Applying Semantic Analysis to Training, Education, and Immersive Learning

Robby Robson
Eduworks Corporation
Corvallis, OR
robby.robson@eduworks.com

Fritz Ray
Eduworks Corporation
Corvallis, OR
fritz.ray@eduworks.com

ABSTRACT

The last decade has seen major advances in the areas of natural language processing and semantic analysis. Theoretical advances and increased computational power have resulted in applications that detect topics and sentiments in communications, automatically classify unstructured data in enterprise settings, and win Jeopardy contests. This paper surveys how these same methods apply to a variety of problems in education and training. Applications include automatic grading and question generation, guiding the behavior of intelligent tutoring systems, aligning content to competencies and educational standards, and improving search in digital repositories. This paper describes the methods, explains how they are applied and evaluated, and discusses their potential for use virtual worlds and immersive learning environments.

ABOUT THE AUTHORS

Dr. Robby Robson began developing web-based learning content and learning management systems in 1995, chaired the IEEE Learning Technology Standards Committee from 2000 – 2008, and has helped dozens of organizations develop eLearning technology strategies. He led several Department of Defense projects that explored the use of emerging technologies for training and has contributed extensively to SCORM and to the theory and practice of reusable design. His most recent work is in applications of semantic technology and natural language processing to training and competency management. Dr. Robson co-founded Eduworks in 2001 where he has guided research, services and product development. He holds a doctorate in mathematics from Stanford University and has held posts in both academia and industry.

Edward "Fritz" Ray is the lead software engineer at Eduworks Corporation where he has architected and developed applications in areas ranging from educational digital libraries to semantic analysis, semantic search, competency management, and patent analysis. In addition, Fritz has researched and made improvements to semantic similarity algorithms and conceived and developed middleware for rapid web service development and deployment in domains that use unstructured data.

Applying Semantic Analysis to Training, Education, and Immersive Learning

Robby Robson
Eduworks Corporation
Corvallis, OR
robby.robson@eduworks.com

Fritz Ray
Eduworks Corporation
Corvallis, OR
fritz.ray@eduworks.com

INTRODUCTION

Semantic analysis broadly refers to using computers to determine and analyze the meaning of natural language. Semantic analysis is a subfield of the more general fields of *natural language processing* (NLP) and *computational linguistics.* Typical semantic analysis problems include:

Word sense disambiguation: Determine the meaning of a particular word in a given context, e.g. in the sentence "a tank is parked in the exhibit hall," does "tank" refer to an Army vehicle or a tank of water?

Topic detection: Determine the topics discussed in a passage of text. A simpler form of topic detection is keyword generation, i.e. automatically associating a useful set of keywords with a document.

Semantic Similarity: Find resources that are related in meaning to a given document or text passage.

This paper describes how computers solve problems of this nature and how they automate processes such as competency alignment, grading, and question generation. We describe a number of tools and techniques used in traditional online learning and that appear fundamental to the operation of systems such as the U.S. Army Generalized Intelligent Framework for Tutoring (Goldberg, Holden, Brawner et al., 2011), see Figure 4 at the end of the paper. We start by explaining how semantic analysis work and what type of results to expect.

DOCUMENTS AND CORPORA

Semantic analysis studies *documents*. A document is any digital object containing text, e.g. a Word™ or PDF™ file, a web page, a PowerPoint™ presentation, a spreadsheet, a blog, a tweet. An image, movie, game or simulation can also be a document if text can be extracted from it.

A collection of related documents is called a *corpus.* Researchers use corpora tagged by human experts to train and test semantic analysis algorithms. In a typical procedure, experts label documents in a training corpus with properties (e.g. their topics) and an algorithm is programmed or trained to replicate expert opinion. Researchers then use the algorithm to find all documents with a desired property in a second corpus tagged by experts. The percentage of retrieved documents that have the desired property (according the experts) is the *precision* of the algorithm. If the algorithm returns a ranked list of results (as is common with search algorithms) *precision at rank k* is the proportion of the highest ranked *k* results with the desired property. The *accuracy* of the algorithm is the percentage of documents that the algorithm correctly identifies as having and as not having the desired property. Researchers use other measures such as the *F-measure* (Tan, Steinbach, & Kumar, 2006) as well, but we will only discuss precision and accuracy.

Semantic analysis can be challenging for both people and computers. Researchers consider accuracies of 80% to be excellent. For example, the best performing word sense disambiguation algorithms have accuracies close to 90% only for the general sense of a word and dip to 70% for fine grain distinctions (Navigli, 2009). These rates may seem low, but they are similar to the rates of agreement observed among human annotators!

Despite their overall accuracy, computers can make mistakes that humans would easily avoid. In the 2011 Jeopardy contests that IBM's Watson won, Watson offered "Toronto" as a U.S. city and "Dorothy Parker" as the title of a reference book (Jackson, 2011). Errors of this nature serve to caution that there is a place for human review, especially in high-stakes applications.

TERMS

Documents are made up of words or, more generally, *terms.* A *term* is usually a single word but can also consist of multiple words such as "world class" or "ice cream." In semantic analysis, words with the same stem (e.g. "Navy,"

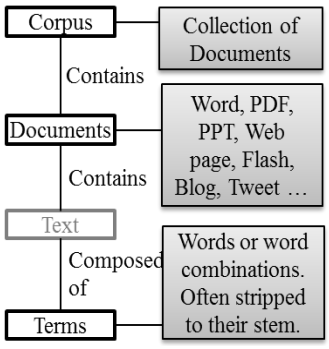

Figure 1: Object Hierarchy

"Navies," and "naval") are often treated as the same word or term. There are numerous algorithms for reducing words to their stems (Hull & Grefenstette, 1996)

Recognizing terms is relatively easy in ordinary prose, but much harder for technical, scientific or specialized terms such as "p37 protein" and "'A' school." It is also hard for computers to recognize the names of people, companies, places and brands. As representative data, (Zhang, Iria, Brewster et al., 2008) compared several term recognition algorithms and observed precisions ranging between 55% and 93% at rank 100 in a corpus 2,000 MEDLINE articles and in a corpus of Wikipedia articles describing 1,052 different animals.

FREQUENCY-BASED SIMILARITY

Many problems in semantic analysis reduce to measuring the semantic similarity between two documents. The simplest measures are *term frequency* (denoted TF) measures that count the co-occurrence of terms. Using TF, two documents are more similar if they contain more terms in common.

Not every term is of equal importance for measuring semantic similarity. Common words such as "the" and "is" appear in almost every document and provide no information. As terms become more specialized their co-occurrence becomes more indicative that two documents address the same topics. For example, two screens displaying the words "throttle valve" "carburetor," and "air filter" are more likely to address exactly the same procedure than two screens sharing only the terms "vehicle," "drive," and "repair."

A common way to adjust for the importance of terms is to use an *Inverse Document Frequency* (denoted IDF) weighting that varies inversely with the observed frequency of the term in a representative corpus. IDF weightings are usually computed for specific domains (e.g. vehicle repair or mathematics) using domain corpora (e.g. vehicle repair manuals or the resources in a mathematics digital library). If a domain corpus contains *N* documents and a term *t* appears in $n \geq 1$ of them, then a standard IDF weighting is $\log(N/n)$, but there are many other IDF weightings as well.

Combing term frequency with IDF gives a TF*IDF measure of semantic similarity as follows. Given a term *t* and a document *d*, let $TF(t,d)$ be the frequency of *t* in *d* and let $IDF(t)$ be an IDF weighting of *t.* Then the product $TF(t,d) * IDF(t)$ roughly measures how much *t* characterizes *d.* Given two documents d_1 and d_2, the product $TF(t,d_1) * TF(t,d_2) * IDF(t)^2$ of these measures is a number that is largest when *t* characterizes both documents well. The TF*IDF similarity between d_1 and d_2 is the sum of these products taken over all terms *t* and normalized to account for the size of the documents.

Linear algebra provides a useful formalism for defining TF*IDF. Let *T* be the real vector space with one basis element for each term *t.* Given a document *d*, let $\boldsymbol{d} \in T$ be the vector whose *t*-coordinate is $TF(t,d) * IDF(t)$. If d_1 and d_2 are two documents, their TF*IDF similarity is $(\boldsymbol{d_1} \cdot \boldsymbol{d_2})/(\|\boldsymbol{d_1}\| \cdot \|\boldsymbol{d_2}\|)$, which is cosine of the angle between $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. This formulation is the starting point for more sophisticated techniques such as Latent Semantic Analysis, described later in this paper.

There are many variations on TF*IDF involving different term weighting, different definitions of terms, different document weighting, and different methods of constructing corpora. In practice, relatively simple TF*IDF measures work well and are computationally efficient. However, all TF*IDF methods assume that the meaning of a document depends only on the set of its terms. This is called a "Bag of Words" approach. The success of TF*IDF measures is remarkable in light of the fact that they ignore many of the grammatical, structural, syntactical, and contextual clues that humans use when reading text.

COMPETENCY MATCHING USING TF*IDF

TF*IDF is used in many applications of semantic analysis to training and education. The first one we discuss is identifying the competencies, learning objectives or curriculum standards addressed by a given learning resource. This is called *competency matching* or *competency alignment.* Labeling resources with competencies helps users find appropriate resources. It is also necessary for interpreting training records. If one thinks of the definition of a competency as a document, then competency matching is the problem of determining which among a set of documents (the competencies) are most similar to a given document (a resource).

TF*IDF has been used for competency matching dating back at least to the 1990's. For example, in 1998 NETg developed *Precision Skilling*™, *a* tool that allowed consultants to match content to customer competencies using weighted keywords (NETg, 1998). The impetus for creating more sophisticated and automated versions of similar tools has come from K-12 education (where competency matching is called *alignment).*

Content Alignment in Education

Standards, by which we mean curriculum standards, are increasingly important in K-12. Teachers teach to

standards, schools test against standards, and state and federal agencies evaluate schools on the basis of these tests. As a result, when teachers look for resources they search based on standards. To enable such searches, educational libraries tag resources with the standards they address--i.e. align them with standards.

To facilitate alignment, several research groups associated with the National Science Foundation's National STEM Digital Library (NSDL) project have developed and studied automated alignment services. The most widely used service is the *Content Alignment Tool* (CAT) developed by the Center for Natural Language Processing at Syracuse University. CAT is based on TF*IDF methods and ranks standards according to how well they match a given document.

The first version of CAT achieved 56% precision at rank 1. This means that if resources were automatically tagged with the most related standard according to CAT, they would be tagged erroneously 44% of the time. Its developers concluded that CAT should make recommendations rather than to automatically tag resources (Diekema & Chen, 2005). However, results have significantly improved. In experiments run on the Digital Library for Earth Systems Education (www.dlese.org), the CAT team used metadata in addition to resources to determine standards alignment (Devaul, Diekema, & Ostwald, 2011). The current version of CAT, which uses resources and their metadata, achieves over 80% average precision at rank 1. This is higher than inter-rater reliability among humans tagging content with standards.

In 2011 the authors of this paper obtained another validation of the effectiveness of TF*IDF for resource alignment. We worked with researchers at Oregon State University who study alignment in the context of Teach Engineering (www.teachengineering.org), a digital library containing over 950 K-12 STEM resources (Reitsma, Marshall, Dalton et al., 2008). When authors contribute to Teach Engineering, they place their content in a topical hierarchy. We provided the Oregon State University researchers with a TF*IDF method exposed as a web service. They used the web service to match Teach Engineering resources to Teach Engineering topics and then compared the results to topics selected by the authors of the resources. This resulted in over 85% - 90% precision at rank 1.

LATENT SEMANTIC ANALYSIS (LSA)

Although TF*IDF methods are very useful, many applications rely on more sophisticated methods. The best known of these methods is *Latent Semantic Analysis* (LSA) or *Latent Semantic Indexing* (LSI) proposed by (Deerwester, Dumais, Furnas et al., 1990). We give a conceptual description of LSA that ties it to other methods described later.

LSA is based on two ideas. The first is reducing the dimensionality of the vector space *T* referred to in the linear algebra formulation of TF*IDF. This reduction results in a vector space with a basis that represents "topics" or "concepts" rather than just terms. The second idea is to examine the co-occurrence of terms in paragraphs rather than arbitrary documents because paragraphs typically encapsulate single topics or ideas. LSA assumes that if two terms tend to appear in the same paragraphs it is because they have some (latent) semantic relationship.

Computationally, LSA starts with a corpus of documents in a domain, breaks the documents into paragraphs and measures term frequencies in these paragraphs. LSA further uses the paragraphs to determine local weightings that are combined with global weightings based on the entire corpus or other domain information (Pincombe, 2004), (Hu, Cai, Wiemer-Hastings et al., 2007). These term frequencies and weights are used when representing a document *d* as a vector in *T*.

LSA then applies singular value decomposition (SVD) to compute an orthonormal basis for *T* and to select basis elements that carry the most information about the meaning of a document. The selected basis elements define a subspace *S* of *T*. The dimension of *S*, i.e. the number of selected basis elements, is usually chosen to be in the low hundreds. LSA represents a document in *S* by mapping it into *T* and projecting onto *S*. The space *S* is often called an LSA space.

Since the basis vectors of the LSA space *S* are vectors in *T*, each one is a linear combination

$$\alpha_1 t_1 + \alpha_2 t_2 + \cdots + \alpha_n t_n$$

of terms. As such, they evoke the notion of a *concept* or *topic* that involves a mix of the original terms. For example, the concept of "U.S. currency" might involve a mix of "dollar," "payment," "paper," "exchange," "bank," and "ATM." Conceptually, LSA replaces large vocabularies with a few hundred concepts that define the semantics of a domain.

In practice, LSA requires tweaking to improve accuracy, but it is a remarkably useful tool for automated grading, intelligent tutoring systems and semantic search. We describe these applications next.

Automated Testing and Grading

The first applications of LSA to education were used to automate test taking and grading, most significantly by

researchers in Colorado (Landauer, Laham, & Foltz, 2003). This research started in the 1996 – 1998 timeframe and has resulted in both patents (Foltz, Landauer, Laham II et al., 2002) and commercial products (Pearson, 2010).

LSA is used to take a multiple choice test by selecting the answer most semantically related to each question. In an early experiment this method scored about 64% on Test of English as a Foreign Language (TOEFL) on questions asking the student to identify synonyms. These scores were about the same as the pool of foreign applicants to U.S. Universities (Landauer & Dumais, 1997). In another experiment it did well enough to pass a class in psychology (Landauer, Foltz, & Laham, 1998).

LSA-based essay graders start with a master set of previously graded essays that represent a range of quality. LSA is used to compare student essays to the master set and to select the most related essays (typically ten). The score assigned to a student essay is the average of the grades on the most related master essays, weighted by the relatedness scores. (Foltz, Laham, & Landauer, 1999) describe this method as "holistic grading" and report correlations of 80% - 86% with grades assigned by the Educational Testing Service on GMAT tests, which is the same level of consistency observed among human graders.

Researchers have combined LSA with other methods for the purpose of essay grading. For example, (Pérez, Gliozzo, Strapparava et al., 2005) combined LSA with a variant of TF*IDF algorithm that looks at the weighted co-occurrence of single terms, pairs, triples and quadruples of terms in two documents. In their experiments they observed that the combined system performed better than either alone.

Applications to Intelligent Tutoring Systems

The tutoring research group at the University of Memphis started using LSA in AutoTutor in the late 1990's (Graesser, Franklin, Wiemer-Hastings et al., 1998). AutoTutor is an intelligent tutoring system that asks students questions and, as in automated essay grading, uses LSA to compare responses to "good" and "bad" answers (Wiemer-Hastings & Graesser, 1999). In AutoTutor, however, the comparison is used to determine a tutoring strategy rather than a grade. If a student demonstrates understanding by answering questions correctly, the tutor may present new material, whereas if a student demonstrates confusion, the tutor will attempt to remediate. More recent variants of AutoTutor use LSA to track student responses along the two dimensions of relevance and novelty. The tutor continues with new material if the student is making statements relevant to recently learned topics and attempts to remediate if responses are incorrect or only relevant to older topics. (Hu & Martindale, 2008).

Some AutoTutor systems also adjust strategies based on affective states such as confusion, flow, boredom, and frustration. These systems apply semantic analysis to detect the emotional content of text. Studies such as (Kim, Valitutti, & Calvo, 2010) indicate that textual analysis is better than random at detecting emotion but falls short of the required precision. (S. K. D'Mello, Craig, Witherspoon et al., 2008) added a machine learning layer but obtained similar results. To reach the desired levels of precision, some tutors use facial expressions and posture (S. D'Mello & Graesser, 2012), while others use sensor data such as heart rate and eye movement. Semantic analysis of textual responses is used as part of affective state evaluation but does not alone suffice.

SEMANTIC SEARCH

Semantic analysis can significantly improve searches for learning and training resources. Word sense disambiguation helps find resources that use common words in specialized ways, and automated meta-tagging enables users to filter by parameters such as interactivity level and learning objectives. Automated meta-tagging is the next subject we address.

Automated Metadata Generation

Automated metadata generation, or *automated metadata extraction*, is the process of using computers to tag resources with metadata (Greenberg, 2003). Automation is important because manual metadata tagging is labor intensive and often neglected.

Semantic analysis is well-suited for automatically generating titles, learning objectives, keywords, interactivity levels and other metadata elements that appear in standards such as the IEEE Learning Object Metadata (IEEE, 2002) standard adopted by SCORM. In a 2009 research project the authors of this paper developed a desktop tool that automatically generated metadata for large U.S. Marine Corps SCORM packages and registered the metadata records in the ADL Registry. The Marine Corps processed 56 objects in a trial that auto-generated several metadata fields including title, description, keyword and author. We used a mix of semantic analysis algorithms and machine learning algorithms to recognize titles, authors, and keywords and to extract descriptions. Users could edit these fields or leave them "as is" prior to registration in the ADL Registry.

According to personal user logs, the metadata records generated for 47% of the files required no changes and 26% required only minor revision to some field. The blue bars in Figure 2 show the proportion of each metadata field that users registered with no changes.

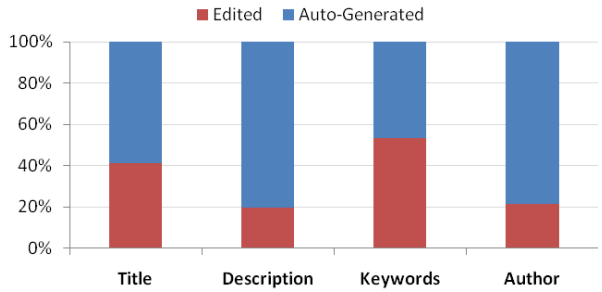

Figure 2: Source of metadata for registered objects

The above results demonstrate the potential of automated metadata generation. Other researchers have obtained similar results. In an extensive paper, (Ochoa & Duval, 2009) define quality metrics for metadata in digital libraries and study several sets of manually and automatically generated metadata records. They conclude that "Owing to the recent developments in automatic metadata generation and interoperability between digital repositories, the production of metadata is now vastly surpassing manual quality control capabilities."

Readability Measures

Readability, or more generally *discourse analysis*, is another fertile area of application for semantic analysis. Measures of readability and style are used when classifying and evaluating learning resources. They have been around for a long time. The best-known measure is the Flesch-Kincaid grade level measure developed for the U.S. Navy in 1975. It is given by

*0.39 * (Average words per sentence) +*
*11.80 * (Average syllables per word) - 0.59*

In 1975 methods such as LSA were unknown and computationally infeasible, but since then researchers such as (Si & Callan, 2001) have developed more sophisticated readability measures. Systems such as the Coh-metrix system described in (Graesser & McNamara, 2011) apply NLP and semantic analysis to produce multiple measures including narrativity, syntactic simplicity, word correctness, referential cohesion and deep cohesion. These provide a deeper understanding of the level and difficulty of a text.

Unfortunately, most publishers still use formulaic measures such as Flesch-Kincaid. However, according to the researchers at the University of Memphis, they are starting to show interest in more sophisticated measures computed using semantic analysis. We anticipate that such measures will play an increasingly important role in the future.

Search using Semantic Similarity

Search engines can apply semantic analysis in many ways. The U.S. Army Reimer Digital Library provides an example. We used a corpus of Army training manuals to generate an LSA space and deployed the resulting similarity measure in the Reimer Library in 2008. Reimer ranked search results by similarity to the search term. The LSA measure also identified other terms related to the original search term. The Reimer interface displayed these in a "tag cloud." For example, a user searching for "vehicle" might see the terms "truck," "car," and "engine." Clicking on one of these would launch a new search.

The authors have also implemented a different type of semantic search as part of a National Science Foundation project called *Ubiquitous Contextualized Access to STEM Educational Resources*, or UCASTER (Eduworks, 2012). UCASTER takes any document as input and retrieves the most relevant resources from a user-designated set of digital libraries. Relevance is based on semantic similarity to the source document. UCASTER searches use an entire document as the search query rather than a few search terms. In our experience this method is superior for retrieving relevant resources from digital libraries.

Semi-automated Corpus Generation

Constructing TF*IDF measures and LSA spaces requires a domain corpus. To broadly apply semantic analytic methods, we need a process for rapidly and inexpensively generating domain corpora. We have recently applied UCASTER for this purpose. For this purpose we start with a few representative documents selected by an expert. We then use a web service version of UCASTER to automatically retrieve documents related to these "seed" documents, thereby converting them into a large domain corpus drawn from multiple sources.

To test this method we created a corpus of about 100 source documents by manually searching public sites for content related to ten categories of common core standards for elementary school mathematics (www.corestandards.org/). We used UCASTER to compare the aggregate of these documents to every

paragraph in every document in the SMILE digital library (http://howtosmile.org/). The SMILE digital library contained 2096 harvestable resources of which 972 were officially categorized as "mathematics." SMILE resources were ranked by their highest paragraph score. Figure 3 shows the precision among the results in various ranges of rankings.

	Ranks				
	1 - 5	**1 - 25**	**26 - 100**	**101 - 200**	**201 - 300**
Precision	100%	84%	80%	63%	67%

Figure 3: Precision of SMILE searches at various ranges

These (and similar) results suggests that using seed documents and UCASTER-style semantic search is a viable technique for generating domain corpora.

TOPIC DETECTION

Competency alignment uses semantic analysis to match documents to a set of competencies. A more general version of this problem is *Topic detection.* Topic detection is a basic tool for evaluating student responses in adaptive learning environments such as intelligent tutor systems.

The goal of topic detection is determine the subject matter of a document by analyzing the text in the document. Topic detection can be less constrained than competency or standards alignment because the universe of possible topics may not be known in advance. In that case methods that compare documents to predetermined list are not applicable.

Modern topic detection algorithms use probabilistic methods. These methods, most notably Probabilistic LSA (PLSA) (Hoffmann, 2001) and Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003), take LSA one step further by explicitly modeling a document as a mix of concepts. In the probabilistic model, each concept randomly generates terms according to a probability distribution. For example, the topic of "currency" might generate "dollar" with probability 0.1, "paper" with probability 0.03, "exchange" with probability 0.07, and so on. LDA assumes a Dirichlet distribution and uses Expectation Maximization (EM) to determine both the concepts and their mixture in a document (Blei et al., 2003; Mariote, Medeiros, & da Torres, 2007).

EXPLICIT SEMANTIC ANALYSIS

IDF weights and LSA spaces usually come from corpora of documents related to a specific domain. The resulting similarity measures may perform poorly in other domains. This is problematic for general applications (such as topic detection) where the domain is not known in advance.

For topic selection, one solution is to find a universal corpus already annotated with topics. The obvious one is Wikipedia.™ At the time of this writing, the English version of Wikipedia contained almost four million articles and 27 million pages (Wikipedia, 2012).

Researchers increasingly use Wikipedia for word sense disambiguation, topic detection and other semantic analysis tasks (Gabrilovich & Markovitch, 2007), (Mihalcea, 2007), (Potthast, Stein, & Anderka, 2008), (Huang, Milne, Frank et al., 2009). They call this practice *explicit semantic analysis* (ESA). To determine the topics in a document, a typical ESA algorithm compares the document to all Wikipedia articles and selects the articles that are appear to be most similar. The titles of these articles serve as the topics in the document.

LEXICONS AND ONTOLOGIES

Semantic similarity measures give one way of finding related terms. A different way is to use dictionaries and *ontologies* to explicitly define relationships. Languages such as Resource Description Framework (RDF) and Web Ontology Language (OWL) are designed to encode relationships in a machine-actionable format. The vision of the "semantic web" (Lee, Hendler, & Lassila, 2001) is a world where computers can retrieve resources by their descriptions and contextualized meaning rather than by their web addresses.

The ontology used most often for semantic analysis is WordNet (Wordnet, 2012). WordNet contains 155,000 classes or *synsets*, each representing a specific meaning of a single term. As an example, synsets for "cat" include "cat, true cat," (the mammal), "guy, cat, hombre …" (informal term for a youth or man), "computerized tomography, CT, computerized axial tomography … CAT" (the medical technology) and "Caterpillar, cat" (the construction equipment).

WordNet includes both terms and relations among terms. WordNet identifies synonyms, meronyms (A is part of B), hyponyms (A is in the class of B), and antonyms (A is the opposite of B). In this way WordNet forms a hierarchical structure. By traversing this structure a computer can, for example, determine that a cat is a feline; felines are carnivores with paws; and dogs are related to cats since they are also carnivores with paws. Semantic algorithms use WordNet for word sense disambiguation and as a thesaurus, but its ontology structure can lead to more

powerful applications such as automated question generation. We discuss this next.

AUTOMATED QUESTION GENERATION

Writing assessment questions is one of the more important and time consuming tasks faced by designers and authors of training materials. It is therefore natural to seek ways to automate the process. Several researchers, including (Brown, Frishkoff, & Eskenazi, 2005), (Mitkov, An Ha, & Karamanis, 2006), (Heilman & Smith, 2009), have developed sophisticated systems that automatically generate questions from text.

The cited systems first perform a syntactical analysis that represents text as a tree labeled with parts of speech (Marcus, Marcinkiewicz, & Santorini, 1993). These trees also appear in algorithms that generate concept maps from textbooks (Person, Olney, D'Mello et al., 2012). Question generation algorithms isolate declarative statements and transform their corresponding trees into questions. The transformation rules can generate multiple types of questions from a single sentence. For example, the sentence "The bird flew away because it saw a cat" might produce "Why did the bird fly away?" and "What did the bird see?" and even "What do birds do?"

Questions can appear as open-ended questions or as multiple choice questions. Multiple choice questions require a correct answer and a set of incorrect answers (called *distractors*) that are plausible and represent potential misconceptions. Algorithms use WordNet to generate distractors by looking for terms that are semantically related to the correct answer in specific ways. For example, an algorithm generating a distractor for the question "What did the bird see?" might determine that "dog" is an animal similar to "cat" and use "a dog" as a distractor.

Automated or semi-automated question generation has practical advantages over manual generation of question banks. For example, (Mitkov & Ha, 2003) generated question banks using an automated method followed by human validation. They reported time savings of up to 74%. (Heilman & Smith, 2009) extended the methods of (Mitkov & Ha, 2003) in several ways. They applied more general transformation rules, used post-processing to eliminate grammatical errors, and used a probabilistic algorithm of (Collins & Koo, 2000) to rank the quality of questions. In tests where human evaluators examined questions for acceptability, they achieved 43% precision at rank 10 and generated an average of 6.8 acceptable questions per 250 words of text. Automated question generation remains an active area of research that has many applications to learning and training.

VIRTUAL ENVIRONMENTS

All of the applications of semantic analysis discussed in this paper apply to adaptive immersive learning environments, often in more essential and intriguing ways than in traditional eLearning. Projects such as the Generalized Intelligent Framework for Tutoring (GIFT) (Goldberg et al., 2011) envision systems that guide trainees through learning experiences based on the status of competencies, affective state, learning goals, and performance data. In these systems, semantic analysis is required to dynamically evaluate learner responses, detect the micro-adaptations that guide the learner experience, and to retrieve the right content for the system to display. Capabilities such as real-time automated grading are "nice to have" in settings where student responses can be manually graded, but are essential for evidence-based assessment models such as those discussed in (Shute, Masduki, & Donmez, 2010) and implemented in simulation and game-based training systems by researchers such as (Smith, DuBuc, & Marvin, 2007). Automated question generation can reduce the time needed to construct assessments used in virtual learning environments, just as it does for online question banks, but with improvement it could also enable intelligent agents to pose contextually meaningful questions without requiring the questions to be written in advance.

Figure 4 summarizes the potential applications of semantic analysis to immersive learning.

Requirement	Semantic Analysis Tools
Find content that matches a scenario or learning objective	• Search by semantic similarity • Automated alignment • Automated metadata generation
Construct concept maps / domain models	• Topic detection • Treebank analysis • Ontological analysis
Assess learner responses along multiple dimensions	• Semantic relevance measures • Discourse and affect analysis
Assess learning outcomes and competency	• Automated question generation • Automated assessment • Competency matching
Enable tutors and avatars to teach and coach effectively	• Automated question generation • Automated detection of affective and cognitive state

Figure 4: Applications of Semantic Analysis

CONCLUSION

We have provided an overview of semantic analysis and its applications to training, education and simulation. As demonstrated by several research groups, techniques from computational linguistics, natural language processing, and artificial intelligence (specifically, machine learning) are effective for evaluating student responses, improving search, generating assessment questions, and classifying content according to competencies. Computers are not perfect at these tasks but perform well and are as consistent as human evaluators. In this regard, semantic analysis holds significant promise for use in the architectures and immersive learning environments being researched today.

AKNOWLEDGEMENTS

Research reported here has been supported in part by the National Science Foundation (Award 1044161), the Advanced Distributed Learning initiative (BAA W91CRB-08-R-0073), and the Army Research Lab (W911QX-12-C-0055). We would also like to thank Dr. Xiangen Hu and members of the IITSEC review panel for reviewing and making useful comments and suggestions on this paper.

REFERENCES

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*(5), 993-1022.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). *Automatic question generation for vocabulary assessment*.

Collins, M., & Koo, T. (2000). *Discriminative reranking for natural language parsing*.

D'Mello, S., & Graesser, A. (2012). Emotions During the Learning of Difficult Material. *The Psychology of Learning and Motivation*, 183.

D'Mello, S. K., Craig, S. D., Witherspoon, A., Mcdaniel, B., & Graesser, A. (2008). Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction, 18*(1), 45-80.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science, 41*(6), 391-407.

Devaul, H., Diekema, A. R., & Ostwald, J. (2011). Computer-assisted assignment of educational standards using natural language processing. *Journal of the American Society for Information Science and Technology, 62*(2), 395-405.

Diekema, A. R., & Chen, J. (2005). *Experimenting with the automatic assignment of educational standards to digital library content.* Paper presented at the Joint Conference on Digital Libraries, Denver.

Eduworks. (2012). UCASTER Retrieved June 2012, from http://ucaster.eduworks.com

Foltz, P. W., Laham, D., & Landauer, T. K. (1999). The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1*(2).

Foltz, P. W., Landauer, T. K., Laham II, R. D., Kintsch, W., & Rehder, R. E. (2002). Methods for analysis and evaluation of the semantic content of a writing based on vector length: Google Patents.

Gabrilovich, E., & Markovitch, S. (2007). *Computing semantic relatedness using wikipedia-based explicit semantic analysis*.

Goldberg, B. S., Holden, H. K., Brawner, K. W., & Sottilare, R. A. (2011). *Enhancing Performance through Pedagogy and Feedback: Domain Considerations for ITSs*. Paper presented at the Interservice Interindustry Simulation Educacation and Training, Orlando, FL. https://litelab.arl.army.mil/system/files/IITSEC2011_Goldberg_etal_Enhancing%20Performance%20Through%20Pedagogy%20and%20Feedback.pdf

Graesser, A. C., Franklin, S., Wiemer-Hastings, P., & Group, T. R. (1998). *Simulating smooth tutorial dialogue with pedagogical value*.

Graesser, A. C., & McNamara, D. S. (2011). Computational analyses of multilevel discourse comprehension. *Topics in Cognitive Science, 3*(2), 371-398.

Greenberg, J. (2003). Metadata extraction and harvesting: a comparison of two automatic metadata generation applications. *Journal of Internet Cataloging, 6*(4), 59-82.

Heilman, M., & Smith, N. A. (2009). Question generation via overgenerating transformations and ranking: DTIC Document.

Hoffmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning, 42*, 177-196.

Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A. C., & McNamara, D. S. (2007). Strengths, limitations, and extensions of LSA. *The handbook of latent semantic analysis*, 401-426.

Hu, X., & Martindale, T. (2008). *Enhance learning with ITS style interactions between learner and content*.

Huang, A., Milne, D., Frank, E., & Witten, I. (2009). Clustering documents using a wikipedia-based concept representation. *Advances in Knowledge Discovery and Data Mining*, 628-636.

Hull, D. A., & Grefenstette, G. (1996). A detailed analysis of english stemming algorithms. *Rank Xerox Research Centre*.

IEEE. (2002). IEEE-1484.12.1-2002: Standard for Learning Object Metadata.

Jackson, J. (2011). IBM Watson Vanquishes Human Jeopardy Foes. *PCWorld, 2012*. Retrieved from

Kim, S. M., Valitutti, A., & Calvo, R. A. (2010). *Evaluation of unsupervised emotion models to textual affect recognition*.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review; Psychological Review, 104*(2), 211.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes, 25*(2-3), 259-284.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automatic essay assessment. *Assessment in education: Principles, policy & practice, 10*(3), 295-308.

Lee, T. B., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34-43.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics, 19*(2), 313-330.

Mariote, L., Medeiros, C., & da Torres, R. (2007). *Parallelized variational EM for latent Dirichlet allocation: An experimental evaluation of speed and scalability*.

Mihalcea, R. (2007). *Using wikipedia for automatic word sense disambiguation*.

Mitkov, R., An Ha, L., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering, 12*(2), 177-194.

Mitkov, R., & Ha, L. (2003). *Computer-Aided Generation of Multiple-Choice Tests.* Paper presented at the HLT-NAACL Workshop on Building Educational Applications Using Natural Language Processing, Edmonton, Canada.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR), 41*(2), 10.

NETg. (1998). NETg Announces Development of Precision Skilling Technology Retrieved June 11, 2012, from http://bit.ly/MolOPR

Ochoa, X., & Duval, E. (2009). Automatic evaluation of metadata quality in digital repositories. [Article]. *International Journal on Digital Libraries, 10*(2/3), 67-91.

Pearson. (2010). Intelligent Essay Assessor (IEA) Fact Sheet: Pearson Education.

Pérez, D., Gliozzo, A., Strapparava, C., Alfonseca, E., Rodríguez, P., & Magnini, B. (2005). *Automatic assessment of students' free-text answers underpinned by the combination of a bleu-inspired algorithm and latent semantic analysis*.

Person, N., Olney, A., D'Mello, S., & Lehman, B. (2012). *Interactive Concept Maps and Learning Outcomes in Guru*.

Pincombe, B. (2004). Comparison of human and latent semantic analysis (LSA) judgements of pairwise document similarities for a news corpus: DTIC Document.

Potthast, M., Stein, B., & Anderka, M. (2008). A wikipedia-based multilingual retrieval model. *Advances in Information Retrieval*, 522-530.

Reitsma, R., Marshall, B., Dalton, M., & Cyr, M. (2008). *Exploring educational standard alignment: in search of'relevance'*.

Shute, V. J., Masduki, I., & Donmez, O. (2010). Conceptual Framework for Modeling, Assessing and Supporting Competencies within Game Environments. *Technology, Instruction, Cognition, and Learning, 8*(2), 137-161.

Si, L., & Callan, J. (2001). *A statistical model for scientific readability*.

Smith, B., DuBuc, C., & Marvin, D. (2007). *Learner Assessment Data Models for standardizing assessment across Live, Virtual and Constructive Domains.* Paper presented at the Interservice/Industry Training, Simulation, and Education Conference

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). Introduction to data mining (1st ed.). Boston: Pearson Addison Wesley.

Wiemer-Hastings, & Graesser, A. (1999). *Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis*.

Wikipedia. (2012). Wikipedia: Size of Wikipedia. *2012*(June 17, 2012). Retrieved from

Wordnet. (2012). Wordnet: A lexical database for English., from http://wordnet.princeton.edu/

Zhang, Z., Iria, J., Brewster, C. A., & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms.