depth of the lowest common hypernym of c_1 and c_2 in the WordNet hierarchy. f_2 is a monotonically increasing function of h (Li et al., 2006):

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \quad (\beta \in [0, 1]). \tag{2}$$

The values of α and β are given in Section 4.

The semantic distance between concepts c_1 and c_2 is defined as:

$$f_{word}(c_1, c_2) = f_1(p) \cdot f_2(h), \tag{3}$$

where f_1 and f_2 are given by Equations 1 and 2. The values of both f_1 and f_2 are between 0 and 1 (Li et al., 2006).

WordNet is based on concepts, not words. Words with different meanings are considered different "words" and are marked with sense tags (Budanitsky and Hirst, 2006). Unfortunately, common corpora (as well as course descriptions) are not sense-tagged. Therefore, a mapping between a word and a certain sense must be provided. Such mapping is called word sense disambiguation (WSD), which is the ability to identify the meaning of words in context in a computational manner (Navigli, 2009). We consider two strategies to perform the WSD: (1) compare all senses of two words and select the maximum score, and (2) apply the first sense heuristic (McCarthy et al., 2004). We will show that the overall performance of the two strategies is about the same.

To improve accuracy, the *parts of speech*[7] (POS) of two words have to be the same before visiting the WordNet taxonomy to determine their semantic distance. Therefore, "book" as in "read a book" and "book" as in "book a ticket" are considered different. We do not distinguish the plural forms of POS from singular forms. Therefore, POS such as "NN" (the singular form of a noun) and "NNS"(the plural form of a noun) are considered the same.

The SDBW module also considers the *stemmed* forms of words. Without considering stemmed words, two equivalent course titles such as "networking" and "data communication" are misclassified as semantically distant because "networking" in WordNet is solely defined as socializing with people, not as a computer network. The stemmed word "network" is semantically closer to "data communication."

Algorithm 1 shows how to determine the semantic distance between two words w_1 and w_2.

The SDBW module uses WordNet as a lexical knowledge base to determine the semantic closeness between words. The path lengths and depths in the WordNet IS-A hierarchy may be used to measure how strongly a word contributes to the meaning of a sentence. However, this approach has a problem. Because WordNet is a manually created lexical resource, it does not cover all the words that appear in a sentence, even though some of these words are commonly seen in literature. Words not defined in WordNet are misclassified as semantically distant when compared with any other words. This is a huge problem for identifying equivalent courses. For example, course names "propositional logic" and "logic" are differentiated solely by the word "propositional," which is not defined in WordNet[8]. The semantic distance measurement between *sentences* therefore cannot be simplified to all pairwise comparisons of words using WordNet. A corpus must be introduced to assess the semantic relatedness of words in sentences.

Algorithm 1 Semantic Distance Between Words

```
1: If two words $w_1$ and $w_2$ have different POS, consider them semantically distant. Return 0.
2: If $w_1$ and $w_2$ have the same POS and look the same but do not exist in WordNet, consider them semantically close. Return 1.
3: Using either maximum scores or the first sense heuristic to perform WSD, measure the semantic distance between $w_1$ and $w_2$ using Equation 3.
4: Using the same WSD strategy as the previous step, measure the semantic distance between the stemmed $w_1$ and the stemmed $w_2$ using Equation 3.
5: Return the larger of the two results in steps (3) and (4), i.e., the score of the pair that is semantically closer.
```

3.2 Semantic Distance Between Sentences

To measure the semantic distance between sentences, Li et al. (2006) join two sentences S_1 and S_2 into a unique word set S, with a length of n:

$$S = S_1 \cup S_2 = \{w_1, w_2, \dots w_n\}. \tag{4}$$

A semantic vector SV_1 is computed for sentence S_1 and another semantic vector SV_2 for sentence S_2. Given the number of words in S_1 as t, Li et al. (2006) define the value of an entry of SV_1 for sentence S_1 as:

$$SV_{1i} = \hat{s_{1i}} \cdot I(w_i) \cdot I(w_{1j}), \tag{5}$$

where $i \in [1, n]$, $j \in [1, t]$, $\hat{s_{1i}}$ is an entry of the lexical semantic vector $\hat{s_1}$ derived from S_1, w_i is a word in S, and w_{1j} is semantically the closest to w_i

[7] We use the part-of-speech tags from the Penn Treebank project: http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

[8] WordNet 3.0 is used in our implementation and experiments.

in S_1. $I(w_i)$ is the information content (IC) of w_i in the Brown corpus and $I(w_{1j})$ is the IC of w_{1j} in the same corpus.

Our work redefines the semantic vector as:

$$SV_{1i} = \hat{s_{1i}} \cdot (TFIDF(w_i)+\epsilon) \cdot (TFIDF(w_{1j})+\epsilon). \tag{6}$$

There are two major modifications in our version. First, we replace the information content with the Term Frequency–Inverse Document Frequency (TFIDF) weighting scheme, which is a bag-of-words model (Joachims, 1997). In the TFIDF formula, each term i in document D is assigned weight m_i:

$$m_i = tf_i \cdot idf_i = tf_i \cdot \log \frac{N}{df_i}, \tag{7}$$

where tf_i is the frequency of term i in D, idf_i is the inverse document frequency of term i, N is the total number of documents, and df_i is the number of documents that contain i (Salton and Buckley, 1987). Our approach uses a smoothing factor ϵ to add a small mass[9] to the TFIDF.

Second, we compute TFIDF over our custom course description corpus instead of the Brown corpus. The course description corpus is built from crawling the course catalogs from two universities' websites. These two modifications find inner relations of words from the course description data domain, rather than from the various domains provided by the Brown corpus.

The semantic distance of S_1 and S_2 is the cosine coefficient of their semantic vectors SV_1 and SV_2 (Li et al., 2006):

$$f_{sent}(S_1, S_2) = \frac{SV_1 \cdot SV_2}{||SV_1|| \cdot ||SV_2||}. \tag{8}$$

Although Li et al. (2006) do not remove *stop words*[10], it is found that the removal of stop words remarkably improves accuracy to identify equivalent courses. (See Section 4.)

While building and deriving the lexical semantic vectors $\hat{s_1}$ for sentence S_1 and $\hat{s_2}$ for sentence S_2, it is found that some words from the joint word list S (Equation 4) which are not stop words, but are very generic, in turn rank as semantically the closest words to most other words. These generic words cannot be simply regarded as domain-specific stop words in that a generic word in a pair of courses may not be generic in another pair. To discourage these generic words, we introduce a *ticketing algorithm* as part of the process to build a lexical semantic vector. Algorithm 2 shows the steps to build the lexical semantic vector[11] $\hat{s_1}$ for sentence S_1. Similarly, we follow these steps to build $\hat{s_2}$ for S_2.

Algorithm 2 Lexical Semantic Vector $\hat{s_1}$ for S_1

```
1: for all words $w_i \in S$ do
2:     if $w_i \in S_1$, set $\hat{s_{1i}} = 1$ where $\hat{s_{1i}} \in \hat{s_1}$.
3:     if $w_i \notin S_1$, the semantic distance between $w_i$ and each word $w_{1j} \in S_1$ is calculated (Section 3.1). Set $\hat{s_{1i}}$ to the highest score if the score exceeds a preset threshold $\delta$ ($\delta \in [0, 1]$), otherwise $\hat{s_{1i}} = 0$.
4:     Let $\gamma \in [1, n]$ be the maximum number of times a word $w_{1j} \in S_1$ is chosen as semantically the closest word of $w_i$. Let the semantic distance of $w_i$ and $w_{1j}$ be $d$, and $f_{1j}$ be the number of times that $w_{1j}$ is chosen. If $f_{1j} > \gamma$, set $\hat{s_{1i}} = d/\gamma$ to give a penalty to $w_{1j}$. This step is called ticketing.
5: end for
```

3.3 Semantic Distance Between Paragraphs

Although Li et al. (2006) claim that their approach is for measuring the semantic similarity of sentences and short texts, test cases show that the accuracy of their approach is not satisfactory on course descriptions. We introduce the semantic distance measure between paragraphs to address this problem.

Given course descriptions P_1 and P_2, the first step is to remove generic data and prerequisite information. Let P_1 be a paragraph consisting of a set of n sentences, and P_2 be a paragraph of m sentences, where n and m are positive integers. For s_{1i} ($s_{1i} \in P_1$, $i \in [1, n]$) and s_{2j} ($s_{2j} \in P_2$, $j \in [1, m]$), the semantic distance between paragraphs P_1 and P_2 is defined as a weighted mean:

[9] In our experiments, ϵ=0.01.

[10] Stop words (such as "the", "a", and "of") are words that appear in almost every document, and have no discrimination value for contexts of documents. Porter et al.'s English stop words list (http://snowball.tartarus.org/algorithms/english/stop.txt) are adapted in this work.

[11] In our experiments, we chose δ=0.2.

$$f_{para}(P_1, P_2) = \frac{\sum_{i=1}^{n} (\max_{j=1}^{m} f_{sent}(s_{1i}, s_{2j})) \cdot N_i}{\sum_{i=1}^{n} N_i}, \tag{9}$$

where N_i is the sum of the number of words in sentences s_{1i} ($s_{1i} \in P_1$) and s_{2j} ($s_{2j} \in P_2$), and $f_{sent}(s_{1i}, s_{2j})$ is the semantic distance between sentences s_{1i} and s_{2j} (Section 3.2). Algorithm 3 summarizes these steps. Optionally the *deletion* flag can be enabled to speed up the computation. Empirical results show that accuracy is about the same whether or not the *deletion* flag is enabled.

Algorithm 3 Semantic Distance for Paragraphs

```
If deletion is enabled, given two course descriptions, select the one with fewer sentences as $P_1$, and the other as $P_2$. If deletion is disabled, select the first course description as $P_1$, and the other as $P_2$.
for each sentence $s_{1i} \in P_1$ do
    Calculate the semantic distance between sentences (Section 3.2) for $s_{1i}$ and each of the sentences in $P_2$.
    Find the sentence pair $\langle s_{1i}, s_{2j} \rangle$ ($s_{2j} \in P_2$) that scores the highest. Save the highest score and the total number of words of $s_{1i}$ and $s_{2j}$. If deletion is enabled, remove sentence $s_{2j}$ from $P_2$.
end for
Collect the highest score and the number of words from each run. Use their weighted mean (Equation 9) as the semantic distance between $P_1$ and $P_2$.
```

We introduce θ to denote how much we weigh course titles over course descriptions. Course titles are compared using the semantic distance measurement discussed in Section 3.2. Given title T_1 and description P_1 of course C_1, and title T_2 and description P_2 of course C_2, the semantic distance of the two courses is defined as:

$$\begin{aligned} f_{course}(C_1, C_2) = \theta \cdot f_{sent}(T_1, T_2) \\ + (1 - \theta) \cdot f_{para}(P_1, P_2). \end{aligned} \tag{10}$$

4 Implementation and Experimental Results

The method proposed in this paper is fully implemented using Python and NLTK (Bird et al., 2009). The WordNet interface built into NLTK is used to retrieve lexical information for word similarities. In our experiments, the default parameters are: $\alpha = -0.2$, $\beta = 0.45$ (Li et al., 2006), $\gamma = 2$, and $\theta = 0.7$. The γ and θ values are found empirically to perform well.

A course description corpus must be built for the experiments. The UMass Lowell (UML) course transfer dictionary lists courses that are equivalent to those from hundreds of other institutions (see Figure 1, shown in Section 1). We only used the transfer dictionary as a test corpus rather than a training corpus to keep the algorithm simple and efficient. Middlesex Community College (MCC) is picked as an external institution in our experiments. The transfer dictionary lists over 1,400 MCC courses in different majors. We remove the rejected courses, elective courses, and those with missing fields from the transfer dictionary. Referring to the equivalencies from the transfer dictionary, we crawl over 1,500 web pages from the course catalogs of both UML and MCC to retrieve over 200 interconnected courses that contain both course names and descriptions. Two XML files are created, one for UML and one for MCC courses. Given an MCC course, the goal is to suggest the most similar UML course. A fragment of the MCC XML file is shown below. Each course entry has features such as course ID, course name, credits, description, and the ID of its equivalent course at UML. The UML XML file has the same layout except that the *equivalence* tag is removed and the root tag is *uml*.

```
<mcc>
  <course>
    <courseid>ART 113</courseid>
    <coursename>Color and Design</coursename>
    <credits>3</credits>
    <description>Basic concepts of composition
    and color theory. Stresses the process and
    conceptual development of ideas in two
    dimensions and the development of a strong
    sensitivity to color.</description>
    <equivalence>70.101</equivalence>
  </course>
  ...
</mcc>
```

After the integrity check, the MCC XML file contains 108 courses and the UML XML file contains 89 courses. The reason there are more MCC courses than UML courses is that the transfer dictionary allows multiple courses from MCC to be transferred to the same UML course.

To monitor the accuracy change over different numbers of documents, we randomly select equivalent courses to create two smaller data sets for UML and MCC respectively in the XML format. The random number of courses in each XML file is shown in Table 1. These three pairs of XML data sets are used both as the corpora and as the test data sets.

XML Datasets	MCC Courses	UML Courses	Total
Small	25	24	49
Medium	55	50	105
Large	108	89	197

Table 1. Number of courses in the data sets

Consider the small data set as an illustration. Each of the 25 MCC courses is compared with all 24 UML courses. All words are converted to lowercase and punctuation is removed. We also remove both *general stop words*[12] (such as "a" and "of") and *domain-specific stop words*[13] (such as "courses," "students," and "reading"). We do not remove words based on high or low occurrence because that is found empirically to *decrease* accuracy. Using the algorithms discussed in Section 3, a score is computed for each comparison. After comparing an MCC course to all UML courses, the 24 UML courses are sorted by score in descending order. The course equivalencies indicated by the transfer dictionary are used as the benchmark. In each run we mark the rank of the real UML course that is equivalent to the given MCC course as indicated by the transfer dictionary. We consider the result of each run correct when the equivalent course indicated by the transfer dictionary is in the top 3 of the sorted list. After doing this for all the 25 MCC courses, we calculate the overall accuracy and the average ranks of the real equivalent courses.

Empirical results show that accuracy drops when some inseparable phrases naming *atomic keywords* (such as "local area networks," "data communications," and "I/O") are tokenized. To address this problem, a list of 40 atomic keywords is constructed manually.

Our approach is compared against two baselines: TFIDF only (Equation 7), and the method by Li et al. (2006). Since the method by Li et al. (2006) does not measure semantic distance between paragraphs, we consider each course description as a sentence. Figure 4 shows that the accuracy of our approach outperforms the TFIDF and Li et al. (2006) approaches over the three sets of documents from Table 1. It is interesting to note that while the accuracies of the TFIDF and Li et al. (2006) approaches *decrease* as the number of documents increases, the accuracy of our approach *increases* when the number of documents increases from 105 to 195. This observation is counter-intuitive and therefore requires further analysis in future work.

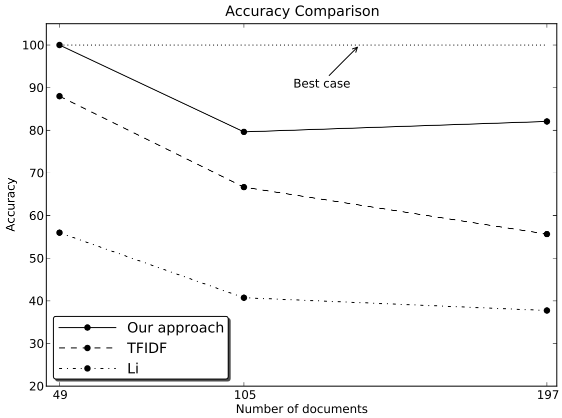


Figure 4. Accuracy of our approach compared to the TFIDF and Li et al. (2006) approaches.

For each of the three different approaches, we note the average ranks of the real equivalent courses indicated by the transfer dictionary. Figure 5 shows that our approach outperforms the TFIDF and Li et al. (2006) approaches. It also shows that the average rank in our approach does not increase as fast as the other two.

The word order similarity module in the Li et al. (2006) approach tokenizes two sentences into a list of unique words. Each of the two sentences is converted into a numbered list where each entry in the list is the index of the corresponding word in the joint set. The word order similarity between these

[12] A list of English stop words in NLTK is used in our experiments.

[13] A list of domain-specific stop words is created manually.

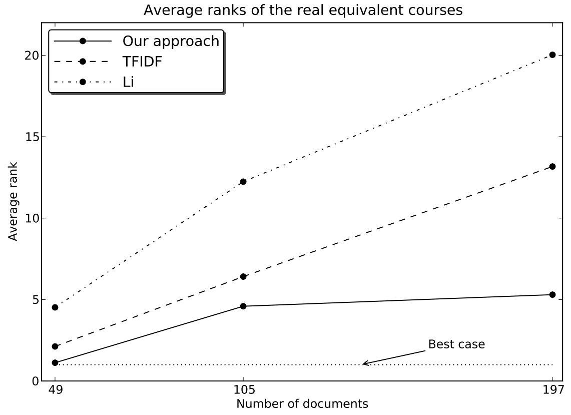


Figure 5. Average ranks of the real equivalent courses.

two sentences is in turn the normalized difference of their word orders. We experiment with enabling and disabling word order similarity to compare accuracy (Figure 6) and speed. Empirical results show that disabling word order similarity increases the accuracy of our approach and the speed is over 20% faster. Therefore, the word order similarity module by Li et al. (2006) is removed from our approach.

We then compare the two WSD strategies as described in Section 3.1: (1) always select the maximum score on all senses of two words (Max), and (2) apply the first sense heuristic. As Figure 7 and Figure 8 suggest, the accuracy of Max is higher than the first sense heuristic, but the average rank of the first sense heuristic is better than Max. Therefore, the overall performance of the two strategies is about the same.

We also experiment with enabling and disabling ticketing (Section 3.2). Results show that both accuracy and average ranks are improved when ticketing is enabled.

5 Future Refinements

This paper presents a novel application of semantic distance to suggesting potential equivalencies for a course transferred from an external university. It proposes a hybrid method that incorporates semantic distance measurement for words, sentences, and paragraphs. We show that a composite weighting scheme based on a lexicographic resource and a bag-of-words model outperforms previous work to iden-

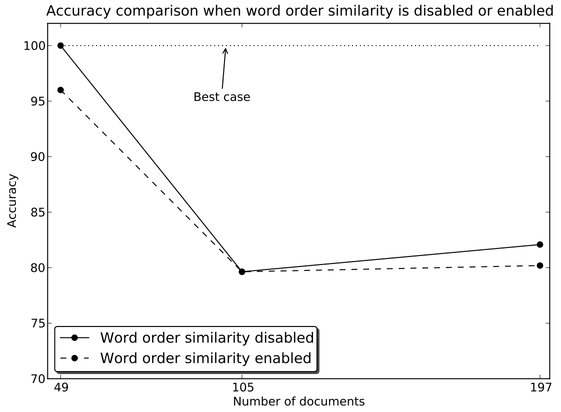


Figure 6. The accuracy of our approach when enabling or disabling word order similarity.

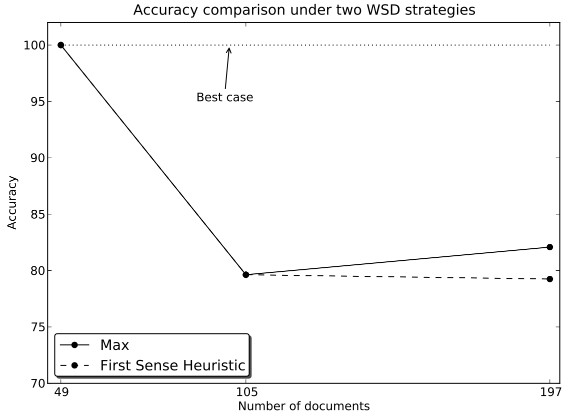


Figure 7. Accuracy comparison under two WSD strategies.

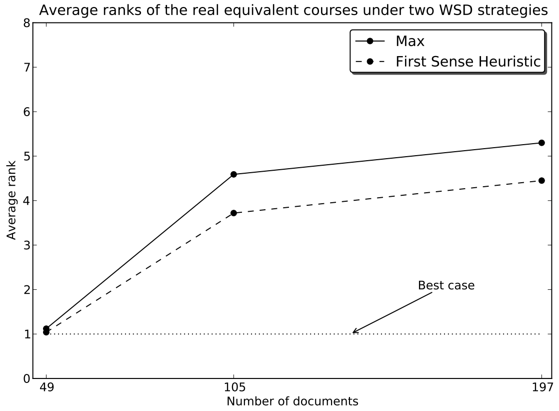


Figure 8. Average ranks of the real equivalent courses under two WSD strategies.

tify equivalent courses. In practice, it is not common for two sentences in the course description corpus to have the exact same word order. Therefore, word order similarity is not very useful for identifying course equivalencies. Empirical results suggest that WSD and POS are helpful to increase accuracy, and that it is necessary to remove general and domain-specific stop words. The ticketing algorithm (Algorithm 2) also improves accuracy.

UML's transfer dictionary is only used as a test corpus in this paper. Alternatively, a set of examples might be constructed from the transfer dictionary to automatically learn equivalent properties without compromising the time complexity. Analyzing transfer dictionaries from other universities might help as well.

Meta data such as course levels, textbooks, and prerequisites can also be used as indicators of course equivalencies, but unfortunately these data are not available in the resources we used. Obtaining these data would require a great deal of manual work, which runs counter to our goal of devising a simple and straightforward algorithm for suggesting course equivalencies with a reasonable time complexity.

WordNet is selected as the lexical knowledge base for determining the semantic closeness between words, but empirical results indicate that WordNet does not cover all the concepts that exist in course descriptions. To address this issue, a domain-specific ontology could be constructed.

We plan to test our approach against other semantic distance measures in addition to the approach by Li et al. (2006), such as the work by Mihalcea et al. (2006) and Islam and Inkpen (2007).

Other directions for future work include: (1) optimizing performance and the exploration of more elegant WSD algorithms, (2) testing the sensitivity of results to values of γ and θ, (3) testing courses from a larger number of universities, (4) proposing robust methodologies that tolerate poorly formed texts, (5) adding more data to the course description corpus, and (6) making the course description corpus publicly available to the research community.

Acknowledgments

The authors thank Dr. Karen M. Daniels for reviewing drafts of this paper. We also appreciate correspondence with Dr. Yuhua Li at the early stage of our work. Last, but not least, we thank the reviewers for their insightful comments that guided improvement of the contents of this paper.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural Language Processing with Python. *O'Reilly Media, Inc.* Sebastopol, CA, USA

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, volume 32.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*. 19(3):370 – 383

Allan M. Collins and M. Ross Quillian. 1969. Retrieval Time from Semantic Memory. *Journal of Verbal Learning and Verbal Behavior*, volume 8.

James R. Curran. 2004. From Distributional to Semantic Similarity. Ph.D. Thesis. University of Edinburgh, Edinburgh, U.K.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, WordNet: An Electronic Lexical Database. *The MIT Press, Cambridge, MA*, pages 305–332.

Aminul Islam and Diana Inkpen. 2006. Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words. *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, pages 1033–1038.

Aminul Islam and Diana Inkpen. 2007. Semantic Similarity of Short Texts. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Bulgaria, September 2007.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan, pages 19–33.

Thorsten Joachims. 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. *Proceedings of International Conference on Machine Learning (ICML).*

Claudia Leacock and Martin Chodorow. 1998. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, 24(1):147–165.

Yuhua Li, Zuhair A. Bandar, and David McLean. 2003. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources.

IEEE Transactions on Knowledge and Data Engineering, volume 15, pages 871–882. IEEE Computer Society.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering* volume 18. IEEE Computer Society. Los Alamitos, CA, USA.

Dekang Lin. 1998a. Extracting Collocations from Text Corpora. *Workshop on Computational Terminology*, Montreal, Canada.

Yutaka Matsuo, Junichiro Mori, Masahiro Hamasaki, Takuichi Nishimura, Hideaki Takeda, Koiti Hasida, and Mitsuru Ishizuka. 2007. POLYPHONET: An Advanced Social Network Extraction System from the Web. *Web Semantics*, volume 5(4). Elsevier Science Publishers B.V., Amsterdam, The Netherlands.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*, Boston.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Using Automatically Acquired Predominant Senses for Word Sense Disambiguation. *In Proceedings of the ACL SENSEVAL-3 Workshop*. Barclona, Spain. pp 151-154.

Saif Mohammad. 2008. Measuring Semantic Distance Using Distributional Profiles of Concepts. Ph.D. Thesis. University of Toronto, Toronto, Canada.

Saif Mohammad and Graeme Hirst. 2006. Determining Word Sense Dominance Using a Thesaurus. *In Proceedings of the 11th conference of the European chapter of the Association for Computational Linguistics (EACL-2006)*, April 2006, Trento, Italy.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):1–69.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, volume 19.

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, volume 1. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Mehran Sahami and Timothy D. Heilman. 2006. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. *In Proceedings of the 15th International Conference on World Wide Web (WWW '06)*. ACM. New York, NY, USA.

Gerard Salton and Chris Buckley. 1987. Term Weighting Approaches in Automatic Text Retrieval. *Technical report*. Ithaca, NY, USA.

Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical machine learning tools and techniques, 2nd Edition. *Morgan Kaufmann*, San Francisco, CA, USA, pages 161–171.

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics And Lexical Selection. *In Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94)*. Association for Computational Linguistics, Stroudsburg, PA, USA.

Dongqiang Yang and David M. W. Powers. 2005. Measuring semantic similarity in the taxonomy of WordNet. *In Proceedings of the 28th Australasian conference on Computer Science (ACSC '05)*, volume 38. Australian Computer Society, Darlinghurst, Australia.