

LSA-BASED SEMANTIC REPRESENTATION OF ACTION GAMES

Katia Lida Kermanidis and Kostas Anagnostou

Department of Informatics, Ionian University, 7 Tsirigoti Square, Corfu 49100, Greece

Keywords: Player modeling, Action games, Latent semantic analysis, Knowledge representation, Semantic similarity.

Abstract: Modeling the semantic space of a complex dynamic domain, like an action game, by automatically identifying the relations governing the game's concepts, entities, actions and other features, is a challenging research objective. In this paper we propose modeling the semantic space of the action game SpaceDebris, in order to identify semantic similarities between players' gaming styles. To this end we employ Latent Semantic Analysis and attempt to identify latent underlying semantic information governing the various gaming techniques. The several challenging research issues that arise when attempting to apply Latent Semantic Analysis to non-textual data describing a complex dynamic problem space (defining the semantic vocabulary and "word" utterances, deciding upon the dimensionality reduction rate, etc.) are addressed, and the framework of the proposed experimental setup is described. The extracted similarities are further employed for player modelling, i.e. grouping players according to their playing styles.

1 INTRODUCTION

Representing the knowledge of a specific domain, i.e. identifying the concepts that carry units of meaning related to it (domain "words"), as well as the semantic relations governing those concepts, is a wide and popular research area. Modeling domain knowledge is essential for developing expert systems, for intelligent prediction and decision making, for intelligent tutoring, user modeling, complex problem solving, reasoning etc. Mastering the semantics of a domain is to learn the "language" of the domain (Lemaire, 1998), i.e. to become exposed to various sequences of domain "words" in numerous contexts. This is similar to the way a foreign language learner learns vocabulary usage by reading, listening to, and writing texts in that language.

There are two possible ways for supplying domain knowledge (Lemaire, 1998): by hand, making use of domain experts' know-how, and automatically, by deriving the semantics from large corpora of "word" sequences. The first approach is more accurate, but domain-dependent, while the second is useful when no hand-crafted knowledge is available.

A widely used method for representing domain knowledge by statistical analysis of word usage is

Latent Semantic Analysis (LSA). LSA is adopted from the field of Information Retrieval (Landauer et al., 1998) and improves retrieval performance by taking into account automatically detected polysemy and synonymy relations between words. LSA identifies these underlying semantic relations by exploiting the occurrence statistics of the words throughout the document collection. By reducing the dimensionality of the initial term-document matrix (the matrix with rows representing index terms and columns representing documents; each cell contains the number of occurrences of a term in a document), hidden semantic similarities between words, between documents, and between words and documents surface, linking together words that may not even appear in the same document, or documents that may not share any common words.

LSA has been applied with significant success to other domains, like essay assessment in language learning (Haley et al., 2005), intelligent tutoring (Graesser et al., 2007), text cohesion measurement (McCarthy et al., 2007), summary evaluation (Steinberger and Jezek, 2004), text categorization (Nakov et al., 2003). Although all previously mentioned LSA applications have been performed on text corpora, some approaches have proposed its use in different non-textual knowledge domains like board game player modeling (Zampa and Lemaire,

2002), complex problem solving (Quesada et al., 2001), gene function prediction (Done et al., 2010; Dong et al., 2006; Ganapathiraju et al., 2005), web navigation behavior prediction (van Oostendorp and Juvina, 2007), collaborative filtering (Hofmann, 2004), semantic description of images (Basili et al., 2007).

In this paper a work in progress is described, that proposes (for the first time to the authors' knowledge) the application of LSA to a new domain, namely digital action games, in order to identify similarities among the playing techniques of various players. Action games have properties that resemble those of complex dynamic environments: causality relations (actions or decisions often affect subsequent actions or decisions), time dependence (the environmental circumstances that affect actions and decisions vary over time), and latent, implicit relations between domain properties that are not straightforward. Identifying the domain vocabulary, as well as well-formed sequences of "words" that constitute complete descriptions of actions or context conditions is of significant research interest.

Throughout the remainder of the paper we will address the research challenges that emerge when attempting to represent the semantics governing the SpaceDebris action shooting game (Anagnostou and Maragoudakis, 2009). The proposed use of the representation is player modeling: unsupervised grouping of players with similar gaming manners. Section 2 provides a bibliographic review of player modeling and categorization. Section 3 presents the basic properties of Latent Semantic Analysis, section 4 introduces the action game SpaceDebris, and finally section 5 describes the cognitive modeling process of the game domain, as well as its use for modeling players.

2 PLAYER MODELING

Several game designers have recently been shifting their focus to the player rather than the game itself. Numerous attempts have been made to identify the gaming technique of each player (e.g. (in)experienced, aggressive, tactical, action player), aiming to adapt the game features to his individual preferences and needs. By personalizing the features of the game, the designer hopes to provide increased satisfaction and entertainment.

Player modeling has been performed within an interactive storytelling game and the use of machine learning techniques (Thue et al., 2007; Roberts et al., 2007), by estimating the statistical behavior

(distribution) of player actions (Thawonmas and Ho, 2007), by using graphical knowledge representation schemata like influence diagrams (Shahine and Banerjee, 2007) and Bayesian networks (He et al., 2008). Further references to player modeling can be found in (Geisler, 2002). In (Anagnostou and Maragoudakis, 2009) SpaceDebris players are grouped into two clusters, using unsupervised learning, according to their playing style (aggressive or tactical).

Unlike previous approaches that either assign one of a set of predefined profiles to a player, or explore explicit actions and decisions made by the player, the present work proposes a knowledge model that attempts to

- identify the vocabulary of the game domain,
- represent complicated game states (action game states are hard to represent, as their definition is not straightforward like in board games), and
- detect hidden, underlying semantic relations between decisions made and actions taken and their context, as well as among domain "words".

3 LATENT SEMANTIC ANALYSIS

As mentioned earlier, LSA is a mathematical/statistical method initially proposed for reducing the size of the term-document matrix in information retrieval applications, as the number of lexicon entries may reach several thousand, and the document collection may contain tens of thousands of documents or more. LSA achieves dimensionality reduction through Singular Value Decomposition (SVD) of the term-document matrix. SVD decomposes the initial matrix A into a product of three matrices and "transfers" matrix A into a new semantic space:

$$A = T S D^T \quad (1)$$

T is the matrix with rows the lexicon terms, and columns the dimensions of the new semantic space. The columns of D represent the initial documents and its rows the new dimensions, while S is a diagonal matrix containing the singular values of A . Multiplication of the three matrices will reconstruct the initial matrix. The product can be computed in such a way that the singular values are positioned in S in descending order. The smaller the singular value, the less it affects the product outcome. By maintaining only the first few of the singular values

and setting the remaining ones to zero, and calculating the resulting product, the initial matrix may be approximated as a least-squares best fit. The dimensions of the new matrix are reduced and equal to the number of selected singular values.

As an interesting side effect, dimensionality reduction reduces or increases the frequency of words in certain documents, or may even set the occurrence of words to higher than zero for documents that they initially did not appear in. Thereby semantic relations between words and documents are revealed that were not apparent at first (latent). It needs to be noted that LSA is fully automatic, i.e. the latent semantic relations are learned in an unsupervised manner. Another significant property is that LSA does not take into account the ordering of words within their context; documents are considered “bags of words”. Extensive information on LSA can be found in (Landauer et al., 1998).

4 SPACEDEBRIS

The videogame used for the purposes of data collection is based on SpaceDebris (Anagnostou and Maragoudakis, 2009). The action takes place within the confines of a single screen, with alien ships scrolling downwards. There are two types of enemy spaceships (next referred to as *enemy 1* and *enemy 2*), the carrier which is slow and can withstand more laser blasts, and a fighter which is fast and easier to destroy. The player wins when he has successfully withstood the enemy ship waves for a predetermined time. The game environment is littered with floating asteroids which in their default state do not interact (i.e. collide) with any of the game spaceships. In order to do so, an asteroid has to be “energized” (hit by player weapon). Also floating are shield and life power-ups which the user can use to replenish his ship’s shield and remaining lives. The player’s ship is equipped with a laser cannon which he can use to shoot alien ships. The laser canon is weak and about 4-5 successful shots are required to destroy an enemy ship (except for the boss which requires many more). The laser can also be used to “energize” an asteroid and guide it to destroy an enemy ship.

5 MODELING SPACEDEBRIS

Several research challenges need to be addressed

when attempting to model the domain of an action game like SpaceDebris using LSA.

5.1 Vocabulary Identification

In board-like games, like tic-tac-toe or chess, domain “words” are easy to identify. Boards may be viewed as grids of cells and each cell state (e.g. “X”, “O” or empty in tic-tac-toe) constitutes a “word” (Lemaire, 1998). In action video games “words” are harder to identify. Should they represent player actions, enemy actions, the state of the context, scoring results, spare lives or ammunition, time parameters? In the firefighting microworld of (Quesada et al., 2001) “words” are actions like appliance moves, or water drops. The definition of a game “word” depends on the intended use of the model. If the intended use is behavior prediction, a “word” needs to model a player’s action, as the player’s sequence of actions (in a given context) defines his behavior.

Table 1: The total number of distinct cell states.

Distinct cell states
The cell contains an asteroid
The cell contains an “energized” asteroid
The cell contains the player’s ship
The cell contains the player’s ship being hit by enemy 1
The cell contains the player’s ship being hit by enemy 2
The cell contains the player’s ship being destroyed
The cell contains the player’s ship firing a laser
The cell contains enemy 1
The cell contains enemy 1 being hit by a laser
The cell contains enemy 1 being hit by an asteroid
The cell contains enemy 1 firing a laser
The cell contains enemy 1 being destroyed
The cell contains enemy 2
The cell contains enemy 2 being hit by a laser
The cell contains enemy 2 being hit by an asteroid
The cell contains enemy 2 firing a laser
The cell contains enemy 2 being destroyed
The cell contains a player laser
The cell contains an enemy 1 laser
The cell contains an enemy 2 laser
The cell contains a life upgrade
The cell contains a life upgrade hit by laser
The cell contains a shield upgrade
The cell contains a shield upgrade hit by laser
Empty cell

In the present work, two approaches to representing “words” are considered. In the first approach, the game terrain is considered a grid, and a “word” is formed by two parts. The first part is the string that derives from the concatenation of the states of each cell in the grid. The state of each cell is determined by several factors, depending on the state of each game entity. Table 1 shows all 25

distinct cell states. A cell might also be in a state that combines a number of states such as those described.

The second part models further out-of-the-grid (non-spatial) information, like score, number of available life upgrades, number of available shield upgrades. The values of these features are concatenated to form an out-of-the-grid string, that is then attached to the cell-states string to constitute a complete “word”.

In this “grid” representation, player or enemy actions are modelled implicitly (indirectly) through the related cell states. For example, a player laser appearing in the cell right above the player ship indicates that the ship fired in the recent game history.

The grid cell size is of importance, as it affects the level of granularity. The smaller the cell size is, the more “generic” the “words” are, and the fewer the combinations of states each cell may appear in. We will experiment with grid sizes 11x8 and 12x6, the first corresponding to a cell size equal to the player ship’s size and the second to a cell size equal to the largest enemy ship size, with a screen resolution of 1024x768 pixels.

Vocabulary size using this representation reaches 2212 with a grid size of 11x8 and 1728 with a grid size of 12x6. Vocabulary size is important, as too many “words” may result to too few co occurrences and LSA will not work. On the other hand, too small a vocabulary may lead to too few similarities and, again, the method will not work (Lemaire, 1998). Optimal vocabulary size is an open research issue and depends on the domain.

The second approach to defining the vocabulary is more “holistic” and resembles in part that of (Quesada et al., 2001). Each “word” represents a player action, like *move to a location* or *fire*. However, unlike (Quesada et al., 2001), each action in a “word” is accompanied by a concatenation of features that represent the state of the context in which the action took place. Thereby causality relations (the reasoning behind the player’s action) are clearly identifiable. The context features taken into account are

- the number of enemies very close to the player (denoting imminent threat)
- the number of enemies close to the player (denoting danger)
- the total number of enemies on screen
- the number of player lasers fired
- the number of enemy 1 lasers fired
- the number of enemy 2 lasers fired
- the position of the player
- the number of life upgrades performed

- the number of shield upgrades performed
- the number of hit asteroids
- the number of visible asteroids
- the number of hit enemy 1 ships
- the number of hit enemy 2 ships
- the score value
- the number of the player’s available life upgrades
- the number of shields available to the player

“Word” examples using an NxM grid (example 1) and the “holistic” (example 2) approach are shown below. The first part (up to X_{NM}) of the string in example 1 consists of tokens that stand for each cell state (tokens are concatenated together with underscores). The second part (after X_{NM}) encodes out-of-the-grid information, as explained earlier. We use 16bit numbers, to denote the presence (1 or 0) of one of the 9 game entities (player, 2 types of enemies, 3 types of lasers, 2 types of upgrades, asteroid). So, each “word” is a sequence of numbers that describe the state of the corresponding cell, while the last three tokens stand for the score, the number of spare lives and spare shields respectively. In this example the first cell contains a life upgrade, the second an asteroid, the third is empty etc. In example 2 the first token is the player’s action (the player moves to location with coordinates (-286, -133)). Each of the following concatenated tokens is a value for each of the features listed above (e.g. 1 enemy is very close, 3 are close, there are 9 enemies on-screen, player has fired a laser, enemies have fired 3 lasers etc.).

2_1_0... X_{NM} _1000_3_100 (ex. 1)

move-286-133_1_3_9_1_3_... X (ex. 2)

The “grid” representation takes into account long-distance semantic dependencies, i.e. the semantics of each cell (no matter how distant) participates in the domain knowledge. The “holistic” representation, on the other hand, detects causality relations between the environment and the player’s reaction to it in a more straightforward way, while the “grid” approach “mines” these causality relations implicitly.

5.2 Game Session Representation

Game sessions play the role of documents in Information Retrieval. As documents are sequences of words that convey a specific meaning and are considered to satisfy a certain information need, game sessions are well-formed sequences of

“words” in the game domain. Each sequence describes a path to a goal: the end of a game. Each “word” constitutes a complete description of a player’s action or of a description of the context (game environment) at a given moment.

One way to represent a game session is to take a sample of the game state at constant pre-defined time intervals (e.g. every 500 msecs) and register the sequence of “words” (“words” are defined using either the grid or the holistic approach) that describe the sample. Each sample represents a game state at the specified time point. The duration of the sampling time interval is very important. Small intervals may lead to consecutive states that are semantically identical (i.e. the player has not had enough time to make a decision or act, or the state of the context has not changed). Long intervals may lead to the loss of semantic information (i.e. player’s actions that occurred between the samples may be missed). We will experiment with various interval sizes in order to find the “optimal” sampling rate.

Another way to represent game sessions is through sampling events that are dynamically triggered by player’s actions. Instead of sampling with a static rate, sampling may be event-driven. Every time the player acts, a game state sample is taken, and the player’s action and game context are recorded. Event-driven sampling will record information that is more related to the player’s actions and disregard irrelevant and superfluous semantic information that is not important for the goal of the session.

5.3 Reduction Rate

The rows of the resulting term-document matrix represent the “words”, and the columns represent game sessions. Each cell contains the frequency of occurrence of the “word” in the row in the column session. Applying LSA to the matrix, another research question arises: What is the optimal number of singular values that should be maintained? In Information Retrieval the number of dimensions of the latent semantic space is usually between 100 and 300 (Lemaire, 1998). More research work needs to be done in order to determine the appropriate number of dimensions when it comes to non-textual domains. Our proposal includes the experimentation with various dimension numbers and the research of their impact on modeling performance.

5.4 Experimental Setup for Measuring Semantic Similarity

As mentioned earlier, the extracted model will be used for identifying similar gaming techniques among players. A group of players will play the game for a given time frame. Players will at first be asked to familiarize themselves with the game by playing off the record for 4-5 minutes. After this introductory phase, game sessions will be recorded for every player. Each game session lasts an average of 3 minutes, and players will be asked to complete a specific number of games. The number of games needed for successfully identifying the player’s gaming style will be experimentally explored. Each game session will constitute a feature vector, which is formed by the set of “words” representing it. Feature vectors both before and after LSA will be stored for comparative analysis of results.

To identify similar gaming techniques, the distance between vectors needs to be computed. Though several distance metrics have been experimented with, pairwise cosine similarity is the most popular measure (Lemaire, 1998). Cosine similarity will link the most semantically similar vectors together, forming clusters of similar gaming techniques. Clustering evaluation may be performed in two ways. Players may be asked to answer a short questionnaire before playing, where they will characterize their individual gaming style, choosing one or more from a set of pre-defined styles. Another way is to ask a game expert to identify the style of each individual player by looking at his actions and decisions throughout the game sessions. The matching degree of the cosine similarity and the expert’s decision (and/or the player’s questionnaire answers) will be measured before and after applying LSA, for detecting its impact.

6 CONCLUSIONS

In this paper we have described a proposal for modeling, in a novel way, the semantic space of a complex non-textual problem, i.e. an action game, using LSA. While the application of LSA to textual data is fairly straightforward, several research issues arise when the data involved are not textual, but represent players’ actions and environmental (contextual) conditions. These research issues have been addressed and an experimental setup has been proposed for the novel use of the extracted model to player modeling.

REFERENCES

- Anagnostou, K., Maragoudakis, M., 2009. Data mining for player modeling in videogames. In *Panhellenic Conference on Informatics (PCI)*. Corfu, Greece.
- Basili, R., Petitti, R., Saracino, D., 2007. LSA-based automatic acquisition of semantic image descriptions. In *Conference on Semantics and Digital Media Technologies*, LNCS 4816, pp. 41-55.
- Done, B., Khatri, P., Done, A., Draghici, S., 2010. Predicting novel human gene ontology annotations using semantic analysis. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7(1), pp. 91-99.
- Dong, Q., Wang, X., Lin, L., 2006. Application of latent semantic analysis to protein remote homology detection. In *Bioinformatics*, vol. 22(3), pp. 285-290, Oxford University Press.
- Ganapathiraju, M., Balakrishnan, N., Reddy, R., Klein-Seetharaman, J., 2005. Computational biology and language. In *Ambient Intelligence for Scientific Discovery*, LNAI 3345, pp. 25-47.
- Geisler, B., 2002. An empirical study of machine learning algorithms applied to modeling player behavior in a first person shooter video game. MSc Thesis, University of Wisconsin-Madison.
- Graesser, A. C., Penumatsa, P., Ventura, M., Cai, Z., Hu, X., 2007. Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language. *Handbook of Latent Semantic Analysis*, T. Landauer, D. McNamara, S. Dennis, W. Kintsch Eds.
- Haley, D. T., Thomas, P., de Roeck, A., Petre, M., 2005. A research taxonomy for latent semantic analysis-based educational applications. In *Conference on Recent Advances in Natural Language Processing*. Borovets, Bulgaria.
- He, S., Du, J., Chen, H., Meng, J., Zhu, Q., 2008. Strategy-based player modeling during interactive entertainment sessions by using Bayesian classification. In *4th International Conference on Natural Computation (ICNC)*, pp.255-261.
- Hofmann, T., 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, vol. 22(1), pp. 89-115.
- Landauer, T., Foltz, P., Laham, D., 1998. An introduction to latent semantic analysis. In *Discourse Processes*, vol. 25, pp. 259-284.
- Lemaire, B., 1998. Models of high-dimensional semantic spaces. In *4th International Workshop on Multistrategy Learning (MSL 98)*.
- McCarthy, P. M., Briner, S., Rus, V., McNamara, D., 2007. Textual signatures: identifying text-types using latent semantic analysis to measure the cohesion of text structures, In A. Kao and S. Poteet (Eds), *Natural Language Processing and Text Mining*, pp. 107-122.
- Nakov, P., Valchanova, E., Angelova, G., 2003. Towards deeper understanding of the LSA performance. In *Conference on Recent Advances in Natural Language Processing*.
- Quesada, J. F., Kintsch, W., 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 J. J. Cañas (Ed.), *Cognitive Research with Microworlds*, pp. 117-131, Granada, Spain.
- Roberts, D., Riedl, M., Isbell, C., 2007. Opportunities for machine learning to impact interactive narrative. In *Workshop on Machine Learning and Games at NIPS*.
- Shahine, G., Banerjee, B., 2007. Player modeling using knowledge transfer. In *EUROSIS GAMEON-NA Conference*, pp. 82-89. Gainesville, Florida, USA.
- Steinberger, J., Jezek, K., 2004. Using latent semantic analysis in text summarization and summary evaluation. In *Conference on Information Systems, Implementation and Modeling (ISIM)*, pp. 93-100.
- Thawonmas, R., Ho, J., 2007. Classification of online game players using action transition probability and Kullback Leibler entropy. In *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Special issue on Advances in Intelligent Data Processing, vol. 11(3), pp. 319-326.
- Thue, D., Bulitko, V., Spetch, M., Wasylshen, E., 2007. Interactive storytelling: a player modelling approach. In *3rd International Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pp. 43-48, Stanford, California, USA.
- van Oostendorp, H., Juvina, I., 2007. Using a cognitive model to generate web navigation support. In *International Journal of Human-Computer Studies*, vol. 65(10), pp. 887-897, Elsevier.
- Zampa, V., Lemaire, B., 2002. Latent semantic analysis for user modeling. In *Journal of Intelligent Information Systems*, vol. 18, pp. 15-30, Kluwer Academic Publishers.