Enhanced Learning with ITS Style Interactions Between Learner and Content

Unclassified

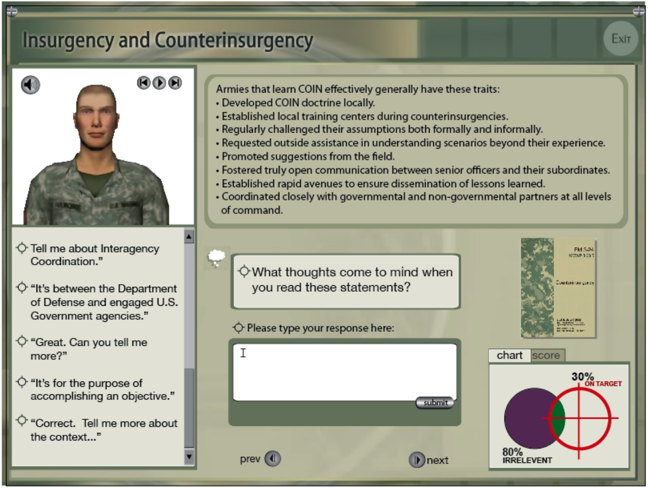
Xiangen Hu & Trey Martindale

Workforce Advanced Distributed Learning (ADL) Co-Lab

Outline

Outline

Example ITS interface



Typical ITS with natural language interaction

Tutoring Scenario Considered

Consider the scenario of ITS environment:

- Tutor asks a seed question with answer key ready
- Student gives a sequence of responses
- Tutor provide feedback, based on comparison of the student's response and the answer

Dialog structure

Typical structure

Tutor: Ask a seed question
Learner: Response 1
Tutor: Feedback 1
Learner: Response 2
Tutor: Feedback 2
...

Human tutor & ITS (Graesser & Person, 1994; Wenger, 1987)

- Learner *construct* answers
- Tutor give categorical responses (positive, neutral, or negative feedback).

Dialog structure

Typical structure

Tutor: Ask a seed question
Learner: Response 1
Tutor: Feedback 1
Learner: Response 2
Tutor: Feedback 2
...

Human tutor & ITS (Graesser & Person, 1994; Wenger, 1987)

- Learner *construct* answers
- Tutor give categorical responses (positive, neutral, or negative feedback).

Student Model and Model-Tracing

Student Model and Model-Tracing in ITS (Holt, Dubs, Jones & Greer, 1994; Wenger, 1987)

- Basic assumption (model) about learner's knowledge in a given context
- Student's action is evaluated against the model
- Feedback is given based on the comparison

Goal of the presentation

- Represent student's knowledge
- Simulation
- Initial application
- Some challenges
- Potentials
- Current projects

Student Model and Model-Tracing

Student Model and Model-Tracing in ITS (Holt et al., 1994; Wenger, 1987)

- Basic assumption (model) about learner's knowledge in a given context
- Student's action is evaluated against the model
- Feedback is given based on the comparison

Goal of the presentation

- Represent student's knowledge
- Simulation
- Initial application
- Some challenges
- Potentials
- Current projects

Outline

Analysis of student's contributions

Analysis a sequence of student's contribution (Hu, Cai, Wiemer-Hasting, Graesser & McNamara, 2007)

- Quality of contributions: Relevant or irrelevant to the answer key?
- New information: Adding new information?

Student model elements

	Old	New
Relevant	O-R	N-R
irrelevant	O-IR	N-IR

Table: Decomposition of student's contribution sequence. N-R: New & relevant; N-IR: New & Irrelevant; O-R: Old & Relevant; and O-IR: Old & Irrelevant.

Analysis of student's contributions

Analysis a sequence of student's contribution (Hu et al., 2007)

- Quality of contributions: Relevant or irrelevant to the answer key?
- New information: Adding new information?

Student model elements

	Old	New
Relevant	O-R	N-R
irrelevant	O-IR	N-IR

Table: Decomposition of student's contribution sequence. N-R: New & relevant; N-IR: New & Irrelevant; O-R: Old & Relevant; and O-IR: Old & Irrelevant.

Learner's Characteristics Curves

A set of four curves with a common variable: Student's contribution in sequence: 1, 2, 3, ...

Possible values

1. Relevant & New
2. Relevant & Old
3. Irrelevant & New
4. Irrelevant & Old

Learner's Characteristics Curves

A set of four curves with a common variable: Student's contribution in sequence: 1, 2, 3, ...

Possible values

1. Relevant & New
2. Relevant & Old
3. Irrelevant & New
4. Irrelevant & Old

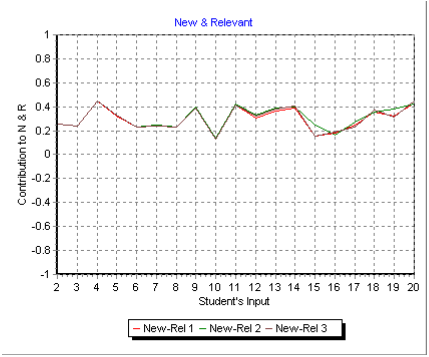


Figure: Example of LCC: R&N

Outline

Vector representation

Semantic spaces (Hu, Cai, Graesser & Ventura, 2005)

assumption 1: Semantic of a language entity (word, phrase, sentence, ...) can be represented as a finite dimension vector

assumption 2: Similarity between two language entities is a function of the two vectors.

Example semantic space: LSA (Landauer, Laham & Foltz, 1998)

- Obtained by SVD of word-document matrix of collection of written text
- Semantics of terms, phrases, sentences, and paragraphs are numerical vectors
- Widely used in information retrieval, text analysis, and text similarity analysis

Basic assumptions

Vector representation

Semantic spaces (Hu et al., 2005)

assumption 1: Semantic of a language entity (word, phrase, sentence, ...) can be represented as a finite dimension vector

assumption 2: Similarity between two language entities is a function of the two vectors.

Example semantic space: LSA (Landauer et al., 1998)

- Obtained by SVD of word-document matrix of collection of written text
- Semantics of terms, phrases, sentences, and paragraphs are numerical vectors
- Widely used in information retrieval, text analysis, and text similarity analysis

Model Setup

Notations

Answer Key: $\mathbf{K}$, a semantic vector

Student's response: $\mathbf{R}_n$, $n = 1, ..., N$, sequence of semantic vectors

Response History: $\mathbf{H}_m$, a function of $(\mathbf{R}_1, ..., \mathbf{R}_{m-1})$, $m = 1, ..., N$. $\mathbf{H}_0$ is a zero vector.

Basic assumptions of LCC

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *similarities* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *similarities* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *similarities* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *similarities* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)

Model Setup

Notations

Answer Key: $\mathbf{K}$, a semantic vector

Student's response: $\mathbf{R}_n$, $n = 1, ..., N$, sequence of semantic vectors

Response History: $\mathbf{H}_m$, a function of $(\mathbf{R}_1, ..., \mathbf{R}_{m-1})$, $m = 1, ..., N$. $\mathbf{H}_0$ is a zero vector.

Basic assumptions of LCC

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *similarities* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *similarities* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *similarities* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *similarities* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic to *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{K}$) and *dissimilarity* between ($\mathbf{R}_n$ and $\mathbf{H}_n$)

Model Setup

Definitions of *semantic similarity* in semantic space (Hu et al., 2005)

Let $S(\mathbf{V}_1, \mathbf{V}_2)$ be the similarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$

- $S(\mathbf{V}_1, \mathbf{V}_2) \in (-M, +M)$
- $S(\mathbf{V}_1, \mathbf{V}_2) = M$, if $\mathbf{V}_1 = \mathbf{V}_2 \neq \mathbf{0}$
- $S(\mathbf{V}_1, \mathbf{V}_2) = 0$, if $\mathbf{V}_2 = \mathbf{0}$ or $\mathbf{V}_2 = \mathbf{0}$

(proposed) Definition of *dissimilarity*: Never explicitly defined in semantic spaces!

Let $D(\mathbf{V}_1, \mathbf{V}_2)$ be the dissimilarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$

- $D(\mathbf{V}_1, \mathbf{V}_2) \in (-M, +M)$
- $D(\mathbf{V}_1, \mathbf{V}_2) = 0$, if $\mathbf{V}_1 = \mathbf{V}_2$
- $D(\mathbf{V}_1, \mathbf{V}_2) = M$, if $\mathbf{V}_2 = \mathbf{0}$ or $\mathbf{V}_2 = \mathbf{0}$

Model Setup

Definitions of *semantic similarity* in semantic space (Hu et al., 2005)

Let $S(\mathbf{V}_1, \mathbf{V}_2)$ be the similarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$

- $S(\mathbf{V}_1, \mathbf{V}_2) \in (-M, +M)$
- $S(\mathbf{V}_1, \mathbf{V}_2) = M$, if $\mathbf{V}_1 = \mathbf{V}_2 \neq \mathbf{0}$
- $S(\mathbf{V}_1, \mathbf{V}_2) = 0$, if $\mathbf{V}_2 = \mathbf{0}$ or $\mathbf{V}_2 = \mathbf{0}$

(proposed) Definition of *dissimilarity*: Never explicitly defined in semantic spaces!

Let $D(\mathbf{V}_1, \mathbf{V}_2)$ be the dissimilarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$

- $D(\mathbf{V}_1, \mathbf{V}_2) \in (-M, +M)$
- $D(\mathbf{V}_1, \mathbf{V}_2) = 0$, if $\mathbf{V}_1 = \mathbf{V}_2$
- $D(\mathbf{V}_1, \mathbf{V}_2) = M$, if $\mathbf{V}_2 = \mathbf{0}$ or $\mathbf{V}_2 = \mathbf{0}$

Outline

Definitions

Definitions of *semantic similarity* in LSA

Let $S(\mathbf{V}_1, \mathbf{V}_2)$ be the similarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$ in LSA space, $S(\mathbf{V}_1, \mathbf{V}_2)$ is the projection of the unit vector $\mathbf{V}_1/\|\mathbf{V}_1\|$ on $\mathbf{V}_2/\|\mathbf{V}_2\|$. It is also called *cosine match*.

(proposed) Definition of *dissimilarity* in LSA

Let $D(\mathbf{V}_1, \mathbf{V}_2)$ be the *dissimilarity* measure between $\mathbf{V}_1$ and $\mathbf{V}_2$ in LSA space, $D(\mathbf{V}_1, \mathbf{V}_2)$ is the projection of the unit vector $\mathbf{V}_1/\|\mathbf{V}_1\|$ on $\mathbf{V}_2^{\perp}/\|\mathbf{V}_2^{\perp}\|$. It can be called *sine match*.

Definitions

Definitions of *semantic similarity* in LSA

Let $S(\mathbf{V}_1, \mathbf{V}_2)$ be the similarity measure between $\mathbf{V}_1$ and $\mathbf{V}_2$ in LSA space, $S(\mathbf{V}_1, \mathbf{V}_2)$ is the projection of the unit vector $\mathbf{V}_1/\|\mathbf{V}_1\|$ on $\mathbf{V}_2/\|\mathbf{V}_2\|$. It is also called *cosine match*.

(proposed) Definition of *dissimilarity* in LSA

Let $D(\mathbf{V}_1, \mathbf{V}_2)$ be the *dissimilarity* measure between $\mathbf{V}_1$ and $\mathbf{V}_2$ in LSA space, $D(\mathbf{V}_1, \mathbf{V}_2)$ is the projection of the unit vector $\mathbf{V}_1/\|\mathbf{V}_1\|$ on $\mathbf{V}_2^{\perp}/\|\mathbf{V}_2^{\perp}\|$. It can be called *sine match*.

LCC in LSA

LCC Definitions

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic function of $S(\mathbf{R}_n, \mathbf{K})$ and $S(\mathbf{R}_n, \mathbf{H}_n)$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic function of $S(\mathbf{R}_n, \mathbf{K})$ and $D(\mathbf{R}_n, \mathbf{H}_n)$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic function of $D(\mathbf{R}_n, \mathbf{K})$ and $S(\mathbf{R}_n, \mathbf{H}_n)$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: Monotonic function of $D(\mathbf{R}_n, \mathbf{K})$ and $D(\mathbf{R}_n, \mathbf{H}_n)$

Algorithm 1

Algorithms

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nk_1} + \mathbf{R}_{nk_2}$. $\mathbf{R}_{nk_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nk_2}$ orthogonal to $\mathbf{K}$

Step 2: Decompose $\mathbf{R}_n = \mathbf{R}_{nh_2} + \mathbf{R}_{nh_2}$. $\mathbf{R}_{nh_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nh_2}$ orthogonal to $\mathbf{H}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_1}, \mathbf{R}_{nh_2})$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_1}, \mathbf{R}_{nh_2})$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_2}, \mathbf{R}_{nh_2})$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_2}, \mathbf{R}_{nh_2})$

Algorithm 1

Algorithms

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nk_1} + \mathbf{R}_{nk_2}$. $\mathbf{R}_{nk_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nk_2}$ orthogonal to $\mathbf{K}$

Step 2: Decompose $\mathbf{R}_n = \mathbf{R}_{nh_2} + \mathbf{R}_{nh_2}$. $\mathbf{R}_{nh_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nh_2}$ orthogonal to $\mathbf{H}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_1}, \mathbf{R}_{nh_2})$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_1}, \mathbf{R}_{nh_2})$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_2}, \mathbf{R}_{nh_2})$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2}\|\|\mathbf{R}_{nh_2}\|S(\mathbf{R}_{nk_2}, \mathbf{R}_{nh_2})$

Algorithm 2

Algorithms

$$\mathbf{R}_n = (\mathbf{R}_{nk_1h_1} + \mathbf{R}_{nk_1h_2}) + (\mathbf{R}_{nk_2h_1} + \mathbf{R}_{nk_2h_2})$$

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nk_1} + \mathbf{R}_{nk_2}$. $\mathbf{R}_{nk_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nk_2}$ orthogonal to $\mathbf{K}$

Step 2: Decompose $\mathbf{R}_{nk_1} = \mathbf{R}_{nk_1h_1} + \mathbf{R}_{nk_1h_2}$. $\mathbf{R}_{nk_1h_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nk_1h_2}$ orthogonal to $\mathbf{H}$

Step 3: Decompose $\mathbf{R}_{nk_2} = \mathbf{R}_{nk_2h_1} + \mathbf{R}_{nk_2h_2}$. $\mathbf{R}_{nk_2h_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nk_2h_2}$ orthogonal to $\mathbf{H}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $||\mathbf{R}_{nk_1h_1}||$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $||\mathbf{R}_{nk_1h_2}||$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $||\mathbf{R}_{nk_2h_1}||$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $||\mathbf{R}_{nk_2h_2}||$

Algorithm 2

Algorithms

$$\mathbf{R}_n = (\mathbf{R}_{nk_1h_1} + \mathbf{R}_{nk_1h_2}) + (\mathbf{R}_{nk_2h_1} + \mathbf{R}_{nk_2h_2})$$

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nk_1} + \mathbf{R}_{nk_2}$. $\mathbf{R}_{nk_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nk_2}$ orthogonal to $\mathbf{K}$

Step 2: Decompose $\mathbf{R}_{nk_1} = \mathbf{R}_{nk_1h_1} + \mathbf{R}_{nk_1h_2}$. $\mathbf{R}_{nk_1h_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nk_1h_2}$ orthogonal to $\mathbf{H}$

Step 3: Decompose $\mathbf{R}_{nk_2} = \mathbf{R}_{nk_2h_1} + \mathbf{R}_{nk_2h_2}$. $\mathbf{R}_{nk_2h_2}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nk_2h_2}$ orthogonal to $\mathbf{H}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1h_1}\|$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_1h_2}\|$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2h_1}\|$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nk_2h_2}\|$

Algorithm 3

Algorithms

$$\mathbf{R}_n = (\mathbf{R}_{nh_1k_1} + \mathbf{R}_{nh_1k_2}) + (\mathbf{R}_{nh_2k_1} + \mathbf{R}_{nh_2k_2})$$

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nh_1} + \mathbf{R}_{nh_2}$. $\mathbf{R}_{nh_1}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nh_2}$ orthogonal to $\mathbf{H}$

Step 2: Decompose $\mathbf{R}_{nh_1} = \mathbf{R}_{nh_1k_1} + \mathbf{R}_{nh_1k_2}$. $\mathbf{R}_{nh_1k_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nh_1k_2}$ orthogonal to $\mathbf{K}$

Step 3: Decompose $\mathbf{R}_{nh_2} = \mathbf{R}_{nh_2k_1} + \mathbf{R}_{nh_2k_2}$. $\mathbf{R}_{nh_2k_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nh_2k_2}$ orthogonal to $\mathbf{K}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_1k_1}\|$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_2k_1}\|$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_1k_2}\|$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_2k_2}\|$

Algorithm 3

Algorithms

$$\mathbf{R}_n = (\mathbf{R}_{nh_1k_1} + \mathbf{R}_{nh_1k_2}) + (\mathbf{R}_{nh_2k_1} + \mathbf{R}_{nh_2k_2})$$

Step 1: Decompose $\mathbf{R}_n = \mathbf{R}_{nh_1} + \mathbf{R}_{nh_2}$. $\mathbf{R}_{nh_1}$ parallel to $\mathbf{H}$ and $\mathbf{R}_{nh_2}$ orthogonal to $\mathbf{H}$

Step 2: Decompose $\mathbf{R}_{nh_1} = \mathbf{R}_{nh_1k_1} + \mathbf{R}_{nh_1k_2}$. $\mathbf{R}_{nh_1k_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nh_1k_2}$ orthogonal to $\mathbf{K}$

Step 3: Decompose $\mathbf{R}_{nh_2} = \mathbf{R}_{nh_2k_1} + \mathbf{R}_{nh_2k_2}$. $\mathbf{R}_{nh_2k_1}$ parallel to $\mathbf{K}$ and $\mathbf{R}_{nh_2k_2}$ orthogonal to $\mathbf{K}$

Algorithms

- $F_{RO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_1k_1}\|$
- $F_{RN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_2k_1}\|$
- $F_{IO}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_1k_2}\|$
- $F_{IN}(\mathbf{R}_n, \mathbf{K}, \mathbf{H}_n)$: $\|\mathbf{R}_{nh_2k_2}\|$

Simulation

simulation

# of terms in simulation: 1000
# of dimensions: 300
# of terms in **K**: 40
# of terms in **R**: 10
# of responses in **H**: 2
p is the proportion of **R** in **K**

Simulation

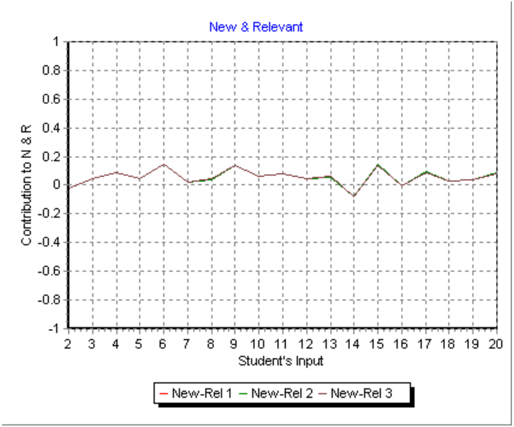


Figure: New and Relevant with $p = 0.0$

Simulation

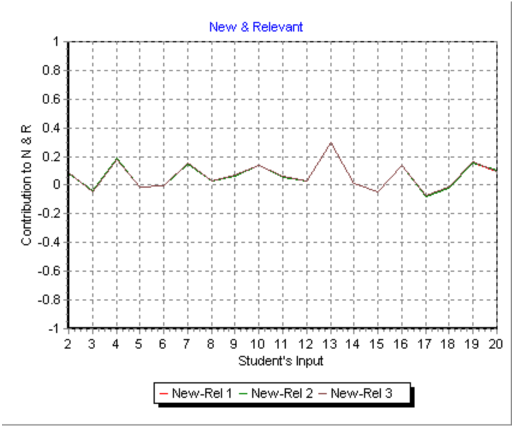


Figure: New and Relevant with $p = 0.1$

Simulation

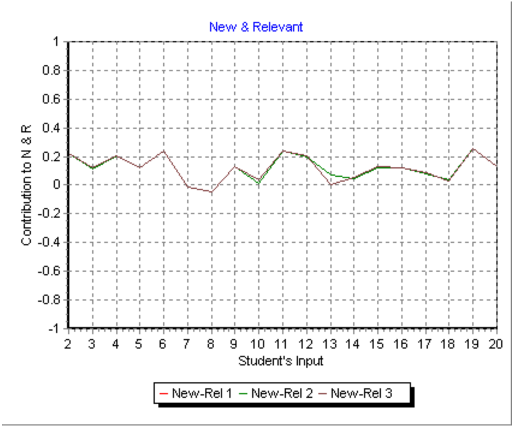


Figure: New and Relevant with $p = 0.2$

Simulation

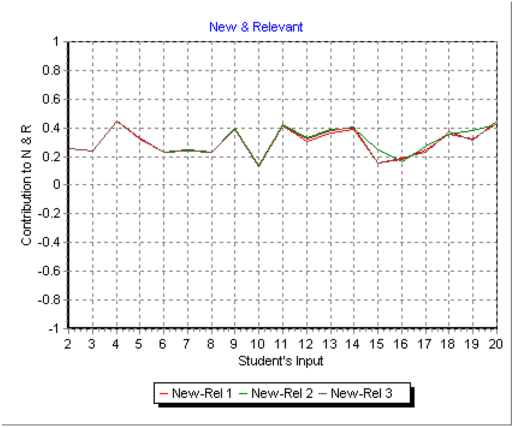


Figure: New and Relevant with $p = 0.5$

Simulation

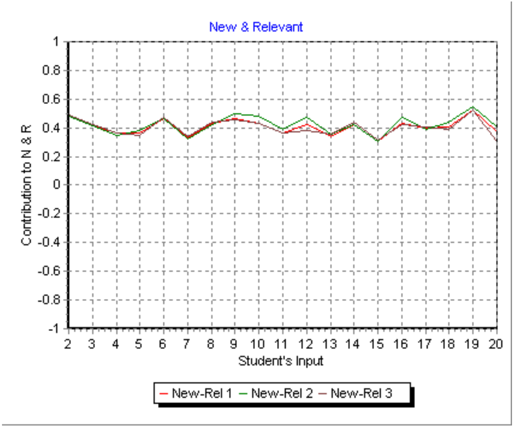


Figure: New and Relevant with $p = 0.9$

Outline

Applying LCC in ITS

Select a framework

- Select semantic engine (such as LSA)
- Encode domain knowledge

Simulate prototypical learner (LCC)

- Determine context parameters
  - encode answer key
  - size of history considered
- Simulation parameters (variables in simulations)
  - size of student's response
  - student's knowledge (use proportion of response with answer key)

Using LCC

- Simulated LCC (with parameters) are used as student model in an ITS application.
- LCC used as basic mechanism for ITS to chose feedback (neutral, positive, or negative)

Applying LCC in ITS

Select a framework

- Select semantic engine (such as LSA)
- Encode domain knowledge

Simulate prototypical learner (LCC)

- Determine context parameters
  - encode answer key
  - size of history considered
- Simulation parameters (variables in simulations)
  - size of student's response
  - student's knowledge (use proportion of response with answer key)

Using LCC

- Simulated LCC (with parameters) are used as student model in an ITS application.
- LCC used as basic mechanism for ITS to chose feedback (neutral, positive, or negative)

Applying LCC in ITS

Select a framework

- Select semantic engine (such as LSA)
- Encode domain knowledge

Simulate prototypical learner (LCC)

- Determine context parameters
  - encode answer key
  - size of history considered
- Simulation parameters (variables in simulations)
  - size of student's response
  - student's knowledge (use proportion of response with answer key)

Using LCC

- Simulated LCC (with parameters) are used as student model in an ITS application.
- LCC used as basic mechanism for ITS to chose feedback (neutral, positive, or negative)

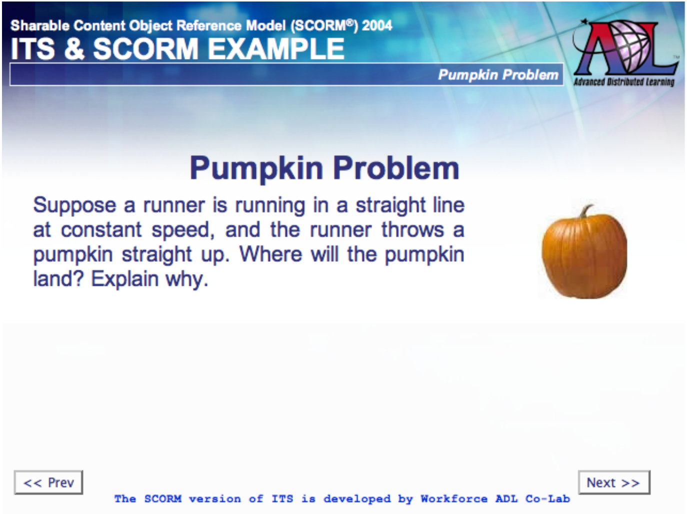


Figure: Newtonian Physics

Newtonian Physics

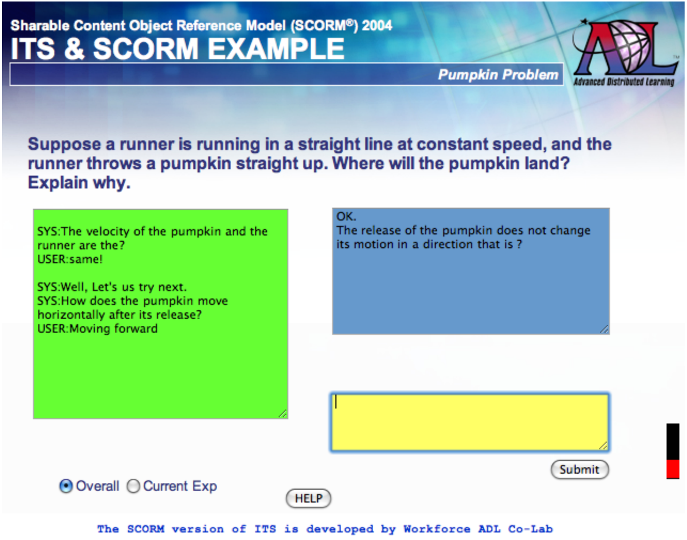


Figure: Newtonian Physics

Counterinsurgency

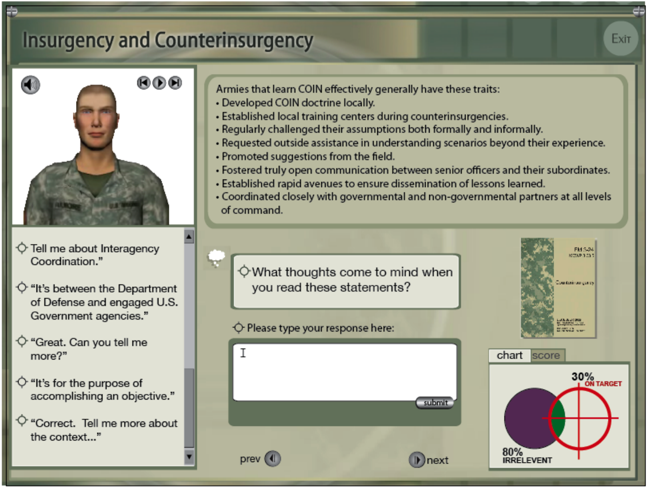


Figure: Counterinsurgency

Outline

Summary

About LCC

- A mathematical model is used to characterize student's learning behavior
- Basic behavior of LCC simulated
- Simple cases of LCC applied in eLearning (ITS)
- Simple properties of LCC explored (by simulation)

Apply cognitive models in eLearning

- LSA is just a easy example to use. Other models (such as global memory models) can be used to characterize learner's behavior
- LCC, as presented and applied in eLearning, is a computational model of sequential recall.
- Expect to use the same methodology to explore behaviors of free recall where subjects retrieve relevant information with given instructions/directions.

Summary

About LCC

- A mathematical model is used to characterize student's learning behavior
- Basic behavior of LCC simulated
- Simple cases of LCC applied in eLearning (ITS)
- Simple properties of LCC explored (by simulation)

Apply cognitive models in eLearning

- LSA is just a easy example to use. Other models (such as global memory models) can be used to characterize learner's behavior
- LCC, as presented and applied in eLearning, is a computational model of sequential recall.
- Expect to use the same methodology to explore behaviors of free recall where subjects retrieve relevant information with given instructions/directions.

Summary

Research questions

- Selection of knowledge representation framework
- With given knowledge representation framework, what would be the characteristics of LCC
  - LSA, HAL, or other vector semantic representations
- Applicable domains
- impact of feedback

Current Projects

Intelligent Delivery of Sharable Content Object (SCO)

- Automatic processing of content information: Extract the textual information, semantic encoding
- Simulate LCC represent typical users (with variable amount of knowledge and response style)
- Use LCC as student model and build Model-Tracing ITS
- Change static SCO into ITS style interactive knowledge object (KO)

Summary

Research questions

- Selection of knowledge representation framework
- With given knowledge representation framework, what would be the characteristics of LCC
  - LSA, HAL, or other vector semantic representations
- Applicable domains
- impact of feedback

Current Projects

Intelligent Delivery of Sharable Content Object (SCO)

- Automatic processing of content information: Extract the textual information, semantic encoding
- Simulate LCC represent typical users (with variable amount of knowledge and response style)
- Use LCC as student model and build Model-Tracing ITS
- Change static SCO into ITS style interactive knowledge object (KO)

Questions