Quesada, J.F, Kintsch, W. and Gomez, E. (2001) A Computational Theory of Complex Problem Solving Using the Vector Space Model (part I): Latent Semantic Analysis, Through the Path of Thousands of Ants. In Cañas (Ed.) Cognitive resarch with Microworlds, pp. 117-131

A Computational Theory of Complex Problem Solving Using the Vector Space Model (part I): Latent Semantic Analysis, Through the Path of Thousands of Ants.

José Quesada, Walter Kintsch

Institute of Cognitive Science
University of Colorado, Boulder
Muenzinger psychology building
Campus Box 344
Boulder, CO 80309-0344
{quesadaj, wkintsch}@psych.colorado.edu

Emilio Gomez

Department of Experimental Psychology
University of Granada
Granada (Spain)
egomez@ugr.es

Abstract. For years, researchers have argued that Complex Problem Solving (CPS) is plagued with methodological problems. The interest of this research paradigm, a hybrid between field studies and experimental ones, is tied to the success of methodological advances that enable performance to be analyzed. This paper introduces a new, abstract conceptualization of *microworlds* research based on two theoretical lines: (1) a representational problem, where protocols can be seen as objects in a feature space and, (2) a similarity measure problem, where a similarity metric has to be proposed. To materialize this conceptualization we introduce Latent Semantic Analysis (LSA), a machine-learning model that induces representations of the meaning of words by analyzing the relation between words and passages in large bodies of representative text, and describe how LSA can be implemented as a theory and technique to analyze performance in CPS, using actions or states as units instead of words and trials instead of text passages. Basic examples of application are provided, and advantages and disadvantages are discussed.

Many real-word decision making and problem solving situations are (1) *dynamic*, because early actions determine the environment in which subsequent decision must be made, and features of the task environment may change independently of the solver's actions; (2) *time-dependent*, because decisions must be made at the correct moment in relation to environmental demands; and (3) *complex*, in the sense that most variables are not related to each other in one-to-one manner. In these situations, the problem requires not one decision, but a long series, and these decisions are, in return, completely dependent on one another. For a task that is changing continuously, the same action can be definitive at moment *t1* and useless at moment *t2*. However,

traditional, experimental problem solving research has focused largely on tasks such as anagrams, concept identification, puzzles, etc. that are not representative of the features described above.

In Europe, a movement of researchers led by Broadbent (e.g., [2]) in the UK and Dörner (e.g., [8]) in Germany were concerned about that fact and started working on a set of computer-based, experimental tasks that are dynamic, time-dependent, and complex, called *Microworlds*[1]. This branch of the thinking and reasoning psychology has been called Complex Problem Solving (e.g., [10]).

Compared to traditional Problem Solving, Complex Problem Solving (CPS) radically changed the kind of phenomena reported, the kind of explanations looked for, and even the kind of data that were generated. However, the results obtained to date are far from being integrated and consolidated. This fact led Funke to affirm that 'Despite 10 years of research in the area, there is neither a clearly formulated specific theory nor is there an agreement on how to proceed with respect to the research philosophy. Even worse, no stable phenomena have been observed' ([11], p. 25). Almost another 10 years after Funke's argument, although more empirical research has been conducted in the area, we cannot say that the situation has changed drastically. At this moment, there is no theory able to explain even part of the specific effects that have been described or how they can be generalized.

A theory of generalization and similarity is as necessary to psychology as Newton's laws are to physics [30]. However, for CPS there is no common, explicit theory to explain why a complex, dynamic situation is similar to any other situation or how two slices of performance taken from a problem solving task can possibly be compared quantitatively. Intractability issues have been raised [23]. This lack of formalized, analytical models is slowing down the development of theory in the field. At least two problems make it difficult to apply the classical problem solving approach to CPS:

(1) The utility of state space representation for tasks with inner dynamics is reduced because in most CPS environments it is not possible to undo the actions. For example, imagine that two participants in *Firechief* [27] are in an identical situation (system state) when the trial starts. One of them proceeds to make a control fire on the right side of a central fire, while the other one is preparing a control fire on the north front of the fire. After 20 seconds have elapsed, the system state is no longer identical for them. Now they have to cope with rather different problems. Moreover, if the first participant wants to apply the same technique that the second participant used, there is absolutely no way to come back to the initial state and begin with a new strategy. This situation is not an issue in static tasks like the tower of Hanoi problem because the last state is always available to 'undo' the wrong actions. Feedback delays (e.g., [1],

[1] This term sometimes has other meanings. For example, educational applications created to teach physics [16], simulated words in the early AI programs like the block word of SHRDLU, [34] and static tasks to study decision making [14] have been called *Microworlds*. However, we are interested only in the tasks that fulfill the conditions described above.

[12]) and upsettingly large number of possible states (e.g., [8], [27]) contribute to the reduced utility of the state space approach.

(2) Traditional techniques of knowledge elicitation seem not to be very suitable: Concurrent verbal protocols consistently interfere with performance [7]; measures based on relatedness judgments like rating correlations or pathfinder distance correlations are not sensitive to context manipulations in naturalistic task like fire fighting [4].

In this paper we introduce a theory and methodology for Complex Problem Solving tasks based on Latent Semantic Analysis (LSA, [24]). The theory addresses issues concerning induction, representation, and application of knowledge. The main idea of LSA is that most knowledge areas contain enormous numbers of weak interrelations that can be used to infer more knowledge than that resulting from simple addition.

In the new LSA perspective, we are not depicting all the possibilities of the system (system's state space), but only the paths that people have actually followed when interacting with it. This offers a realistic view of how the system is understood and used by humans. LSA is a *computational* theory on how environmental constraints are learned and how they can be described. Using the classical Simon's *parable of the ant and the beach* ([31], p. 63), LSA would be describing and inferring the shape of the beach using the constraints that thousands of ants' paths impose n each other. In this sense, LSA can be conceived as a computational extension to theories for describing environmental constraints, such as the abstraction hierarchy (AH, [29], [33]; see also Quesada, Kintsch and Gomez, this issue).

LSA has several interesting features that make it a suitable technique to analyze performance on a complex, dynamic task:

(1) It does not assume independence of decisions; indeed, it uses dependencies between decisions to infer structure. Some methods employed in the past treated CPS performance in a way that assumed that decisions are independent or have 'short term dependencies' only. For example, when transitions between contiguous actions are used as the unit of analysis, the method assumes that the only dependency of interest is the one between an action (or state) and the following one (e.g. [17], [27]).

(2) LSA reduces the dimensionality of the space. Imagine a hypothetical Problem Solving task that, when performed from the beginning to the end in one of the *N* possible ways, traverses 300 states. To make it a really simple example, let us assume that every state is described using 6 dichotomous variables (2^6 = 64 possible states). Since we have 300 states in our sample of performance, there are 64 ^ 300 = 7e541 possible paths in this task. Every sample would be represented as a matrix of 6 x 300 = 1800 values. With LSA, every sample is represented as a vector of only 100- 300 values.

(3) There are no *a-priori* assumptions about *'the beach*'. To continue our previous example, not all of the 7e541 possible paths are followed when one observes actual performances of the task, but only a small proportion of them. In most of the analysis performed on *microworld* data the experimenter has to restrict the variability by imposing some structure (*a-priori*, theoretically driven assumptions) on the data. For example, hypothetical ways of action –strategies- can be described and implemented, and participants' performances can be described in terms of their similarity to the strategies. However, the selection of this theoretical structure (How many strategies are possible? How many are representative enough? Are they generalizable to different conditions?) is indefectibly biasing the results obtained in the analysis. In that sense, the LSA approach is self-organizing, and does not require defining an *a – priori* theoretical structure.

Before we start describing what LSA is and how it can be applied to CPS, we would like to stress some abstract considerations that underlie the approach that we are about to implement. These considerations are independent of the procedure itself (other procedures could be defined using this framework), but, in our opinion, an essential step to dealing with the complexity of the tasks at hand: (1) Each microworld can be conceptualized as a complex, multidimensional feature space. (2) To address the intractability problem, we usually need to create a representation or transformation of this original multidimensional feature space. To do this, we need to find a set of features that represent the characteristics that make participants different, and to obviate those that are not important. (3) Last, each trial of every subject can be conceptualized as an implementation of several values in the feature space. Not only a trial, but every subpart or superpart of a participant performance (strategies or performance patterns) can be thought of as an object in this space.

LSA is one implementation of this framework, and in the following parts of this paper we try to show why we think that it is a successful one. In the next section, we briefly describe *Firechief*, the *microworld* that we use for our examples. An introduction to LSA follows, explaining what LSA is and how it has been used as a theory of knowledge representation and as a tool for text-comprehension-related applications. Next we explain its procedure and mathematical foundations. After that, we describe how LSA can be applied to *microworlds* and show some examples of use. We then compare LSA similarities to judgments of relatedness emitted by human participants exposed to similar information to validate the technique. As a conclusion, we discuss the importance of the approach and the possibilities and future extensions of the technique.

1. A description of the example application task

In our examples, we use the *microworld Firechief* [27], which simulates a forest where a fire is spreading. Their task is to extinguish the fire as soon as possible. In order to do so, they can use helicopters and trucks (each one with particular characteristics) that can be controlled by mouse movements and key presses. There

are four commands that are used to control the movement and functions of the appliances: (1) Drop water on the current landscape segment; (2) Start a control fire (trucks only) on the current landscape segment; (3) Move an appliance to a specified landscape segment, and (4) Design an automatic surveillance area where the vehicle will find and extinguish fires automatically. Command 4 is not considered in the following discussion and was disabled, so it is not further described. Firechief generates pretty complex log files containing every action a participant has done. We have been analyzing these actions to classify participants, in order to detect which strategies lead good performers to a bad performance when the environmental situation changes slightly [28], [5].

Every time a participant perform an action, it is saved in a log file as a row containing action number, command (e.g. drop water or move) or event[2] (e.g., a wind change or a new fire), current performance score, appliance number, appliance type, position, and landscape type. Most of these variables are not continuous, but on a nominal scale, such as type of movement. For more information on the structure of the log files, see [27].

The set of trials that was used in this report (referred as *corpus*) is formed by four experiments (described in Quesada, et al. [28] and Cañas et al. [5]). The particular hypothesis, manipulations and results are not of interest for our current purpose of explaining the application of LSA to *microworld* log files.

2. Introduction to LSA

LSA is a machine-learning model that induces representations of the meaning of words by analyzing the relation between words and passages in large bodies of representative text. LSA is both a method (tool) used in industry to develop technology to improve educational applications, and a theory of knowledge representation used to model well known experimental effects in text comprehension and priming, among others [24]. Latent Semantic Analysis was originally developed in the context of information retrieval [6] as a way of overcoming problems with polysemy and synonymy. Some words appear in the same contexts (synonyms) and an important part of word usage patterns is blurred by accidental and inessential information. As Kintsch [19] put it, why did an author choose a particular word in a specific place instead of some other alternative? The method used by LSA to capture the essential semantic information is dimension reduction, selecting the most important dimensions from a co-occurrence matrix decomposed using Singular Value Decomposition (see below). As a result, LSA offers a way of assessing semantic similarity between any two samples of text in an automatic, unsupervised way.

[2] Events are generated by the system, while actions are generated by the user. Events are also lines in the log file. Only 1-2% of the lines in a log file are events.

LSA has been used in applied settings with a surprising degree of success in areas like automatic essay grading [9] and automatic tutoring to improve summarization skills in children [19]. As a model, LSA's most impressive achievements have been in human language acquisition simulations [24] and in modeling of high-level comprehension phenomena like metaphor understanding, causal inferences and judgments of similarity [20].

The best way of understanding the LSA induction mechanism is through an example, on text passages and words, and the best small-scale example is described in Landauer, Foltz and Laham [25][3]. The reader is strongly recommended to consult these references for a better understanding of the technique.

Although LSA has been mostly used on text corpora, our basic point is that LSA can be applied to any domain of knowledge where there are a high number of weak relations between tokens, as in Complex Problem Solving log files. Instead of using word co-occurrence statistics and huge samples of text, we have used a representative amount of activity in controlling dynamic systems, and actions or states have been used to develop the much-wanted objective measure of similarity in the changing, time-dependent, highly complex experimental tasks known as *microworlds.* The next sections show the basic steps to perform the analysis and will present some examples of the powerful analysis that can be conducted.

3. LSA procedure

The procedure starts with creating a matrix of actions[4] by trials. If the log files record state information instead of actions, states can be used[5]. Note that this is not an exhaustive state space, or a mapping of all possible transitions between actions (since in most of the systems –other than small ones like Hayes and Broadbent's sugar factory and the like- this task would be excessively demanding, see Buchner, Funke and Berry, [3]). Our *corpus* was composed of 360,199 actions in 3441 trials. Among them, only 75,565 were different actions, which means that on average each action appears 6.25 times in the *corpus.* Note that we are representing *only* the information that actual people interacting with the system experienced, not all possible actions in this *microworld*.

3 The same example has been used also in Deerwester et al. [6] and Landauer and Dumais [24].
4 The information contained in each action (in this microworld, an action is one line of code in the log files) was transformed into a single token using underscores instead of the spaces that separated values of different variables.

5 From here on, we talk about actions as the basic token. However, states can be considered the basic token with no further assumptions for the model. We use actions because *Firechief* states are forest matrices (24x15) containing information about type of terrain and fire status (safe, burned, fire intensity). In this particular example it is more convenient to use actions than states, although in other microworlds like *DuressII* states are a better option (see Quesada, Kintsch, and Gomez, this issue).

Each of these 75,565 rows stands for a unique action, and each of the 3441 columns stand for a trial. Each cell contains the frequency with which the action of its row appears in the trial denoted by its column. Note that most of the cells will contain a frequency of zero, since most actions appear in only a few trials and do not appear in the rest.

This matrix of frequencies is decomposed using Singular Value Decomposition (SVD). In SVD, a rectangular matrix is decomposed into the product of three other matrices. The first matrix is containing the original row entities but the columns are derived orthogonal factor values; the third matrix describes the original column entities in the same manner; the second (and most important) matrix is a diagonal matrix containing scaling values such that when the three matrices are multiplied, the result is the original matrix. It has been mathematically proven that any matrix can be decomposed and then recomposed perfectly using only as many factors as the smallest dimension of the original matrix. However, the interesting phenomenon occurs when the original matrix is recomposed using fewer dimensions than necessary: the reconstructed matrix is a least-squares best fit. This feature has been successfully used in fields ranging from satellite information compression to solving huge linear equation systems (e.g., [32]).

When the actions-by-trials matrix is recomposed using a small fraction of the available dimensions (usually between 100 and 300 first dimensions, zeroing out the rest), the new matrix contains information that has been inferred from the dependencies between actions and the context where these actions appeared. In fact, the contexts where these actions did not appear are as important -carry as much information- as those where they did. The *microworld* is a new multidimensional feature space, where both actions and context (trials) are *represented* in a way that amplifies those characteristics that make participants different, and obviate those that are not important for classifying their performance. Other features of the theory, such as the election of the cosine between vectors as a measure of similarity[6], or the weighting schemes that are applied to the frequency matrix[7] are of less importance.

6 Cosine and correlation have a very similar interpretation. But classic studies on information retrieval [18] address an important advantage in the cosine measure. Both measures can be conceptualized as an inner product divided by the product of vector lengths. The main difference is that in correlation the scores are normalized by subtracting the mean of each dimension. This normalization creates a vector whose components are deviations of corresponding components in the original vector from the mean of the original vector. This causes the correlation measure to lose angle monotonicity and radial monotonicity, and it is why the cosine measure is preferred.

7 See [15].

4. LSA Applied to *Microworlds: S*ome Examples and Possible Analysis

This transformation, and its particular use in former applications of LSA to infer word meaning, can be related in a very interesting way when the application to Complex Problem Solving is considered. Some actions can be considered as *functional synonyms*: they appear in the same contexts, and fulfill approximately the same function. The following example illustrates this idea.

In Table 1, some actions are defined. For simplicity, some variables that are normally contained in the log files have been removed[8]. Example1 contains a movement to the point (11, 9) in the screen, which is of type forest, and then, a drop water action there. Example 3 shows a very similar picture, where the movement is done to a contiguous cell (10,9) that is also of type forest. From a human point of view, these two examples are highly similar. For LSA they are too, as can be seen in their similarity expressed as a cosine of 0.854 in Table 2.

The second example has a rather different 'meaning' since the cell targeted is (15,15), quite far from the cell used in examples 1 and 3. The cosines between them and example2 (.124 and .125) are, accordingly, smaller than the one between 1 and 3.

Example 4 describes an action that has been performed in the same cell as in example 1 (11,9), but this time is a control fire instead of a drop-water action. The cosine between 1 and 4 is high (0.56), expressing a certain similarity between the two actions, but not as high as in examples 1 and 3, where the objective similarity is more evident.

Tables 3 and 4 depict a more complex example where wider slices of performance (8 actions) are compared. The samples labeled example1, example2, and example 3 are beginnings of trials that have been selected randomly from the *corpus.* This time, all the usable information contained in the log file is displayed. Each action has six components: type of action, appliance number, appliance type, departure cell, arrival cell and type of arrival cell.

	Time *t1*	**Time *t2***
Example 1	move_11_9_forest	drop_11_9_forest
Example 2	move_15_15_forest	drop_15_15_forest
Example 3	move_10_9_forest	drop_10_9_forest
Example 4	move_11_9_forest	control_11_9_forest

Table 1: Example of how LSA captures similarity at a very molecular level (action level). Time T1 and Time T2 are two consecutive snapshots of performance, containing one action each.

8 Types of appliance, appliance number and departure point have been removed, as well as generation number (a timestamp), and overall performance score at this particular moment. See Omodei and Wearing [27] for more information on the variables.

	Example 1	Example 2	Example 3	Example 4
Example 1		0.124049	0.854328	0.6626815
Example 2			0.1259655	0.0772655
Example 3				0.5660885
Example 4				

Table 2: Comparisons between the four examples. Cells are cosines that represent similarity between all possible pairs. These have been obtained by averaging the cosines for different temporal moments. This means that the value in cell example1 – example2 has been calculated as the average of the cosine (example1 in *t1* vs. example2 in *t1*) and the cosine (example1 in *t2* vs. example2 in *t2*).

One difficulty arises. When LSA is used on text, cosines are easily understood since every reader has an intuitive experience of meaning (e.g., the sentences `'The man was driving a yellow car'` and `'The man was traveling in a red car'` have a cosine of .89, and our common sense tells us that these sentences convey similar information). When LSA is used on samples of performance from a *microworld*, there is no way the reader can understand the 'meaning' of the log files without watching a replay or having an extraordinarily vivid imagination plus experience with the task. For most researchers, the following extracts in table 3 are hardly understandable. For researchers familiar with *Firechief*, they should be as clear as a piece of sheet music to a musician. However, understanding the contents of these examples is not *conditio sine qua non* for understanding the advantage of LSA analysis over two other methods, namely exact matching and correlation between transition matrices. Suffice it to say that examples 1 and 2 are very similar and example 3 is very different from them. The attentive observer could induce this from the locations (coordinates in the Firechief map), the type of actions, and type of landscape cell.

An *exact matching* method would count the number of times that the same action is in two examples. Then, the number of matchings divided by the total number of actions in the example would be the similarity between two samples. This method would render a similarity of 1/8 between example 1 and 2, and zero in comparisons 1 vs. 3 and 2 vs. 3. This method is an equivalent to keyword counting in text and has been proven to be insufficient to capture similarity in meaning, because of the polysemy and synonymy effects described before.

A somewhat more flexible method is the use of *transitions between actions*, proposed by Howie and Vicente [17] and used in Quesada et al. [28] and Cañas et al. [5]. It is based on counting the number of times that one type of action precedes any other type. The frequencies of every transition are registered in cells in a table, and then the resulting tables for two examples are correlated. The method cannot account for all the variability in actions, because of the huge amount of zero entries that artificially increase the correlation, so only action type was considered. This analysis is shown in tables 4(*a*,*b*,*c*). Since lots of information contained in the log files has been dropped,

the method does not distinguish between these examples. The correlation between table *a* and *b* is 0.971; exactly the same correlation is obtained for tables b and *c*, and the comparison between *a* and *c* is 1 since the sequence of *type of action* is exactly the same. Note that even though this method is seriously limited and is showing very flawed similarity estimations, it has been used in several works in the literature.

example1	**Example2**
move_2_truck_4_11_13_3_forest	move_2_truck_4_11_12_15_forest
move_1_truck_4_14_16_14_forest	move_1_truck_4_14_13_5_forest
move_3_copter_8_6_11_12_forest	move_4_copter_11_4_11_9_forest
move_4_copter_11_4_11_9_forest	drop_water_4_copter_11_9_forest
control_fire_2_truck_13_3_forest	move_4_copter_11_9_13_8_forest
control_fire_1_truck_16_14_forest	control_fire_2_truck_12_15_forest
move_2_truck_13_3_17_7_clearing	move_2_truck_12_15_13_14_forest
move_1_truck_16_14_20_12_forest	control_fire_2_truck_13_14_forest

example1	**Example3**
move_2_truck_4_11_13_3_forest	move_2_truck_4_11_2_2_pasture
move_1_truck_4_14_16_14_forest	move_1_truck_4_14_0_5_forest
move_3_copter_8_6_11_12_forest	move_4_copter_8_6_8_4_clearing
move_4_copter_11_4_11_9_forest	move_3_copter_8_6_8_10_clearing
control_fire_2_truck_13_3_forest	control_fire_2_truck_2_2_pasture
control_fire_1_truck_16_14_forest	control_fire_1_truck_0_5_forest
move_2_truck_13_3_17_7_clearing	move_4_copter_8_4_4_2_forest
move_1_truck_16_14_20_12_forest	move_3_copter_8_10_2_3_clearing

Table 3: First 8 movements in 3 slices randomly sampled from the *Firechief* experiments described in Quesada et al., [27] and Cañas et al. [5]. When an action is shared by two extracts, it is marked as a shaded cell.

Finally, let us look at the results of similarity estimation using LSA cosines. The vector representing the sample has been calculated as the average of the 8 action vectors. Example 1 vs. example 2 has a cosine of 0.721928, a high similarity value. Even though these samples share only 1/8 of the actions, LSA has correctly inferred that the remaining actions, although different, are functionally related. Comparisons between 1-3 and 2-3 have cosines as low as 0.056770 and 0.071135 respectively, showing that these performances are different indeed.

Table 5 shows a comparative between the three methods. The exact matching method is characterized by an underestimation of the similarity. Since the current examples only have 8 actions from the beginning of the trial and one happened to be the same in examples 1 and 2, the similarity estimated by this method is not zero, but .125. However, in real examples where the number of actions is much higher, the

probability of finding repeated actions decreased, and so does the similarity calculated with this method, no matter what the objective similarity is. This method would predict correctly the lack of similarity, but it is not sensitive to real similarity. Since virtually every pair of trials in *Firechief* shares very few actions, or even none, even when they are objectively similar, this method is not appropriate.

(a)

Example 1

	drop	move	control
drop	0	0	0
move	0	4	1
control	0	1	1

(b)

Example 2

	drop	move	control
drop	0	1	0
move	1	2	2
control	0	1	0

(c)

Example 3

	drop	move	control
drop	0	0	0
move	0	4	1
control	0	1	1

Table 4 Transitions between actions considering *type of action* only as described in Quesada et al., [27] and Cañas et al. [5], for the examples 1,2 and 3. Cells contain frequencies of the transition defined by its row and its column. For instance, the number 4 in the center cell in table 4a means that in example 1 the transition move-move has appeared four times.

method	**example1 to example2**	***comparisons*** **example1 to example3**	**example2 to example3**
Objective similarity	high	low	Low
LSA Cosine	0.7219	0.0567	0.0711
Exact matching	0.125	0	0
Transitions between actions	0.971	1	0.971

Table 5 Similarity between examples 1, 2, and 3 using the methods LSA, exact matching and transitions between actions as described in Quesada et al., [27] and Cañas et al. [5].

On the other hand, the Transitions between actions method is characterized by an overestimation of the similarity. Most of the information contained in the log files is disregarded due to the limitations of the technique (exponential growing of the number of possible pairs). This produces that series of actions that are completely unrelated are considered exactly the same because the 'type of action' information is coincidentally the same, as in examples 1 and 3. In contrast, the similarity values computed by LSA reflect the objective similarity very closely.

5. Correlations between LSA and Human Judgment

An important point is that LSA similarity judgments can be compared with human standards. Up to this point in the paper, only the experimenter's common sense and knowledge of the task have been used to validate the results. However, if LSA captures similarity between complex problem solving performances in a meaningful way, any person with experience on the task could be used as a validation. The problem is that, contrary to what happens when one uses LSA to model text comprehension, it is not easy to find experts in the task at hand. Everybody is a perfect example of the expert reader, but not everybody is an expert in controlling the particular dynamic system called *Firechief*. To test our assertions about LSA, we recruited 3 persons and exposed them to the same amount of practice as our experimental participants, so they could learn the constraints of the task.

After 24 practice trials, these participants were used to assess the external validity of LSA similarities. Using *Firechief's* replay option, participants had to watch 7 pairs of trials (at a pace faster than normal) and express similarity judgments about these pairs. People watched a randomly ordered series of trials, in a different order for each participant, which were selected as a function of the LSA cosines (pairs A, B, C, D, E, F, G with cosines 0.75, 0.90, 0.53, 0.60, 0.12 and 0.06 respectively). One of the pairs was presented twice to measure test-retest reliability. That is, for example, pair G was exactly the same as pair A for one participant, the same as pair F for another participant, etc. Filling out a form that presented all the possible pairings of 'stimuli pairs'[9] were presented. They had to answer which pair seemed more similar to them. For example, LSA would say that pair B is more related (>) than pair C, since the cosines are 0.90 and .53 respectively.

LSA cosines predicted human similarity judgments very well indeed. For 3 participants in this pilot study, the proportion of agreement LSA-human was 6/19, 14/19, and 13/19 respectively. Participants with strong agreement with LSA also showed more consistency in their judgments, that is they answered to the repeated item in the same way. The participant who had low agreement with LSA had a very bad performance grading the repeated item, which suggests that she could have been answering randomly. Even so, the average agreement between LSA cosines and human judgments was 0.57, far superior to the agreement expected by chance, 0.5 ^ 19 = 2e-5.

However, some problems have to be considered. The application of any analysis based on contingency tables was not possible due to two issues. First, because of the small number of observations (19) some cells in the contingency table had a frequency less than five, which make the results of procedures such as the phi coefficient unreliable. Second, the stimuli were presented using all possible combinations, that is AB, AC, AD, … FG. Due to the characteristics of the stimuli

[9] This method was selected so the cosines were very distinctive and easy to compare. A subspace where a few items (instead of pairs) have very distinctive cosines could be found too, but this approach was used because the sample of stimuli was easier to obtain.

(see cosine list), most of the pairs should be answered with a 'greater than' response, and this is contrary to the assumptions of the phi coefficient, that will give flawed results if there is an extreme imbalance in the marginal distributions of the contingency table. As a consequence of this flaw in the design of the human judgments pilot experiment we cannot distinguish clearly if the agreement between LSA and participant was due to a response bias 'answer always greater than' or to the fact that the similarity that participants perceived was captured by the LSA cosines. Future designs will address these shortcomings, for example having a greater number of items to judge, and presenting them in a way that distributes the correct responses equally (example half of the options in the form BA, and half AB). A possible alternative is to substitute the dichotomy in the answer with an ordinal, or even interval scale, but we still need to make sure that participants are able to do fine-grained distinctions when emitting similarity judgments for replays of this kind of tasks.

5. Conclusions

LSA seems to be a promising new way of approaching Complex Problem Solving performance that overcomes some of the known limitations of previous methods. Apart from the features listed in the introduction, there are some pragmatic LSA advantages worth noting: (1) Since the basic unit of analysis is the token (action or state), even systems that are described in terms of nominal (discrete) variables can be analyzed. Both actions and states can be used as units. (2) The 'semantic' matching mechanism permits discovery of similarities beyond simple coincidence in the log files containing actions or states. That is, participants who are using different interventions to realize the same strategy will be considered similar even if their log files share no actions (or states). (3) The level of granularity (whether we are working with individual tokens, slices of performance, whole trials, or collections of trials) is not defined *a-priori*. Since every object, from one token to the participant's whole performance, can be represented as a vector of n dimensions, analysis can be performed at any level of detail.

Not less important are the following disadvantages: (1) A huge sample of data is needed. (2) Order effects are not taken into account. This means that for LSA a trial where the tokens have been scrambled to a random order has exactly the same meaning as the original version. This could be considered an important shortcoming (although see [26] for the contrary view), and current research is being performed in this direction. (3) Though the SVD analysis is common practice and can be found in several statistical packages[10], a powerful computer is needed to run large analysis.

One more question is in the air: Is LSA a method or a theory of CPS? The answer may be: both. In a seminal paper, Gigerenzer [13] argued that the way social scientists

10 R and matlab have an SVD function. To inquire about LSA computer programs, address Telcordia (Formerly know as Bellcore, who owns the patent for information retrieval) at http://lsi.argreenhouse.com/

discover theories of mind is closely bound to the emergence of new tools for data analysis, rather than new data. He exemplifies this phenomenon with the *signal detection theory*, a cognitive theory of perception that is a generalization of the Neyman-Pearsonian statistical hypothesis testing technique. The tools-to-theory proposal exemplifies perfectly the situation of LSA as a statistical method that can be understood as a theory of knowledge representation. At this moment, LSA is a tool that can be used to complement and extend the reach of some other non-computational theories. However, learning in a CPS environment is often thought of as predicting each successive constituent from those already analyzed on the basis of the knowledge acquired from past experience with the system (e.g. [11]). It should not be too difficult to employ LSA to implement that proposal. The LSA approach to this idea would rely on the following assumptions: (a) Actions would be generated by comparing a small amount of preceding context with a vast amount of knowledge acquired in past interactions with the system. (b) The adequacy of an action would be computed by examining past contexts where this action has appeared, and comparing this set of contexts with the current one. This assumption would agree with recognition-based theories such as Klein's Recognition Primed Decision Making ([21], [22]). (c) In LSA, context and actions (or states) are both represented as vectors, and similarity is estimated using the cosine between the vectors. In normal conditions (lots of knowledge about the situation), the action selected would be the one with the highest cosine with the current context. However, the validity of these assumptions is not assured, and future research should address this issue.

Acknowledgements

Our acknowledgements to Tom Landauer for proposing interesting issues concerning the selection of the unit of analysis in Complex Problem Solving. We are grateful to Kim Vicente and John Hajdukiewicz for sharing experimental data and insightful discussions during a visit of José Quesada to the University of Toronto. Many thanks to Bill Oliver, who provided passionate methodological discussions and theoretical contributions. The manuscript was thoroughly revised by Nancy Mann, improving greatly the language and presentation.

This research was in part supported by Grant EIA – 0121201 from the national science foundation.

References

1. Brehmer, B., Feedback delays in complex dynamic decision tasks. In P. Frensch and J. Funke, (Eds.) Complex Problem Solving: The European Perspective. Hillsdale, NJ: Lawrence Erlbaum (1995)

2. Broadbent, D. E. Levels, hierarchies and the locus of control. Quarterly Journal of Experimental Psychology 32, (1977) 109-118

3. Buchner, A., Funke, J., Berry, D.: Negative correlations between control performance and verbalizable knowledge: Indicators for implicit learning in process control tasks? Quarterly Journal of Experimental Psychology, 48A, (1995) 166-187

4. Calderwood, R.: The role of context in modeling domain knowledge. Unpublished doctoral dissertation, University of New Mexico (1989)

5. Cañas, J.J., Quesada, J.F., Antolí, A., Fajardo, I.: Cognitive flexibility and adaptability to environmental changes in dynamic complex problem solving tasks (submitted to ergonomics)

6. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. Indexing By Latent Semantic Analysis. Journal of the American Society For Information Science, 41, (1990) 391-407

7. Dickson, J., McLennan, J, Omodei, M. M.: Effects of concurrent verbalization on a time pressured dynamic decision task. Journal of General Psychology. 127, (2000) 217-228

8. Dörner, D.: Wie Menschen eine Welt verbessern wollten und sie dabei zerstörten How people wanted to improve the world. Bild der Wissenschaft, Heft 2 (populärwissenschaftlicher Aufsatz) (1975)

9. Foltz, P. W. , Laham, D., Landauer, T. K.,: The Intelligent Essay Assessor: Applications to Educational Technology Interactive Multimedia Education Journal of Computer enhanced learning On-line journal., 1(2). http://imej.wfu.edu/articles/1999/2/04/index.asp (1999)

10. Frensch, P., Funke, J.: Complex Problem Solving: The European Perspective. Hillsdale, NJ: Lawrence Erlbaum (1995)

11. Funke, J.: Dealing with dynamic systems: Research strategy, diagnostic approach and experimental results. German Journal of Psychology, 16, (1992) 24-43

12. Gibson, F. P.: Feedback delays: How can decision makers learn not to buy a new car every time the garage is empty? Organizational Behavior and Human Decision Processes Vol. 83, No. 1 (2000) 141 - 166

13. Gigerenzer, G.: From tools to theories: A heuristic of discovery in cognitive psychology. Psychological Review, 98 (1991) 254-267

14. Green D. W.: Understanding microworlds. Quarterly Journal of Experimental Psychology, Section A-Human Experimental Psychology, 54 (3) (2001) 879-901

15. Harman, D. An experimental study of the factors important in document ranking. In Association for Computing Machinery Conference on Research and Development in Information Retrieval, New York: Association for Computing Machinery (1986)

16. Henderson, L., Klemes, J., Eshet, Y.: Just playing a game? Educational simulation software and cognitive outcomes. Journal of Educational Computing Research, 22 (1) (2000) 105-129

17. Howie, D. E., Vicente, K. J.: Measures of operator performance in complex, dynamic microworlds: Advancing the state of the art, Ergonomics, vol. 41, (1998) 85-500

18. Jones W. P. and Furnas, G. W.: Pictures of relevance: A Geometric Analysis of similarity measures. Journal of the American society for information science, 38(6) (1987) 420 -442

19. Kintsch, E., Steinhart, D., Stahl, G., Matthews, C., Lamb, R., the LSA Research Group Developing summarization skills through the use of LSA-backed feedback, Interactive Learning Environments, 8 (2), (2000) 87-109

20. Kintsch, W. Predication. Cognitive Science 25, (2001) 173-202

21. Klein G.: A Recognition-primed decision model of rapid decision making. In Klein, G.A., Orasanu, J., Calderwood, R., Zsambok, C.E. (Eds.), Decision making in action: Models and methods. Norwood, NJ: Ablex Publishing Corporation, (1993)

22. Klein, G.: The recognition-primed decision model: looking back, looking forward. In Zsambok C., Klein, G. (Eds.), Naturalistic Decision Making. Mahwah, N.J.: Lawrence Erlbaum Associates, 1997

23. Klein, G.A., Orasanu, J., Calderwood, R., Zsambok, C.E. (Eds.), Decision making in action: Models and methods. Norwood, NJ: Ablex Publishing Corporation, (1993)

24. Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240

25. Landauer, T. K., Foltz, P. W., Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284

26. Landauer, T. K., Laham, D., Rehder, B., Schreiner, M. E.,: How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In Shafto, M. G., Langley, P. (Eds.) Proceedings of the 19th annual meeting of the Cognitive Science Society. Mahwah, NJ: Erlbaum (1997) 412-417

27. Omodei, M. M., & Wearing, A. J. (1995). The Fire Chief microworld generating program: An illustration of computer-simulated microworlds as an experimental paradigm for studying complex decision-making behavior. Behavior Research Methods, Instruments & Computers, 27, 303-316

28. Quesada, J.F., Cañas, J.J., Antoli, A.: An explanation of human errors based on environmental changes and problem solving strategies. In Wright, P., Dekker, S., Warren C.P. (Eds.) ECCE-10: Confronting Reality. Sweden: EACE (2000)

29. Rasmussen, J.: The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man, and Cybernetics, SMC-15 (2): (1985) 234-243

30. Shepard, R.N. (1987). Toward a universal law of generalization for psychological science. Science, 237, 1317-1323

31. Simon, H.A.: The sciences of the artificial. MIT press (1981)

32. Strang, G.: Linear Algebra and Its Applications. Harcourt, Brace and Jovanovich (1988)

33. Vicente, K.: Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-based work. Lawrence Erlbaum associates, London (1999)

34. Winograd, T: A procedural model of language understanding. In Schank, R., Colby, K. (eds.) : Computers models of thought and Language, Sand Francisco W.H. Freeman (1972)