The Law School Admission Council (LSAC) is a nonprofit corporation that provides unique, state-of-the-art products and services to ease the admission process for law schools and their applicants worldwide. Currently, 218 law schools in the United States, Canada, and Australia are members of the Council and benefit from LSAC’s services. All law schools approved by the American Bar Association are LSAC members. Canadian law schools recognized by a provincial or territorial law society or government agency are also members. Accredited law schools outside of the United States and Canada are eligible for membership at the discretion of the LSAC Board of Trustees; Melbourne Law School, the University of Melbourne is the first LSAC-member law school outside of North America. Many nonmember schools also take advantage of LSAC’s services. For all users, LSAC strives to provide the highest quality of products, services, and customer service.

Founded in 1947, the Council is best known for administering the Law School Admission Test (LSAT®), with about 100,000 tests administered annually at testing centers worldwide. LSAC also processes academic credentials for an average of 60,000 law school applicants annually, provides essential software and information for admission offices and applicants, conducts educational conferences for law school professionals and prelaw advisors, sponsors and publishes research, funds diversity and other outreach grant programs, and publishes LSAT preparation books and law school guides, among many other services. LSAC electronic applications account for 98 percent of all applications to ABA-approved law schools.

© 2013 by Law School Admission Council, Inc.

All rights reserved. No part of this work, including information, data, or other portions of the work published in electronic form, may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission of the publisher. For information, write: Communications, Law School Admission Council, 662 Penn Street, PO Box 40, Newtown, PA, 18940-0040.

This study is published and distributed by LSAC.
Executive Summary

Text similarity measurement provides a rich source of information and is increasingly being used in the development of new educational and psychological applications. However, due to the high-stakes nature of educational and psychological testing, it is imperative that a text similarity measure be stable (or robust) to avoid uncertainty in the data. The present research was sparked by this requirement. First, multiple sources of uncertainty that may affect the computation of semantic similarity between two texts are enumerated. Second, a method for achieving the requirement of a robust text similarity measure is proposed and then evaluated by applying it to data from the Law School Admission Test (LSAT). While further evaluation of the proposed method is warranted, the preliminary results were promising.

Introduction

The measurement of semantic similarity between two texts has numerous applications. It has been used for text classification (Rocchio, 1971), word sense disambiguation (WSD; Lesk, 1986; Schutze, 1998), summarization (Lin & Hovy, 2003; Salton, Singhal, Mitra, & Buckley, 1997), and automatic evaluation of machine translation (Papineni, Roukos, Ward, & Zhu, 2002). In psychometrics, it has been used for:

- Expanding feature space for decision-tree-based item difficulty modeling by adding semantic features (Belov & Knezevich, 2008; Paap, He, & Veldkamp, 2012; Sheehan, Kostin, & Persky, 2006). For each item (i.e., test question) these features include: semantic similarity between passage and key, semantic similarity between key and distractors, self-similarity of the passage, and so forth.
- Generating enhanced item-writing materials (Becker & Olsen, 2012), where semantic similarity is used to link a passage with a subset of chapters from textbooks in order to generate a useful compilation of information that would save time for item writers.
- Identifying pairs of enemy items (Belov & Kary, 2012; Li & Shen, 2012). Two items are called “enemies” if and only if they were judged by a test specialist to be too similar in subject matter to allow both to be part of the same test.
- Automatic essay scoring (Cook, Baldwin, & Clauser, 2012; Foltz, 2012), where new unscored essays are classified into one of the clusters of scored essays.
- Screening for psychiatric disorders (He, Veldkamp, & de Vries, 2012), where blog posts or diary entries of patients are screened for features indicating the presence of a disorder, which enables early detection and diagnosis.

A typical approach to computing text similarity is to use a matching method (Mihalcea, Corley, & Strapparava, 2006), which produces a similarity score based on the number of semantically similar pairs of words from two texts. There are a large number of word-to-word semantic similarity measures using approaches that are either
knowledge based (Leacock & Chodorow, 1998; Resnik, 1998; Wu & Palmer, 1994) or corpus based (Turney, 2001).

As usual with artificial intelligence, there is no universal method for computing semantic similarity between two texts that would work well for any data. Moreover, even mathematical formalization of semantics and semantic similarity is still an active area of research (Palmer, 2011).

The present research was sparked by a common requirement for multiple applications of computational linguistics in psychometrics. Text is a rich source of information increasingly used in the development of new educational and psychological measures. However, due to the high-stakes nature of educational and psychological testing, these measures need to be robust given the uncertainty in data. This report first analyzes the major stages of computing text similarity from the standpoint of uncertainty, and then demonstrates how the uncertainty negatively affects multiple applications enumerated above. A new measure of text similarity that is stable under uncertainty is introduced based on the method of robust discrete optimization by Bertsimas and Sim (2003). This robust measure utilizes error estimates of semantic similarity between words from two given texts. In order to estimate these errors, a Monte Carlo approach is proposed. An application of the robust semantic similarity measure to real data is shown.

First, the computation of semantic similarity between two words (assuming that their parts of speech and senses are identified) using the semantic dictionary called WordNet is discussed. Second, real data from past administrations of the Law School Admission Test (LSAT) used in this study is described. Third, the major stages of computing text similarity and corresponding sources of uncertainty are analyzed, followed by an illustration of the impact of uncertainty on WSD (as one of the major stages). Then the robust measure of text similarity is introduced, followed by a Monte Carlo approach to estimate error bounds. The report concludes with an application example and a brief summary.

**WordNet**

WordNet (Fellbaum, 1998) is a semantic dictionary designed as a graph. Each vertex of the graph is called a *synset*, which represents a specific meaning (*sense*) of a word. It includes the word, its explanation (gloss), and its synonyms. Synsets are connected to one another through explicit semantic relations. In WordNet, given a part of speech (noun, verb, adjective, adverb) and semantic relation (hyponyms for nouns and verbs; synonyms for adjectives and adverbs), a word can have multiple senses, each corresponding to a particular synset (Fellbaum, 1998). Therefore, the terms sense and synset will be used interchangeably throughout this report. For a given word, all of its senses are sorted from the most common to the least common. For a noun, each sense defines a unique path (determined by “this is a kind of” semantic relation) to the noun *entity*. Figure 1 shows two senses and their paths for the noun *otter*. The noun *otter* could refer either to fur or to mammal. This noun in the WordNet corpus appeared more often as fur (Figure 1).
FIGURE 1. Two senses and their paths for the noun otter

The semantic similarity between two synsets can be computed from the distance between them in the semantic graph. The shorter the distance from one synset to another, the more similar they are (Resnik, 1998). The distance \( \Delta(x, y) \) is the number of synsets between (and including) synset \( x \) and synset \( y \). In order to illustrate how semantic differences can be calculated, let’s examine the paths of the corresponding senses for two nouns: psychometrics and psychophysics (Figure 2).
FIGURE 2. Semantic paths for two nouns in WordNet: (a) psychometrics; (b) psychophysics
The following equation to compute semantic similarity between two synsets was used:

\[ \delta(x, y) = \frac{1}{\Delta(x, y)} \]  

(1)

Obviously, \( \delta(x, y) \in [0,1] \). The similarity is 0 if the paths have no common synsets or no path exists. The similarity is 1 if two synsets are the same (e.g. two synonyms). For example, if \( x = \text{psychometrics} \) and \( y = \text{psychophysics} \), then the similarity computed by Equation (1) is 0.25, since four synsets are counted, beginning with \( \text{psychometrics} \) and terminating with \( \text{psychophysics} \) (Figure 3). Other methods for computing semantic similarity between two words can be found in Budanitsky and Hirst (2006).

FIGURE 3. An example showing the shortest distance between psychometrics and psychophysics.

Semantic similarity between two words is used for computing semantic similarity between two texts in the following manner. One could apply Equation (1) to compute semantic similarity for each pair of synsets, where the first synset is from the first text and the second synset is from the second text. This results in a weighted bipartite graph, where each vertex represents a synset and each edge has a weight—semantic similarity between corresponding synsets. This graph is often represented as an \( m \times n \) matrix \( R \) such that \( r_{ij} = \delta(x_i, y_j) \) is semantic similarity between synset \( x_i \) and synset \( y_j \). Finally, an optimal solution to the maximum weighted bipartite matching problem (MWBM; Papadimitriou & Steiglitz, 1982) will provide an estimate of semantic similarity between two texts.
This method for computing semantic similarity between texts seems rather straightforward; however, it highly depends on a correct transformation of a text to a set of synsets. For example, when the sentence *an otter spends half of his life on land* is transformed into a set of synsets, the algorithm has to recognize that the least popular synset for the noun *otter* has to be selected; otherwise the wrong path will be used to calculate semantic distances. These and other uncertainties might affect the accuracy of the semantic similarity estimate. But before the elaboration on how the various uncertainties play a role, the studied data is introduced.

**Data Type**

This study uses passages from Logical Reasoning (LR) items from past administrations of the LSAT. These passages ranged from about 30 to 100 words in length, with a mean word count of about 60.

LR passages are written in ordinary English and do not presuppose a specialized vocabulary. Most LR passages are based on short arguments drawn from published sources such as scholarly books and periodicals, newspapers, and general interest magazines. Any LR passages that are not based on published sources are written to resemble real-world argumentation.

Most LR subtypes have passages that take the form of arguments, consisting of an explicitly drawn conclusion, one or more premises offered in support of the conclusion, and (optionally) background information that sets the stage for the premises and conclusion. Here is a sample passage from the datasets:

Biologists have noted reproductive abnormalities in fish that are immediately downstream of paper mills. One possible cause is dioxin, which paper mills release daily and which can alter the concentration of hormones in fish. However, dioxin is unlikely to be the cause, since the fish recover normal hormone concentrations relatively quickly during occasional mill shutdowns and dioxin decomposes very slowly in the environment.\(^1\)

The first two sentences of this passage are largely devoted to presenting background information. The third sentence states the conclusion of the argument—that dioxin is unlikely to be the cause of the observed reproductive abnormalities in fish. Two premises are then offered in support of this conclusion: (a) the fish in question recover normal hormone concentrations relatively quickly when the mill shuts down, and (b) dioxin decomposes very slowly in the environment.

\(^1\) All LR passages ©1997–2007 by Law School Admission Council, Inc.
A dataset used in the study was compiled by a test developer, drawing on operational experience in reviewing LSAT test forms for subject matter overlap. The dataset consists of 10 LR passages. Two of the 10 passages \((P_1, P_2)\) were judged by the test developer to be semantically similar to each other. For the purposes of this study, two passages were deemed “semantically similar” (i.e., enemies) if and only if they were judged to be too similar in subject matter to allow both to be part of the same LSAT test form. The other eight passages \((P_3, \ldots, P_{10})\) were judged to be semantically dissimilar from both \(P_1\) and \(P_2\). For the purposes of this study, two passages were deemed “semantically dissimilar” if and only if there would be no reasonable subject-matter-based objection to having both passages on the same test form.

Sources of Uncertainty in Computing Text Similarity

This section analyzes major stages of a typical process to compute semantic similarity between two texts. Roughly speaking, the whole process starts with tokenization, whereby a text is partitioned into a list of words. After that, the corresponding part of speech is tagged for each individual word. The next step is stemming, whereby each word is transformed to each root. Finally, the sense for each word is identified. Basically, these stages transform given text into a list of WordNet synsets that should represent the semantic content of the text.

It will be demonstrated that each stage has a potential source of uncertainty impacting an optimal solution to MWBM. Obviously, earlier stages have a larger impact than later stages. Given two texts, a typical procedure for computing semantic similarity consists of the following stages:

**Stage 1 (Tokenization):** In this step, each text is partitioned into a list of words by using regular expressions, where the stop words are removed. Stop words are frequently occurring, insignificant words like a, the, for, in, on, etc. This is where the first source of uncertainty occurs. Some stop words can be part of collocations like: one of a kind, also ran, down on his luck. Therefore, the removal of stop words may also remove some important collocations carrying a semantic load for the text. For example, one of a kind is disintegrated into semantically different entities—one and kind—thus creating uncertainty. See Belov and Kary (2012) for a method of tokenization such that all collocations existing in WordNet are identified in the text.

**Stage 2 (Part-of-speech tagging):** For each word the corresponding part of speech is identified. Although a noun form and a verb form of the same word often bear similar meaning (e.g., use, speed, study), in general this does not have to be the case. Moreover, an error made at this stage may propagate further with a potentially large negative effect (e.g., fashion, harp).

**Stage 3 (Stemming):** Each word is transformed to its root form. For example, the word homes from one text and the word home from another should be transformed to one root form home. This stage is known for two types of errors. The first type is
called *understemming*, which is when words that should be merged together (*adhere* and *adhesion*; *perceived* and *perceptible*) remain distinct after stemming. The second type is called *overstemming*, which is when words that are really distinct will be wrongly merged (*experiment* and *experience*; *relation* and *relativity*). For more details on stemming errors see Paice (1996).

**Stage 4 (Word sense disambiguation):** As shown above, each word in WordNet may have several senses for each part of speech. For example, the noun *seal* has 9 senses (Figure 4); and if one text is about seals in documents and another text is about seals in the Baltic Sea, then incorrect sense identification may cause a higher semantic similarity between two semantically different texts. Sense identification is often performed by analyzing the text around each word (Banerjee & Pedersen, 2002; Lesk, 1986).

![WordNet 2.1 Browser](image)

**FIGURE 4.** *All senses for the noun seal*
Final stage (Computing semantic similarity): Given two lists of synsets \( X = (x_1, x_2, \ldots, x_m) \) and \( Y = (y_1, y_2, \ldots, y_n) \), identified for both texts in previous stages, compute the \( m \times n \) matrix \( R \) such that \( r_{ij} = \delta(x_i, y_j) \) is the semantic similarity between synset \( x_i \) and synset \( y_j \) [see Equation (1)], \( i = 1, 2, \ldots, m \), \( j = 1, 2, \ldots, n \). Then the corresponding instance of MWBM is formulated and solved:

\[
\begin{align*}
\text{maximize} \quad & \frac{1}{\min(m, n)} \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} z_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{n} z_{ij} \leq 1 \quad j = 1, 2, \ldots, n \\
& \sum_{i=1}^{m} z_{ij} \leq 1 \quad i = 1, 2, \ldots, m \\
& z_{ij} \in \{0, 1\}
\end{align*}
\]

Since \( r_{ij} \in [0, 1] \) and \( z_{ij} \in \{0, 1\} \), the objective of Problem (2) is between 0 and 1. Problem (2) can be solved efficiently by the Hungarian method (Papadimitriou & Steiglitz, 1982). However, Stages 1–4 may provide incorrect lists \( X \) and \( Y \), which will cause an optimal solution of Problem (2) to be far from its optimal solution with correct lists \( X \) and \( Y \). The next section will demonstrate this negative effect and discuss its implications.

**Impact of Uncertainty in Computing Semantic Similarity**

To illustrate the impact of uncertainty on computing semantic similarity, this report will focus on one of the sources of uncertainty introduced in the previous section. The impact of uncertainty in Stage 4 (WSD) on the optimal solution of the MWBM in Problem (2) is demonstrated on the real dataset described earlier in the report. First, the heuristic by Belov and Kary (2012) was applied to identify lists of nouns from each of the 10 passages. Then the following methods of WSD were applied to each list of nouns:

**WSD1**: Manual method performed by a test developer analyzing the actual passages and comparing the nouns in these passages to the available synsets from WordNet. It is assumed that this method provides a true value of sense for each noun in the list. Then an optimal solution to the corresponding instance of Problem (2) is considered to be a true value of semantic similarity between passages.

**WSD2**: Method assigning the most popular sense. For words having more than one synset, this method might assign the wrong one.
**WSD3:** Method described by Banerjee and Pedersen (2002). A modification of the classical algorithm by Lesk (1986), Banerjee and Pedersen’s method performs sense identification by analyzing the text around each word.

The results of applying the above WSD methods are presented in Figure 5, where semantic similarity was computed between \( P_1 \) and \( P_2, P_3, \ldots, P_{10} \). To compute semantic similarity between \( P_i \) and \( P_j \), \( i = 2, 3, \ldots, 10 \), a corresponding instance of Problem (2) was solved by IBM ILOG CPLEX 12.3 (International Business Machines Corporation, 2012); there were \( 3 \times 9 = 27 \) instances solved. Each method of WSD resulted in three different orders of nine passages, where the passages were sorted by their semantic similarity to \( P_1 \). If orders were identical, one would see three monotonically decreasing curves on Figure 5. However the orders provided by WSD2 and WSD3 are different from the true order provided by WSD1: \( P_2, P_{10}, P_4, P_3, P_8, P_5, P_6, P_9, P_7 \). For numerical estimation of the violation of monotonicity, the following measure is used:

\[
v = \sum_{i=2}^{9} \left| \min(0, s_{j,i} - s_{j,i-1}) \right|
\]

where \( s_{j,i} \) is the semantic similarity between \( P_1 \) and \( P_{j,i} \), \( i = 2, 10, 4, 3, 8, 5, 6, 9, 7 \) (indices of passages sorted by their true similarity to \( P_1 \)). In Figure 5, WSD1 provided \( v = 0 \); WSD2 provided \( v = 0.044 \); and WSD3 provided \( v = 0.055 \).

---

**FIGURE 5.** Semantic similarity computed between \( P_1 \) and \( P_2, P_3, \ldots, P_{10} \) without taking into account uncertainty [see Problem (2)], where three different methods of WSD were applied.
Such unstable behavior of the semantic similarity measure will negatively affect various applications in psychometrics and test development (see Introduction). For example, if a decision tree is built to predict item difficulty and the semantic similarity between the passage and the key is used for splitting items in each node, then different methods of WSD will produce different and incorrect regression trees.

This section demonstrated on real data that uncertainty in elements of matrix $R$ (caused by different WSD methods as shown above) negatively influence an optimal solution to Problem (2)—that is, the measure of semantic similarity between two texts. Therefore, a robust method for computing semantic text similarity is needed.

**Robust Measure of Semantic Similarity**

Optimization under uncertainty is a well-studied topic of optimization. There are two major approaches to finding a solution that is stable under random changes in parameters of the problem: stochastic optimization (Birge & Louveaux, 1997) and robust optimization (Bental, Ghaoui, & Nemirovski, 2009). Robust optimization has applications in multiple fields (Bertsimas, Brown, & Caramanis, 2011), including psychometrics (Veldkamp, 2013). This report will employ robust optimization due to its controllable conservatism and computational tractability.

Assume that every element $r_{ij}$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$ of matrix $R$ [see Problem (2)] is random and $r_{ij} = [b_q - d_q, b_q]$, $q = n(i - 1) + j$, $q = 1, 2, ..., mn$. Problem (2) can be restated as the following:

$$\frac{1}{\min(m, n)} \max_{z_q} \sum_{q=1}^{mn} c_q z_q$$
subject to $Az \leq 1$, $z_q \in \{0, 1\}$

(4)

where $c_q = r_{ij}$, $A$ is an $(m+n) \times mn$ matrix of 0s and 1s built from inequalities of Problem (2), and $1$ is an $(m+n)$-dimensional vector where each coordinate equals 1. Following the method by Bertsimas and Sim (2003), the robust counterpart of Problem (4) is formulated as follows:

$$\frac{1}{\min(m, n)} \max_{z_q} \left[ \sum_{q=1}^{mn} b_q z_q - \max_{(S|S|n\times |S|n)} \sum_{q=1}^{mn} d_q z_q \right]$$
subject to $Az \leq 1$, $z_q \in \{0, 1\}$

(5)

where $N = \{1, 2, ..., mn\}$ and $\Gamma$ is a parameter indicating the level of uncertainty. An optimal solution to Problem (5) defines a semantic similarity above a certain threshold that holds under uncertainty in at most $\Gamma$ cells of matrix $R$. Therefore, the estimate of
semantic similarity provided by an optimal solution to Problem (5) is usually more conservative than the estimate of semantic similarity provided by an optimal solution to Problem (4). In this report $\Gamma = \min(m, n)$, which is the most conservative since no more than $\min(m, n)$ elements of vector $z$ can be equal to 1 due to the constraint $Az \leq 1$.

Clearly, Problem (5) cannot be solved directly. However, Bertsimas and Sim (2003) developed a method to solve (5) by solving $mn + 1$ problems directly:

$$
\frac{1}{\min(m, n)} \max_{l=1}^{mn+1} \left[ -\Gamma d_i + \max \left\{ \sum_{q=1}^{m} b_q z_q - \sum_{q=1}^{n} (d_q - d_i) z_q \right\} \right]
$$

subject to

$$
Az \leq 1, \quad z_q \in \{0,1\}
$$

(6)

where, without loss of generality, $d_1 \geq d_2 \geq ... \geq d_{mn} \geq d_{mn+1} = 0$ is assumed.

However, in order to solve Problem (6), error bounds $[b_q - d_q, b_q], \ q = 1, 2, ..., mn$ have to be estimated first. This means that for every combination of words from both texts, the impact of errors on the distance between both words has to be estimated. A Monte Carlo approach can be applied to assign a probability to the corresponding decision made about each word at Stages 1–4 (tokenization, part-of-speech tagging, stemming, WSD). A concrete realization of this Monte Carlo approach depends on the stages involved. A Monte Carlo algorithm for WSD was developed to estimate the probability distribution of senses for each word in a given text (see Appendix A).

### Results of the Monte Carlo Approach to Estimating Error Bounds

The following Monte Carlo approach estimates the level of uncertainty in the parameters of the robust optimization problem in (6). First, Monte Carlo WSD (see Appendix A) is applied to each passage, resulting in a distribution of senses for each word. Second, assuming independence, a distribution of semantic similarity $p_{ij}$ is computed by employing Equation (1) and distributions of senses for each pair of words $x_i, y_j$ from two texts. Figure 6 illustrates 10 distributions $p_{ij}$ between random pairs of words $x_i, y_j$ for two given texts.
Finally, mean $\mu_{ij}$ and standard deviation $\sigma_{ij}$ are computed from the distribution of semantic similarity $p_{ij}$ in order to set the error bounds. In this report, the bounds are set as follows: $b_q = \mu_{ij} + \sigma_{ij}$, $d_q = 2\sigma_{ij}$, $q = n(i-1) + j$, $q = 1, 2, ..., mn$, since uncertainty in the distances between two synsets could result in either longer or shorter distances. One can be more conservative and use the following bounds: $b_q = \mu_{ij}$, $d_q = \sigma_{ij}$, $q = n(i-1) + j$, $q = 1, 2, ..., mn$.

The optimization problem in (6) was solved to find the similarity between the first and the nine other texts in our dataset. The results of applying the robust method for estimating semantic similarity were compared to the true value resulting from application of the WSD1 method. The nine texts were ordered based on true semantic similarity. The results are plotted in Figure 7, which indicates an almost monotonic behavior of the resulting robust semantic similarity measure (dashed line) with $v = 0.011$. In other words, the robust semantic similarity method was capable of ordering the nine texts almost correctly. Comparing Figure 7 with Figure 5, one can conclude that an optimal solution to Problem (6) provides a more stable measure of text similarity than an optimal solution to Problem (2).

As expected, robust optimization results in a more conservative estimate of the semantic similarity between the texts (Figure 7).
FIGURE 7. Dashed line shows robust semantic similarity [see Problem (6)] computed between $P_1$ and $P_2, P_3, \ldots, P_{10}$, where Monte Carlo WSD (see Appendix A) was applied (see Figure 5).

Conclusion

The major purpose of this report is to raise awareness about potentially negative effects caused by uncertainty generated by various computational linguistics procedures. The effect of uncertainty on semantic similarity between two texts was studied using real data from the LSAT. It was demonstrated that the uncertainty caused by the WSD can negatively affect various applications in psychometrics. To alleviate the problem of uncertainty, a robust semantic similarity measure was proposed. This measure employs the method of robust discrete optimization by Bertsimas and Sim (2003). However, in order to apply this method one has to estimate error bounds for each element of matrix $R$. For this a Monte Carlo approach was proposed. In an experiment using real LSAT data, the developed Monte Carlo method provided reasonable error bounds. However, these results are only preliminary (just to demonstrate the potential of the proposed method), and further testing with real data is needed.
References


Appendix A

Monte Carlo word sense disambiguation algorithm (MCWSD)

**Parameters:** Number of Monte Carlo iterations \( k \) (\( k = 100 \) was used in this paper) and size of search window \( h \) (\( h = 2 \) was used in this paper).

**Input:** List of words \( L \) (in this paper only nouns were considered).

**Output:** Empirical distribution of senses for each word from list \( L \).

**Step 1:** Randomly select word \( w_i \) from list \( L \).

**Step 2:** Form a list of neighbored words \( G = (w_{i-h}, \ldots, w_i, \ldots, w_{i+h}) \).

**Step 3:** Generate a random vector of corresponding senses \((s_{i-h}, \ldots, s_i, \ldots, s_{i+h})\), where \( s_j \) is a sense selected randomly from all senses of word \( w_j \), \( j = i-h, i-h+1, \ldots, i+h \).

Compute a score:

\[
\theta = \sum_{i-h}^{h} \sum_{j=h}^{i+h} g(s_{i+h}, s_{i+j})^2, \quad (A-1)
\]

where \( g(s_{i+h}, s_{i+j}) \) is the number of words in the longest sequence of consecutive words (stop words are not counted) that occurs in both glosses (augmented according to Banerjee and Pedersen (2002)) corresponding to synsets \( s_{i+h} \) and \( s_{i+j} \).

Senses \( s_{i+h} \) and \( s_{i+j} \) for which \( g(s_{i+h}, s_{i+j}) > 0 \) are marked as **identified**.

**Step 4:** Add \( \theta \) to the weights of identified senses from \((s_{i-h}, \ldots, s_i, \ldots, s_{i+h})\).

**Step 5:** Set \( k := k - 1 \). If \( k > 0 \) then go to Step 1.

**Step 6:** Compute the empirical distribution of senses for each word in list \( L \) using weights identified by Monte Carlo iterations at Steps 1–5.

The above algorithm (MCWSD) is a Monte Carlo modification of the algorithm by Banerjee and Pedersen (2002). MCWSD can be used to compute distributions of senses or to estimate the most probable sense. The algorithm by Banerjee and
Pedersen (2002) enumerates through every possible vector of synsets \((s_{i-h}, \ldots, s_i, \ldots, s_{i+h})\) in order to find a maximum of score \(\theta\), which makes it limited to a small neighborhood \((h \leq 2)\). MCWSD does not have this limitation and is able to explore larger neighborhoods \((h > 2)\). Lower values of parameter \(k\) will result in faster convergence of MCWSD; higher values of parameter \(k\) will result in deeper search and possibly a better solution provided by MCWSD.