Domain-Specific Semantic Relatedness From Wikipedia: Can A Course Be Transferred?

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

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

Abstract

Semantic relatedness, or its inverse, semantic distance, measures the degree of closeness between two pieces of text determined by their meaning. Related work typically measures semantics based on a sparse knowledge base such as WordNet[1] or CYC that requires intensive manual efforts to build and maintain. Other work is based on the Brown corpus, or more recently, Wikipedia. Wikipedia-based measures, however, typically do not take into account the rapid growth of that resource, which exponentially increases the time to prepare and query the knowledge base. Furthermore, the generalized knowledge domain may be difficult to adapt to a specific domain. To address these problems, this paper proposes a domain-specific semantic relatedness measure based on part of Wikipedia that analyzes course descriptions to suggest whether a course can be transferred from one institution to another. We show that our results perform well when compared to previous work.

1 Introduction

Many NLP techniques have been adapted to the education field for building systems such as automated scoring, intelligent tutoring, and learner cognition. Few, however, address the identification of transfer course equivalencies. A recent study by the National Association for College Admission Counseling[2] reveals that 1/3 of US college students transfer to another institution. Correspondingly, University of Massachusetts Lowell (UML) accepts hundreds of transfer students every year. Each transfer course must be evaluated for credits by *manually* comparing its course description to courses offered at UML. This process is labor-intensive and highly inefficient. There is a publicly available *course transfer dictionary* which lists course numbers from hundreds of institutions and their equivalent courses at UML, but the data set is sparse, non-uniform, and always out of date. External institutions cancel courses, change course numbers, etc., and such information is virtually impossible to keep up to date in the transfer dictionary. Furthermore, the transfer dictionary does not list course descriptions. From our experience, course descriptions change over the years even when course numbers do not, and this of course affect equivalencies.

This work proposes a domain-specific semantic relatedness measure using Wikipedia that automatically suggests whether two courses from different institutions are equivalent by analyzing their course descriptions. The goal is to assist transfer coordinators by suggesting equivalent courses within a reasonable amount of time on a standard laptop system. Our model is a mapping function: $f : (C_1, C_2) \rightarrow n, n \in [0, 1]$, where C_1 is a Computer Science (CS) course from an external institution, and C_2 is a CS course offered at UML. Output n is the semantic relatedness score, where a bigger value indicates C_1 and C_2 are more related. Each course description is a short text passage:

- C_1: **[Analysis of Algorithms]** Discusses basic methods for designing and analyzing efficient algorithms empha-

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

[2] Special Report on the Transfer Admission Process: http://www.nacacnet.org/research/research-data/Documents/TransferFactSheet.pdf

Proceedings of the NAACL HLT 2012 Student Research Workshop, pages 35–40,
Montréal, Canada, June 3-8, 2012.

sizing methods used in practice. Topics include sorting, searching, dynamic programming, greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning trees, tree traversals), matrix operations, string matching, NP completeness.

- C_2: **[Computing III]** Object-oriented programming. Classes, methods, polymorphism, inheritance. Object-oriented design. C++. UNIX. Ethical and social issues.

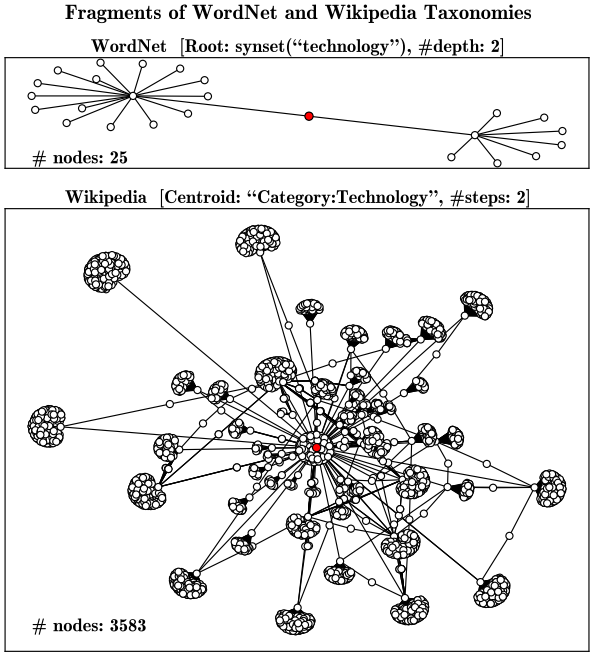

Figure 1. Fragments of WordNet 3.0 (top) and English Wikipedia of 2011/7 (bottom) taxonomies. The root/centroid node is shown in red.

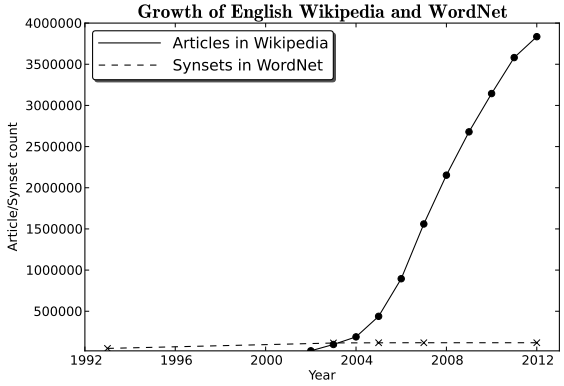

Figure 2. Growth of Wikipedia and WordNet

We choose Wikipedia as the knowledge base due to its rich contents (Figure 1) and continuously coalescent growth (Bounova, 2011). Although Wikipedia was launched 10 years later, it grew much faster than WordNet over the last decade (Figure 2).

The contributions of this paper are twofold. First, we address the problem of domain-specific semantic relatedness using Wikipedia. We propose a method to suggest course equivalencies by computing semantic relatedness among Computer Science course descriptions. Our approach can be easily modified for other majors and even other languages. Second, we evaluate the correlation of our approach and a human judgment data set we built. Both accuracy and correlation indicate that our approach outperforms previous work.

2 Related Research

Semantic relatedness has been used in applications such as word sense disambiguation, named entity disambiguation, text summarization and annotation, lexical selection, automatic spelling correction, and text structure evaluation. WordNet is commonly used as a lexicographic resource to calculate semantic relatedness (Budanitsky and Hirst, 2006). A WordNet-based method uses one or more edge-counting techniques in the WordNet taxonomy (Leacock and Chodorow, 1998; Hirst and St-Onge, 1998). The relatedness of two concept nodes is a function of the minimum number of hops between them.

Some related work calculates co-occurrence on one or more large corpora to deduce semantic relatedness (Sahami and Heilman, 2006; Cilibrasi and Vitanyi, 2007). Two words are likely to be related if they co-occur within similar contexts (Lin, 1998). Others combine lexicographic resources with corpus statistics (Jiang and Conrath, 1997). It has been shown that these composite methods generally outperform lexicographic resource- and corpus- based methods (Budanitsky and Hirst, 2006; Curran, 2004; Mohammad, 2008). Li et al. (2006) propose 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 the overall sentence similarity. Yang and Heines (2011) modify this work to suggest transfer course equivalencies, but the experiment is based on non-technical courses. Due to the WordNet sparsity on technical terms, the experiment does not perform well on Computer Science courses.

In recent years, there has been increasing interest in applying Wikipedia and related resources to question answering (Buscaldi and Rosso, 2006), word sense disambiguation (WSD) (Mihalcea and Csomai, 2007), name entity disambiguation (Ni et al., 2010), ontology evaluation (Yu et al., 2007), semantic web (Wu, 2010), and computing semantic relatedness (Ponzetto and Strube, 2007). Ponzetto and Strube (2007) deduce semantic relatedness of words by modeling relations on the Wikipedia category graph. Gabrilovich and Markovitch (2009) introduce the Explicit Semantic Analysis (ESA) model which calculates TF-IDF (Manning et al., 2008) values for every word in Wikipedia and further uses local linkage information to build a second-level semantic interpreter.

Our approach is different from prior work on Wikipedia. While Mihalcea and Csomai (2007) use the annotation in the page title of a concept to perform WSD, our approach uses a page's parent category as a cue to the correct sense. Ponzetto and Strube (2007) limit their measurement to word pairs, while our work focuses on text of any length. Gabrilovich and Markovitch (2009) computes TF-IDF statistics for every word and every document of Wikipedia which is highly inefficient. They also remove category pages and disambiguation pages. In contrast, our model is mainly based on the category taxonomy and the corpus statistics are limited to metadata that are mostly available in Wikipedia. Furthermore, we compute concept relatedness on a domain-specific hierarchy that weighs both path lengths and diversions from the topic. The domain-specific hierarchy is much smaller than the entire Wikipedia corpus. As a result, our algorithm is more efficient[3] than previous work.

[3]In our experiment, the average time needed to compare one pair of course descriptions ranged from 0.16 second (with partial caching) to 1 minute (without caching) on a 2.6Ghz Quad-Core PC. The most time-consuming part before comparing courses was to index all the Wikipedia tables in a MySQL database, which took overnight (same for ESA). It only took us 15 minutes to go through 19K pages to build a hierarchy of $D = 4$. In contrast, ESA's first level semantic interpreter (which tokenizes every Wikipedia page to compute TF-IDF) took 7 days to build over the same 19K pages. Both implementations were single-threaded, coded in Python, and tested over the English Wikipedia of July 2011.

3 Proposed Method

Our method contains four modules. Section 3.1 explains how to construct a domain-specific hierarchy from Wikipedia. Section 3.2 presents semantic relatedness between concepts. Section 3.3 describes the steps to generate features from course descriptions. And section 3.4 evaluates course relatedness.

3.1 Extract a Lexicographical Hierarchy

When a domain is specified (e.g., CS courses), we start from a generic Wikipedia category in this domain, choose its parent as the root, and use a depth-limited search to recursively traverse each subcategory (including subpages) to build a lexicographical hierarchy with depth D. For example, to find CS course equivalencies, we built a hierarchy using the parent of "Category:Computer science," i.e., "Category:Applied sciences," as the root. The generic category's parent is chosen as the root to make sure the hierarchy not only covers the terms in this domain, but also those in neighbor domains. The hierarchy of "Category:Applied sciences" not only covers Computer Science, but also related fields such as Computational Linguistics and Mathematics.

Both the number of nodes and number of edges in the hierarchy grow exponentially[4] as the depth increases. Therefore, D need not be a big number to cover most terms in the domain. We have found the hierarchy speeds up the semantic measurement dramatically and covers almost all the words in the specific domain. In our experiment on CS courses (D=6), we eliminated over 71% of Wikipedia articles,[5] yet the hierarchy covered over 90% of CS terminologies mentioned in the course descriptions.

3.2 Semantic Relatedness Between Concepts

Similar to the work of Li et al. (2006), the semantic relatedness between two Wikipedia concepts,[6] t_1 and t_2 in the hierarchy is defined as:

$$f'(t_1, t_2) = e^{-\alpha p} \cdot \frac{e^{\beta d} - e^{-\beta d}}{e^{\beta d} + e^{-\beta d}} \quad (\alpha, \beta \in [0, 1]), \quad (1)$$

where p is the shortest path between t_1 and t_2, and d is the depth of the lowest common hypernym of t_1

[4]In the hierarchy we built with "Category:Applied sciences" as the root, the number of edges grows from 177,955 at D=4 to 494,039 at D=5 and 1,848,052 at D=6.

[5]The hierarchy contains 1,534,267 unique articles, as opposed to 5,329,186 articles in Wikipedia.

[6]Each concept corresponds to a Wikipedia page.

and t_2 in the hierarchy (Section 3.1). This is different from related work on semantic relatedness from Wikipedia (Ponzetto and Strube, 2007) in that we not only consider the shortest path (p) between two concepts but also their common distance (d) from the topic, which in turn emphasizes domain awareness.

3.3 Generate Course Description Features

The built-in redirection in Wikipedia is useful for spelling corrections because variations of a term redirect to the same page. To generate features from a course description C, we start by generating n-grams ($n \in [1, 3]$) from C. We then query the *redirection data* to fetch all pages that match any of the n-grams.

The identified pages are still sparse. We therefore query the *title data* to fetch those that match any of the n-grams. Page topics are not discriminated in this step. For example, unigram "Java" returns both "Java (software platform)" and "Java (dance)."

Wikipedia contains a collection of disambiguation pages. Each disambiguation page includes a list of alternative uses of a term. Note that there are two different Wikipedia disambiguation pages: *explicit* and *implicit*. A page is *explicit* when the page title is annotated by Wikipedia as "disambiguation," such as "Oil (disambiguation)." A page is *implicit* when it is *not* so annotated, but points to a category such as "Category:Disambiguation pages," or "Category:All disambiguation pages." We iterate over the pages fetched from the last step, using disambiguation pages to enrich and refine the features of a course description.

Unlike the work of Mihalcea and Csomai (2007) which uses the annotation in the page title of a concept to perform WSD, our approach uses a page's parent category as a cue to the correct sense. Typically, the sense of a concept depends on the senses of other concepts in the context. For example, a paragraph on programming languages and data types ensures that "data" more likely corresponds to a page under "Category:Computer data" than one under "Category:Star Trek."

Algorithm 1 explains the steps to generate features for a course C.

Given the C_1 and C_2 in section 1, their generated features F_1 and F_2 are:

F_1: Shortest path problem, Tree traversal, Spanning tree, Tree, Analysis, List of algorithms, Completeness, Algorithm, Sorting, Data structure, Structure, Design, Data.

F_2: Unix, Social, Ethics, Object-oriented design, Computer programming, C++, Object-oriented programming, Design.

Algorithm 1 Feature Generation (F) for Course C

1. $T_c \leftarrow \emptyset$ (clear terms), $T_a \leftarrow \emptyset$ (ambiguous terms).
2. Generate all possible n-grams ($n \in [1, 3]$) G from C.
3. Fetch the pages whose titles match any of $g \in G$ from Wikipedia *redirection data*. For each page pid of term t, $T_c \leftarrow T_c \cup \{t : pid\}$.
4. Fetch the pages whose titles match any of $g \in G$ from Wikipedia *page title data*. If a disambiguation page, include all the terms this page refers to. If a page pid corresponds to a term t that is not ambiguous, $T_c \leftarrow T_c \cup \{t : pid\}$, else $T_a \leftarrow T_a \cup \{t : pid\}$.
5. For each term $t_a \in T_a$, find the disambiguation that is on average most related (Equation 1) to the set of clear terms. If a page pid of t_a is on average the most related to the terms in T_c, and the relatedness score is above a threshold δ ($\delta \in [0, 1]$), set $T_c \leftarrow T_c \cup \{t_a : pid\}$. If t_a and a clear term are different senses of the same term, keep the one that is more related to all the other clear terms.
6. Return clear terms as features.

Algorithm 2 Semantic Vector SV_1 for F_1 and J

1. **for all** words $t_i \in J$ **do**
2. if $t_i \in F_1$, set $SV_{1i} = 1$ where $SV_{1i} \in SV_1$.
3. if $t_i \notin F_1$, the semantic relatedness between t_i and each term $t_{1j} \in F_1$ is calculated (Equation 1). Set SV_{1i} to the highest score if the score exceeds the preset threshold δ, otherwise $SV_{1i} = 0$.
4. **end for**

3.4 Determine Course Relatedness

Given two course descriptions C_1 and C_2, we use Algorithm 1 to generate features F_1 for C_1, and F_2 for C_2. Next, the two feature lists are joined together into a unique set of terms, namely J. Similar to previous work (Li et al., 2006), semantic vectors SV_1 (Algorithm 2) and SV_2 are computed for F_1 and F_2.

Each value of an entry of SV_1 for features F_1 is reweighed as:

$$SV_{1i} = SV_{1i} \cdot I(t_i) \cdot I(t_j), \quad (2)$$

where SV_{1i} is the semantic relatedness between $t_i \in F_1$ and $t_j \in J$. $I(t_i)$ is the information content of t_i, and $I(t_j)$ is the information content of t_j. Similarly, we reweigh each value for the semantic vector SV_2 of F_2.

The information content $I(t)$ of a term t is a weighed sum of the category information content $I_c(t)$ and the linkage information content $I_l(t)$:

$$I(t) = \gamma \cdot I_c(t) + (1 - \gamma) \cdot I_l(t). \quad (3)$$

Inspired by related work (Seco et al., 2004), the category information content of term t is redefined as a function of its siblings:

$$I_c(t) = 1 - \frac{log(siblings(t) + 1)}{log(N)}, \quad (4)$$

where $siblings(t)$ is the number of siblings for t on average, and N is the total number of terms in the hierarchy (Section 3.1).

The linkage information content is a function of outlinks and inlinks of the page pid that t corresponds to:

$$I_l(t) = 1 - \frac{inlinks(pid)}{MAXIN} \cdot \frac{outlinks(pid)}{MAXOUT}, \quad (5)$$

where $inlinks(pid)$ and $outlinks(pid)$ are the numbers of inlinks and outlinks of a page pid. $MAXIN$ and $MAXOUT$ are the maximum numbers of inlinks and outlinks that a page has in Wikipedia.[7]

Finally, the relatedness of two courses is a cosine coefficient of the two semantic vectors:

$$f(C_1, C_2) = \frac{SV_1 \cdot SV_2}{||SV_1|| \cdot ||SV_2||}. \quad (6)$$

4 Experimental Results

Wikipedia offers its content as database backup dumps (wikidumps) freely available to download. Our application is based on the English wikidump[8] of July 2011. We have extracted redirections, titles, categories, and links from the wikidump into separate tables in MySQL. Using the steps outlined in Section 3.1, we built a table for the hierarchy with "Category:Applied sciences" as the root. The attributes of each table were indexed to speed up queries. Our experiment used $\alpha = 0.2$, $\beta = 0.5$, $\delta = 0.2$, and $\gamma = 0.6$. These values were found empirically to perform well over randomly selected samples.

We randomly selected 25 CS courses from 19 universities that can be transferred to University of Massachusetts Lowell (UML) according to the transfer dictionary. Each transfer course was compared to all 44 CS courses offered at UML, a total of 1,100 comparisons. The result was considered correct for each course if the real equivalent course in UML appears among the top 3 in the list of highest scores. We excluded all Wikipedia pages whose titles contained specific dates or were annotated as "magazine", "journal", or "album." We removed both general and domain stop words (such as "course," "book," and "student") from course descriptions. If a course description contains the keywords "not" or "no," e.g., "This course requires no computer programming skills," the segment after such keyword is ignored.

We tested our approach against the work by Li et al. (2006) and TF-IDF on the same data set of course descriptions. The accuracy of our proposed approach is 72%, compared to 52% using Li et al. (2006), and 32% using TF-IDF.

Algorithm	Pearson's correlation	p-value
Our approach	0.85	$6.6 \cdot 10^{-10}$
Li et al. (2006)	0.57	0.0006
TF-IDF	0.73	$2 \cdot 10^{-6}$

Table 1. Pearson's correlation of course relatedness scores with human judgments.

Since the transfer dictionary is always out of date, we found a few equivalent course pairs that were unintuitive. To make a more meaningful evaluation, we set up a human judgment data set. We gave 6 annotators (CS students and professors) a list of 32 pairs of courses, with only course titles and descriptions. They independently evaluated whether each pair is equivalent on a scale from 1 to 5. We averaged their evaluations for each pair and converted the scale from [1,5] to [0,1]. Next, we ran our approach, the work by Li et al. (2006), and TF-IDF on the same 32 course pairs. Table 1 reports the Pearson's correlation coefficient of course relatedness scores with human judgment, and statistical significances. Our approach has a higher correlation to the human judgment data set compared to previ-

[7] The computation of $MAXIN$ and $MAXOUT$ could be time-consuming. They are therefore based on the entire Wikipedia instead of the constructed hierarchy to avoid the recalculation when the domain changes. This also ensures the maximum linkage information is unbiased for every domain. For the July 2011 wikidump, page "Geographic coordinate system" has the most in-links, a total of 575,277. Page "List of Italian communes (2009)" has the most out-links, a total of 8,103.

[8] http://dumps.wikimedia.org/enwiki/20110722/

ous work. Furthermore, a smaller p-value indicates our approach is more likely to correlate with human judgment.

During the experiment, we have found some misclassified categories in the wikidump.[9] For example, "Category:Software" has over 350 subcategories with names similar to "Category:A-Class Britney Spears articles," or "Category:FA-Class Coca-Cola articles." None of these appears in the Wikipedia website or the Wikipedia API[10] as a subcategory of "Category:Software." More study is required on how they are formed.

5 Conclusion

This paper presents a domain-specific algorithm to suggest equivalent courses based on analyzing their semantic relatedness using Wikipedia. Both accuracy and correlation suggest our approach outperforms previous work. Future work includes comparing our approach with ESA, experimenting on more courses from more universities, and adapting our work to courses in other languages.

Acknowledgments

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

References

Gergana Bounova. 2011. *Topological Evolution of Networks: Case Studies in the US Airlines and Language Wikipedias*. Ph.D. thesis, MIT.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating Wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32:13–47.

David Buscaldi and Paolo Rosso. 2006. Mining knowledge from Wikipedia from the question answering task. In *Proc. 5th Int'l. Conf. on Language Resources & Evaluation*, Genoa, Italy.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. 2007. The google similarity distance. *IEEE Trans. on Knowledge & Data Engineering*, 19:370–383.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, Univ. of Edinburgh.

[9] We have analyzed wikidumps of July 2011 and Oct 2010 and the problem persists in both versions.

[10] https://www.mediawiki.org/wiki/API

Evgeniy Gabrilovich and Shaul Markovitch. 2009. Wikipedia-based semantic interpretation for NLP. *J. AI Research*, 34:443–498.

Graeme Hirst and David St-Onge, 1998. *Lexical Chains as Representation of Context for the Detection and Correction Malapropisms*. The MIT Press.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

Claudia Leacock and Martin Chodorow. 1998. Using corpus statistics and Wordnet relations for sense identification. *Computational Linguistics*, 24:147–165.

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 Trans. on Knowledge and Data Engineering*, 18.

Dekang Lin. 1998. Extracting collocations from text corpora. In *Workshop on Computational Terminology*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proc. 16th ACM Conf. on Information & Knowledge Management*, pages 233–242.

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

Yuan Ni, Lei Zhang, Zhaoming Qiu, and Wang Chen. 2010. Enhancing the open-domain classification of named entity using linked open data. In *Proc. 9th Int'l. Conf. on the Semantic Web*, pages 566–581.

Simone Paolo Ponzetto and Michael Strube. 2007. Knowledge derived from Wikipedia for computing semantic relatedness. *J. AI Research*, 30:181–212, October.

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. 15th Int'l. Conf. on WWW*.

Nuno Seco, Tony Veale, and Jer Hayes. 2004. An intrinsic information content metric for semantic similarity in Wordnet. In *Proc. 16th European Conf. on AI*.

Fei Wu. 2010. *Machine Reading: from Wikipedia to the Web*. Ph.D. thesis, Univ. of Washington.

Beibei Yang and Jesse M. Heines. 2011. Using semantic distance to automatically suggest transfer course equivalencies. In *Proc. 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 142–151.

Jonathan Yu, James A. Thom, and Audrey Tam. 2007. Ontology evaluation using Wikipedia categories for browsing. In *Proc. 16th ACM Conf. on Information and Knowledge Management*, pages 223–232.