NLP Techniques in Intelligent Tutoring Systems

Chutima Boonthum
Hampton University, USA

Irwin B. Levinstein
Old Dominion University, USA

Danielle S. McNamara
The University of Memphis, USA

Joseph P. Magliano
Northern Illinois University, USA

Keith K. Millis
The University of Memphis, USA

INTRODUCTION

Many Intelligent Tutoring Systems (ITSs) aim to help students become better readers. The computational challenges involved are (1) to assess the students' natural language inputs and (2) to provide appropriate feedback and guide students through the ITS curriculum. To overcome both challenges, the following non-structural Natural Language Processing (NLP) techniques have been explored and the first two are already in use: word-matching (WM), latent semantic analysis (LSA, Landauer, Foltz, & Laham, 1998), and topic models (TM, Steyvers & Griffiths, 2007).

This article describes these NLP techniques, the iSTART (Strategy Trainer for Active Reading and Thinking, McNamara, Levinstein, & Boonthum, 2004) intelligent tutor and the related Reading Strategies Assessment Tool (R-SAT, Magliano *et al.*, 2006), and how these NLP techniques can be used in assessing students' input in iSTART and R-SAT. This article also discusses other related NLP techniques which are used in other applications and may be of use in the assessment tools or intelligent tutoring systems.

BACKGROUND

Interpreting text is critical for intelligent tutoring systems (ITSs) that are designed to interact meaningfully with, and adapt to, the users' input. Different ITSs use different Natural Language Processing (NLP) techniques in their system. NLP systems may be structural, *i.e.,* focused on grammar and logic, or non-structural, *i.e.,* focused on words and statistics. This article deals with the latter.

Examples of the structural approach include ExtrAns (Extracting Answers from technical texts question-answering system; Molla *et al*., 2003) which uses minimal logical forms (MLF; that is, the form of first order predicates) to represent both texts and questions and C-Rater (Leacock & Chodorow, 2003) which scores short-answer questions by analyzing the conceptual information of an answer in respect to the given question. Turning to the non-structural approach, AutoTutor (Graesser *et al*., 2000) uses LSA to analyze the student's input against expected sets of answers and CIRCSIM-Tutor (Kim *et al.*, 1989) uses a word-matching technique to evaluate students' short answers. The systems considered more fully below, iSTART (McNamara *et al.*, 2004) and R-SAT (Magliano *et al.*, 2006) use both word-matching and LSA in assessing quality of students' self-explanation. Topic models (TM) were explored in both systems, but have not yet been integrated.

MAIN FOCUS OF THE CHAPTER

This article presents three non-structural NLP techniques (WM, LSA, and TM) which are currently used or being explored in reading strategies assessment and training applications, particularly, iSTART and R-SAT.

Word Matching

Word matching is a simple and intuitive way to estimate the nature of an explanation. There are two ways to compare words from the reader's input (either answers or explanations) against benchmarks (collections of words that represent a unit of text or an ideal answer): (1) Literal word matching and (2) Soundex matching.

Literal word matching – Words are compared character by character and if there is a match of sufficient length then we call this a *literal match*. An alternative is to count words that have the same stem (*e.g.,* indexer and indexing) as matching. If a word is short a complete match may be required to reduce the number of false-positives.

Soundex matching – This algorithm compensates for misspellings by mapping similar characters to the same soundex symbol (Christian, 1998). Words are transformed to their soundex code by retaining the first character, dropping the vowels, and then converting other characters into soundex symbols: *1* for *b, p; 2* for *f, v; 3* for *c, k, s; etc.* Sometimes only one consecutive occurrence of the same symbol is retained. There are many variants of this algorithm designed to reduce the number of false positives (*e.g.,* Philips, 1990). As in literal matching, short words may require a full soundex match while for longer words the first *n* soundex symbols may suffice.

Word-matching is also used in other applications, such as, CIRCSIM-Tutor (Kim *et al.*, 1989) on short-answer questions and Short Essay Grading System (Ventura *et al.*, 2004) on questions with ideal expert answers.

Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA; Landauer, Foltz, & Laham, 1998) uses statistical computation to extract and represent the meaning of words. Meanings are represented in terms of their similarity to other words in a large corpus of documents. LSA begins by finding the frequency of terms used and the number of co-occurrences in each document throughout the corpus and then uses a powerful mathematical transformation to find deeper meanings and relations between words. When measuring the similarity between text-objects, LSA's accuracy improves with the size of the objects, so it provides the most benefit in finding similarity between two documents but as it does not take word order into account, short documents may not receive the full benefit. The details for constructing an LSA corpus matrix are in Landauer & Dumais (1997). Briefly, the steps are: (1) select a corpus; (2) create a term-document-frequency (TDF) matrix; (3) apply Singular Value Decomposition (SVD; Press *et al.*, 1986) to the TDF matrix to decompose it into three matrices (L x S x R; where S is a scaling, matrix). The leftmost matrix (L) becomes the LSA matrix of that corpus. The optimal size is usually in the range of 300–400 dimensions. Hence, the LSA matrix dimensions become N x D where N is the number of unique words in the entire corpus and D is the optimal dimension (reduced from the total number of documents in the entire corpus).

The similarity of terms (or words) is computed by comparing two rows, each representing a term vector. This is done by taking the cosine of the two term vectors. To find the similarity of sentences or documents, (1) for each document, create a document vector using the sum of the term vectors of all the terms appearing in the document and (2) calculate a cosine between two document vectors. Cosine values range from ±1 where +1 means highly similar.

To use LSA in the tutoring systems, a set of benchmarks are created and compared with the trainee's input. Examples benchmarks are the current target sentence, previous sentences, and the ideal answer. A high cosine value between the current sentence benchmark and the reader's input would indicate that the reader understood the sentence and was able to paraphrase what was read. To provide appropriate feedback, a number of cosines are computed (one for each benchmark). Various statistical methods, such as discriminant analysis and regression analysis, are used to construct the feedback formula. McNamara *et al.* (2007) describe various ways that LSA can be used to evaluate the reader's explanations: either LSA alone or a combination of LSA with WM. The final conclusion is that a fully-automated (*i.e.*, less hand-crafted benchmarks construction), combined system produces the better results.

There are a number of other intelligent tutoring systems that use LSA in their feedback system, for examples, Summary Street (Steinhart, 2001), Auto-

Related Content