Tutoring Systems based on Latent Semantic Analysis

Benoît Lemaire
L.S.E.
University of Grenoble II
BP 47 – 38040 Grenoble Cedex 9
France
Benoit.Lemaire@upmf-grenoble.fr

Abstract. Latent Semantic Analysis is a model of language learning, based on the exposure to texts. It predicts to which extent semantic similarities between words are learned from reading texts. We designed the framework of a tutoring system based on this model. Domain examples and student productions are represented in a high-dimensional semantic space, automatically built from a statistical analysis of the co-occurrences of their lexemes. We also designed tutoring strategies to select among the examples of a domain the one which is best to expose the student to. Two systems are presented: the first one successively presents texts to be read by the student, selecting the next one according to the comprehension of the prior ones by the student. The second plays kalah with the student in such a way that the next configuration of the board is supposed to be the most appropriate with respect to the semantic structure of the domain and the previous student's actions.

1 Introduction

This paper relies on a new model of human learning called Latent Semantic Analysis (LSA) [8]. Although the authors sketched out the outline of a general model of learning, they mainly focused on language learning. This model predicts to which extent semantic relations are drawn by a subject reading a set of texts. It does so by automatically extracting semantic relations between words by a method presented in the next section.

Our contribution is twofold. First, we designed a framework for representing both domain and student knowledge of a tutoring system by means of LSA. We also designed tutoring strategies for selecting among the various stimuli the student can be exposed to, the one which is supposed to be the best for improving learning. Second, we built two systems we implemented on top of this framework. In the first one, stimuli are texts whereas in the second one they are boards of a strategy game.

2 The Latent Semantic Analysis Model of Learning

LSA was primarily designed as a tool for text retrieval [2] but because of its good performance, its scope was extended to information filtering [5], cross-language information retrieval [4], automatic grading of essays [6], measuring of text coherence [8], assessment of knowledge [11], machine learning [9] and then modelling human learning [8].

Before presenting LSA as a model of learning and knowledge representation, we will present its principles.

2.1 LSA: a Tool for Text Retrieval

One of the problems in the field of text retrieval is to be able to retrieve pieces of texts given a list of keywords. However, because of polysemy, synonymy and inflexion, retrieving only the texts that contain one or more of the keywords does not work well. For instance, Steinbeck's book *Of Mice and Men* should be retrieved given the keywords *mouse* and *man* although none of these words appear in the title. Therefore, retrieval should also be based on semantic information.

In order to perform such semantic matching, LSA relies on large corpora of texts to build a semantic high-dimensional space containing all words and texts, by means of a statistical analysis. This semantic space is built by considering the number of occurrences of each word in each piece of text (basically paragraphs). For instance, with 300 paragraphs and a total of 2000 words, we get a 300x2000 matrix. Each word is then represented by a 300-dimensional vector and each paragraph by a 2000-dimensional vector. Nothing new so far since it is just occurrence processing. The power of LSA rather lies in the reduction of these dimensions. It is this process that induces semantic similarities between words. All vectors are reduced by a method close to eigenvector decomposition to, for instance, 100 dimensions. The matrix X is decomposed as a unique product of three matrices: $X = T_0 S_0 D'_O$ such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. This is called singular value decomposition. Then only the 100 columns of T_0 and D_0 corresponding to the 100 largest values of S_0 are kept, to obtain T, S and D. The reduced matrix $\overline{X}$ such that: $\overline{X} = TSD'$ permits all words and pieces of texts to be represented as 100-dimensional vectors. It is this reduction which is the heart of the method because it extracts semantic relations: if a word (i.e. bike) statistically co-occurs with words (i.e. handlebars, pedal, ride) that statistically co-occur with a second word (i.e. bicycle) *and* the first word statistically does not co-occur with words (i.e. flower, sleep) that do not co-occur with the second one, then the two words are considered quite similar. If the number of dimensions is too small, too much information is lost. If it is too big, not enough dependencies are drawn between vectors. A size of 100 to 300 gives the best results in the domain of language [8].

This method is quite robust: a word could be considered semantically close to another one although they never co-occur in texts. In the same way, two documents could be considered similar although they share no words. An interesting feature of the method is that the semantic information is derived only from the lexical level. There is no need to represent a domain theory.

Several experiments were performed which showed that LSA works quite well. An experiment [8] consisted in building a general semantic space from a large corpora of English texts, then testing it with the synonymy tests of the TOEFL (Test Of English as a Foreign Language). Given a word, the problem is to identify among 4 other words the one that is the semantically closest. LSA performed the test by choosing the word with the highest similarity between its vector and the vector of the given word. LSA results compare with the level of foreign students admitted to American universities.

2.2 LSA: a Model of Learning

Apart from interesting results in the field of text retrieval, LSA can also be viewed as a general model of human learning. First of all, we will describe LSA as a model of language learning but we will see later that it can be extended to other kinds of knowledge.

LSA is a model of the way humans learn from texts since, given a set of texts read by a subject, it predicts to which extent semantic relations between words are learned. The more texts LSA gets, the more accurate are the semantic proximities.

The model has been tested by simulating word learning between age 2 and 20 [8]. They estimated that human beings read about 3500 words a day, and learn an average of 7 to 15 words per day during that period. If provided with a similar amount of texts, LSA learns 10 words a day to get a performance similar to the humans by the age of 20 (defined as the performance to the TOEFL test described earlier). This result is coherent with the human rate of learning. A similar method based on a high-dimensional representation shows its ability to model the human semantic memory [10] .

We formalized that model in the following way:

- A domain D is composed of lexemes. In the domain of language, lexemes are words. In the domain of problem solving, lexemes are facts and conclusions (`high-fever`, `prescribe-penicillin`, `meningitis`, etc. in the domain of medicine). In the domain of game playing, lexemes are positions of pieces (`pawn-in-E2`, `queen-in-F4`, etc. in chess).

- A student learns the domain by being exposed to sequences of lexemes (sequences of words, sequences of facts and conclusions, sequences of pieces' positions, etc.).

- What is learned is semantic similarities between lexemes or sequences of lexemes (for instance, `high-fever` and `meningitis`). In chess, two boards can be semantically similar although their pieces are not in the same positions; it is a characteristic of chess masters to be able to recognize two boards as being similar.

- LSA predicts the semantic similarity between two lexemes or sequences of lexemes given the sequences of lexemes the student has been exposed to.

Some of the sequences of lexemes that are presented to the student will highly improve learning because their structure map the semantic structure of the domain. Other sequences will be of poor interest for the student because they are either too close to or too far from the student knowledge. Knowing which sequence is best for the student depends on the semantic structure of the domain and the student knowledge. Therefore, we need to represent both in the LSA formalism of knowledge representation. Afterwards, we will be able to design tutoring strategies for selecting the sequences of lexemes that are best to present to a given student. All this is akin to the well-known structure of tutoring systems [13]: expert module, user model and pedagogical module.

3 High-Dimensional Representation of Domain Knowledge

One of the main interests of our approach is that the representation of domain knowledge is automatically built from examples. These examples should be semantically valid and therefore given by experts (verbally or from books). For instance, in the domain of language learning, examples are just well-formed texts. In the domain of game playing, examples are boards as well as an indication whether it is a winning board or a losing board (this information is obtained at the end of the game).

From these examples, LSA builds a semantic space in which all lexemes and examples are represented. There is no need to build by hand semantic networks or logic formulas to represent the domain. All is automatically done by LSA. The only requirement is to find a good formalism to represent the examples.

We do not need to represent all possible examples of the domain, but only enough of them to statistically capture a significant part of the latent semantics of the domain.

4 High-Dimensional Representation of Student Knowledge

In the same way, we represent the student knowledge, that is the student's meaning of entities (we call entity an element of the semantic space, either a lexeme or a sequence of lexemes). Lexemes are described twice:

- as a domain entity to represent the right meaning of the term, constructed as shown previously from the word usage in the language;
- as a student entity to represent the student's meaning of the term;

For instance, the semantic space may contain one instance of `pneumonia` as a domain entity and one instance as a student entity. In the same way, sequences of lexemes are represented in the semantic space. Before being represented, that knowledge needs to be extracted. There are several ways to do that:

- the student freely produces one or several texts: student entities are therefore new entities;
- all the sequences of lexemes the student was exposed to and tested about are represented in the space. That way, student entities are domain entities weighted by a score corresponding to the comprehension of the domain entity by the student.

The goal is that the student entities cover all the domain entities. If the student entities are all in the same part of the space, it probably means that the student has a gap in his knowledge. In that case, the goal of the system is to provide him with appropriate sequences of lexemes so that his knowledge cover a larger part of the space.

5 Tutoring strategies

Now that we have a representation of both domain and student knowledge, we need to design tutoring strategies. As we said before, in the LSA model, learning results from the exposure to sequences of lexemes. By being exposed to sequences of lexemes in a random fashion, the student would certainly learn some lexemes in the same way a child learn new words by reading various books. However, the process of learning could be speeded up by selecting the right sequence of lexemes given the current state of student entities. Therefore, the problem is to know which text (for language learning) or which move (for game learning) has the highest chance of enlarging the part of the semantic space covered by the student entities.

5.1 Selecting the closest sequence

Suppose we decide to select the sequence which is the closest to the student sequences. If $\{s_1, s_2, \ldots s_n\}$ are the student sequences and $\{d_1, d_2, \ldots d_p\}$ the domain sequences, we select d_j such that: $\sum_{i=1}^{n} proximity(s_i, d_j)$ is minimal. Figure 1 shows that selection in a 2-dimensional representation (remind that LSA works because it uses a lot of dimensions).

Let us illustrate this by means of an example. Suppose the domain is composed of 82 sequences of lexemes corresponding each one to an Aesop's fable. Then suppose that a beginner student was asked to provide an English text in order for the process to be initiated. The user model is composed of only this sequence of lexemes:

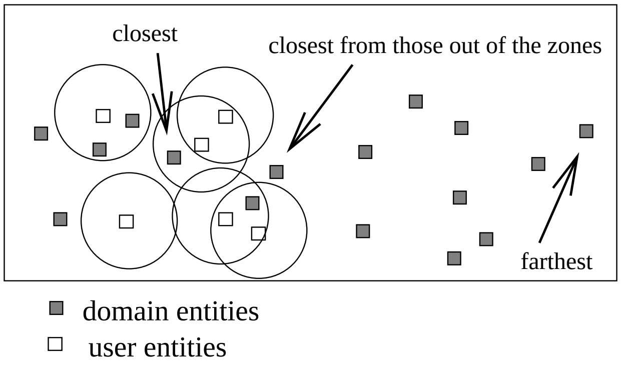

Figure 1: *Various ways of selecting the next sequence*

> My English is very basic. I know only a few verbs and a few nouns. I live in a small village in the mountain. I have a beautiful brown cat whose name is Felix. Last week, my cat caught a small bird and I was very sorry for the bird. He was injured. I tried to save it but I could not. The cat did not understand why I was unhappy. I like walking in the forest and in the mountain. I also like skiing in the winter. I would like to improve my English to be able to work abroad. I have a brother and a sister. My brother is young.

Running LSA, the closest domain sequence is the following:

> Long ago, the mice had a general council to consider what measures they could take to outwit their common enemy, the Cat. Some said this, and some said that; but at last a young mouse got up and said he had a proposal to make, which he thought would meet the case. ”You will all agree,” said he, ”that our chief danger consists in the sly and treacherous manner in which the enemy approaches us. Now, if we could receive some signal of her approach, we could easily escape from her. I venture, therefore, to propose that a small bell be procured, and attached by a ribbon round the neck of the Cat. By this means we should always know when she was about, and could easily retire while she was in the neighborhood.” This proposal met with general applause, until an old mouse got up and said: ”That is all very well, but who is to bell the Cat?” The mice looked at one another and nobody spoke. Then the old mouse said: It is easy to propose impossible remedies.

It is hard to tell why this text ought to be the easiest for the student. A first answer would be to observe that several words of the fable occured already in the student’s text (like cat, young, small, know, etc.). However, LSA is not limited to occurrence recognition: the mapping between domain and student’s knowledge is more complex. A second answer is that the writer of the first text actually found that fable the easiest from a set of 10 randomly selected ones. The third answer is that LSA has been validated several times as a model of knowledge representation; however, experiments with many subjects need to be performed to validate that particular use of LSA.

Although the closest sequence could be considered the easiest by the student, it is probably not suited for learning because being too close to the student’s knowledge.

5.2 Selecting the farthest sequence

Another solution would be then to choose the farthest sequence (Figure 1):

> A Horse and an Ass were travelling together, the Horse prancing along in its fine trappings, the Ass carrying with difficulty the heavy weight in its panniers. ”I wish I were you,” sighed the Ass; ”nothing to do and well fed, and all that fine harness upon you.” Next day, however, there was a great battle, and the Horse was wounded to death in the final charge of the day. His friend, the Ass, happened to pass by shortly afterwards and found him on the point of death. ”I was wrong,” said the Ass: Better humble security than gilded danger.

That sequence was found quite hard to understand by our writer. Choosing the farthest sequence is probably neither appropriate for learning, because it is too far from the student's knowledge.

5.3 *Selecting the closest sequence among those that are far enough*

None of the previous solutions being satisfactory, a solution would be then to ignore the domain sequences that are too close to any of the student sequences. A zone is therefore defined around each student sequence and domain sequences inside these zones are not considered (we present in the next section a way of implementing that procedure). Then by using the same process described in section 5.1, we select the closest sequence from the remaining ones. Figure 1 illustrates this selection.

The idea that learning is optimal when the stimuli is neither too close nor too far from the student's knowledge has been theorized by Vygotsky [12] with the notion of *zone of proximal development*. He influenced Krashen [7] who defined the *input hypothesis* as an explanation of how a second language is acquired: the learner improves his linguistic competence when he receives second language 'input' that is one step beyond his current stage of linguistic competence.

5.4 *Integrating the strategy into a language learning program*

We designed a program in C, based on the previous procedures. It is based on the result that most of the words we know, we learned from reading (evidence for that assertion is provided in [8]. Therefore, the goal is, at each step, to find the most appropriate English text for French students to read in order to stretch the student subspace

First, LSA is run to place all domain texts in a semantic space. The systems works in the following way: it selects dynamically a text, presents it to the student, tests his comprehension, then selects another text, etc. The process is initialized with a text the student provides (it will also work if the student provides no text ; it will just take more time to reach a correct behavior of the system). At the beginning, the system does not know much about the student and therefore, the selection of the next text might not be optimal. However, after a while the user model is more and more precise and the choice of the system more accurate.

After each text is provided, the student is required to rate his comprehension on a 1 to 5 scale. The text is then added to the student model as well as its weight. This is used to compute the proximity between a domain text and a student text: the similarity provided by LSA is multiplied by the weight. Therefore, texts that were well understood by the student play a more important role in the selection of the next text.

Improvements could be made by allowing the student to select words or parts of the texts that were not understood. This is going to be part of our future work.

6 A system to help learning kalah

In the same way, we designed a program to help a student learn an African game called kalah. This program, written in C, can be viewed as a tutor: it plays in such a way that the student is driven towards a state which should be optimal for his learning.

Kalah is played on a board composed of two rows of 6 pits. Each player owns a row of 6 pits as well as special pit called kalah. Pits contain initially 6 stones, and both kalahs are empty. Each player takes all the stones in any of his 6 pits, then spread them over the pits anticlockwise, one stone per pit, including his kalah (but not the opponent's kalah). If the last

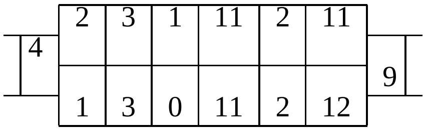

Figure 2: A state of the game of kalah

stone lands in the kalah, the player goes again. If the last stone lands in an empty pit, and the opponent has stones in the opposite pit, then all these stones go in the kalah. The goal is to get as many stones as possible in the kalah.

Lexemes are elements for describing a state. For instance, the lexeme `a3` indicates that there are 3 stones in the pit `a` (pits including kalahs are labelled from `a` to `n`).

A sequences of lexemes represents a state of the game. For instance the state shown in figure 2 is represented by the following sequence of lexemes: `a1 b3 c0 d11 e2 f12 g9 h11 i2 j11 k1 l3 m2 n4`.

A semantic space was built from 50,000 states automatically generated by two C programs using a traditional minmax algorithm at a depth of 3. This semantic space therefore covers a large part of the kalah semantics. Each time the student encounters a new configuration of the board, it is recorded into the space as a student entity.

The system plays against the student. At each turn, it looks for the different possible moves. Each one results in a new state, that is a new sequence of lexemes. The system looks for the new sequence of lexemes which is close enough to the student model but not too close (we rely on the same procedure described earlier). Then it plays the corresponding move. For instance, in the previous figure, there are 6 possible moves, therefore 6 possible new states (the system plays the upper row):

1. `a2 b3 c0 d11 e2 f12 g9 h11 i2 j11 k1 l3 m0 n5`
2. `a2 b3 c0 d11 e2 f12 g9 h11 i2 j11 k1 l0 m3 n5`
3. `a1 b3 c0 d11 e2 f12 g9 h11 i2 j11 k0 l4 m2 n4`
4. `a2 b4 c1 d12 e3 f13 g9 h12 i2 j0 k2 l4 m3 n5`
5. `a1 b3 c0 d11 e2 f12 g9 h11 i0 j12 k2 l3 m2 n4`
6. `a2 b4 c1 d12 e3 f12 g9 h0 i3 j12 k2 l4 m3 n5`

States 1 and 2 give the student the opportunity to play one more move and to put the last stone in an empty hole (which would allow him to capture the opposite stone). State 4 shows to the student 13 stones in a hole, which is a special number because after going round the board the last stone falls necessarily in an empty hole. State 5 gives the student the possibility to play again. States 3 and 6 have noting special.

According to the student model, state 5 is considered the most appropriate. Therefore move 5 will be played by the machine.

This system does not play optimally; indeed it sometimes plays badly since it is only concerned with driving the student towards an appropriate part of the semantics of the domain. As long as the student is aware that it plays against a player with an apparently random behavior, we believe that this is no problem.

7 Conclusion

In this paper, we relied on a high-dimensional representation of the lexemes of a domain to build the framework of a tutoring system. The goal of this tutoring system is to select the next stimuli to expose the student to in order for his learning to be optimal. Our approach is based on a model that has been validated in the field of cognitive psychology. We believe that many domains can be concerned with our approach as long as their examples can be expressed as sequences of lexemes. Our next task will be to design experiments with human subjects to develop more elaborate ways of selecting stimuli.

I thank Susan Dumais and the Bellcore Labs to have allowed me to use and hack the code of the basic LSA programs. I am also grateful to Erica de Vries and Philippe Dessus for their comments on previous versions of this article.

References

[1] C. Burgess and K. Lund, Modelling Parsing Constraints with High-dimensional Context Space, *Language and Cognitive Processes* 12(2/3) (1997) 177-210.

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshmann, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science* 41 (1990) 391:407.

[3] S.T. Dumais, Improving the retrieval of information from external sources, *Behavior Research Methods, Instruments, & Computers* 23(2) (1991) 229-236.

[4] S.T. Dumais, T.A. Letsche, M.L. Littman and T.K. Landauer, Automatic cross-language retrieval using Latent Semantic Indexing, In *AAAI Spring Symposium on Cross-Language Retrieval using Latent Semantic Indexing*, 1997.

[5] P.W. Foltz and S.T. Dumais, Personalized Information Delivery: An Analysis of Information Filtering Methods, *Communications of the ACM* 35(12) (1992) 51-60.

[6] P.W. Foltz, Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments, & Computers* 28(2) (1996) 197-202.

[7] S.D. Krashen, *Second Language Acquisition and Second Language Learning*, Prentice-Hall International, 1988.

[8] T.K. Landauer and S.T. Dumais, A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge, *Psychological Review* 104(2) (1997) 211-240.

[9] B. Lemaire, Models of High-dimensional Semantic Spaces, In *Proceedings of the 4th International Workshop on MultiStrategy Learning (MSL'98)*, June 1998.

[10] K. Lund and C. Burgess, Producing high-dimensional semantic spaces from lexical cooccurrence, *Behavior Research Method, Instruments, & Computers* 28(2) (1996) 203-208.

[11] B. Rehder, M.E. Schreiner, M.B. Wolfe, D. Laham, T.K. Landauer, and W. Kintsch, Using Latent Semantic Analysis to assess knowledge: Some technical considerations, *Discourse Processes* 25 (1998) 337-354.

[12] L.S. Vygotsky, *Thought and Language*, Cambridge, M.I.T. Press, 1962.

[13] E. Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufman, 1987.