Semantic texual Similarity Analysis Based Recommender System with Deep Learning

by

Mingzhe Du

Ph.D. Proposal in

Computer Science and Engineering

The College of Engineering and Computing

University of South Carolina

2016

Dr. Jose M. Vidal, Major Professor

Dr. Maton M. Matthews, Committee Member

Dr. Jianjun Hu, Committee Member

Dr. Rose?, Committee Member

Dr. Barry Markovsky, Committee Member

CHAPTER 1

Introduction

With the continuous expansion of Internet activities and online merchandise, recommender systems play an increasingly critical role in the interactive Internet environment[1-5]. Recommender systems apply data analysis in order to help users find their most wanted products from online merchandise sites. For instance, a personalized recommender system on Amazon (www.amazon.com) suggests music and books to customers based on the user's personal shopping experience, hobbies and areas of concern. Non-personalized recommender systems, like Zagat(www.zagat.com) and Yelp(www.yelp.com), provide a general restaurant guide based on the input of millions of individuals. The same reviews and rating scores are presented to users no matter who is looking up their sites.

In our research, we have a system with a lexicon which consists of words and definitions. One word may have multiple definitions. The system recommends pre-defined existing {term: definition} pairs when a user tries to add a new {term: definition} pair to the lexicon. The system consists of a repository of sociology terms while trying to keep a minimal lexicon and eliminating redundant definitions. Instead of using reviews and ratings data, our recommendation is based on the semantic textual analysis of definitions. In recent years, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Tree-Structured LSTM, Deep Structured Semantic Model (DSSM) and some similar deep learning frameworks have been used to compute semantic similarity between two text snippets[6-11].

# Add a background story of wikitheoria

We propose to develop an algorithm that will calculate and output one or more formal representations of an agent’s behavior, similar to a Markov decision process, but with support for cycles, as needed, from the collected information, as well as a tool that will easily allow the alteration of each decision process to meet the requirements of the modeler. This tool will then produce a file in a standard format that can be efficiently read and used in a variety of applications, in order to create initial decision processes for believable behaviors.

CHAPTER 2

Related Work

Many natural language processing (NLP) applications and recommender systems such as paraphrase recognition (Dolan et al., 2004), automatic machine translation evaluation (Kauchak and Barzilay, 2006), textual summarization (Aliguliyev, 2009), tweets search (Sriram et al., 2010), student answer assessment (Rus and Lintean, 2012; Niraula et al., 2013) and recommender synonymy challenge (wikipedia page) are constrained by the effectiveness of semantic textual similarity (STS) analysis.

There are three classes of models where real-valued vectors are used to represent the meaning of phrases and sentences: bag-of-words models, sequence models, and tree-structured models. In bag-of-words models, the representation of a sentence is independent of word order (Landauer and Dumais, 1997; Foltz et al., 1998). Sequential models construct the sentence representation as an order-sensitive sequence (Elamn, 1990; Miolov, 2012). Tree-structured models compose each sentence representation from its constituent sub-phrases according to a given syntactic structure over the sentence (Goller and Kuchler, 1996; Socher et al., 2011).

Considering the importance of capturing semantic difference of word sequence (e.g.,"Cat eats fish" vs. "Fish eats cat"), the order-sensitive sequential models or tree-structured models are better sentence representations due to their relation to syntactic interpretation of sentence structure[7-8]. Recurrent neural network is an important type of deep learning framework, which is designed for sequence problems[6]. Due to its capability for processing arbitrary length sequences, RNNs are a natural choice for sequence modeling tasks.

In this project, we will explore the Sentences Involving Compositional Knowledge (SICK) dataset (Marelli et al., 2014) using deep learning with GPU acceleration. The dataset consists of 9927 sentence pairs in a 4500/500/4927 train/dev/test split. Each sentence pair is annotated with a relatedness score which was assigned by different human annotators. The training of the RNN with massive data and deep learning has high computational intensity. Thus, it is critical to increase the computation speed for our semantic similarity based recommender system.

Graphic Processing Unit (GPU) can significantly increase the computational power in the RNN training process. Several available deep learning platforms including Theano and Tensorflow support NVIDIA GPU wit CUDA and increase the computation performance significantly over CPU-only mode.

Our research could broadly impact in multiple fields, including sociology, computer science and linguistics. We expect our research outcome draws the attraction from the scientific community to emphasize the application of deep learning with GPU-acceleration in this field.

CHAPTER 3

Proposed Research

Thesis Goal:

Our proposed research has three stages: 1) In the first stage, we plan to experiment with RNNs and sequential LSTM models. Two commonly-used sequential LSTMs are the Bidirectional LSTM and the Multilayer LSTM. In this stage, we want to provide the effectiveness of RNN, and we also hope the features can be located more accurately comparing to Bag-of-words approaches; 2) A limitation of the LSTM is that they only allow for strictly sequential information propagation. The tree structured LSTM structure is believed to allow for richer network topologies where each LSTM is able to incorporate information from multiple child units. In the second stage, the research will be revealing the mechanism of tree structured LSTM with standard LSTM unit on gating vectors and memory cell updates. This allows a Tree-LSTM model learn to emphasize semantic heads in a semantic relatedness task[9-11]. 3) In the third stage, we will apply the algorithms developed from first two stages to our non-personalized recommender system. During the integration testing period, the performance of these two stages will be evaluated.

Currently, we are using the Theano[13], a deep learning framework[12], as the backend. On top of Theano, we are using the Keras, a high-level neural networks library, written in Python, to run Convolutional Neural Network[14] and Recurrent Neural Network algorithms on CPU only. As described in the official document, with GPU, Theano performs data-intensive calculations up to 140x faster than with CPU. We will install the proper program to accelerate the computation using the request GPU.

In the future, we will also work on Tensorflow[15], which is another popular deep learning framework. Tensorflow also recommends using GPU to accelerate the computation speed.

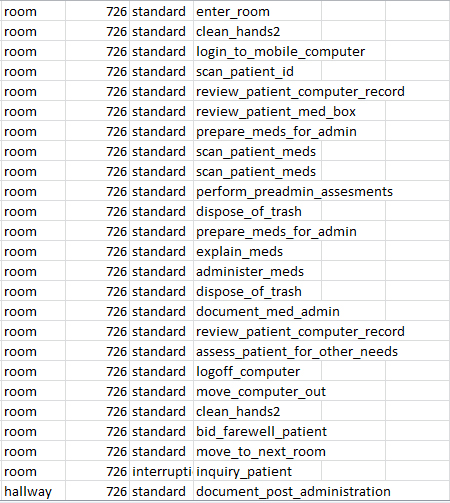
Plan Of Work:

To reach our goal of creating one or more believable agents based on the automated analysis of sequential behavior, we have the following projects:

The MAGIC (Models Automatically Generated from Information Collected) Algorithm:

The data we are using to create the decision process for each agent is sequential data. Therefore, a variation of the Markov Decision Process, which we will refer to as a Sequential Compressed Markov Decision Process, or SCMDP, appears to be the best method, as the MDP is inherently sequential, as shown by Papadimitriou and Tsitsiklis (1987). To create the SCMDP, the collected data must be analyzed statistically. Calculating this by hand can be difficult and time-consuming, particularly if the expert that requires the model is not the model’s creator. The MAGIC algorithm automates sequential data analysis and creates an SCMDP for the agent, allowing the modeler to focus on the implementation of the behaviors themselves rather than the creation of the actual decision process.

In the MAGIC algorithm, the sequential data is read from a text file as strings of behaviors, or sequences of behavior observations, as shown in Figure 1 below, taken from the nursing administration simulation:



**Figure 1: Data Used in the Nursing Administration Simulation**

These behaviors were observed by students following nurses during their rounds over a period of three months and recorded using an iPad app. The results were combined into one comma delimited text file. The start and end of each sequence is determined by the associated room number. The behavior string is made up of an ordered list of the behaviors in a given sequence.

Because the behavior strings are of finite length, and some behaviors are unlikely to occur first or last, start and end states are added to each behavior string. Groups (or cycles) of behaviors that occur together repeatedly are regarded as one behavior state, and the string is changed accordingly to avoid miscalculation of transitional probabilities. Each behavior is regarded as a state, and the transitional probabilities from one state to the next are calculated using the Markov assumption of conditional independence. Cycle checking and transitional probability calculations are done using a dynamic programming approach, and a file is then output with the SCMDP states and transitional probabilities written in a standardized format that we will refer to as ABML, or Agent-Based Modeling Language. The specifics of ABML have yet to be determined and form part of the work proposed for this thesis. The current version of the algorithm is in Figure 2, below:

MAGIC (File filename, Int maxTaskRepeat)

Open File filename

Create List of taskNames

Read File filename

ForEach task in File filename:

If task is not in List of taskNames, add task to List of taskNames

Create List of behaviorStrings

While not end of file:

Read task

If task is first in behavior string, add “start” to behavior String

Add task to behaviorString

If task is last in behaviorString, add “end” to behaviorString

Create list of cycles

Foreach behaviorString in behaviorStrings:

Foreach cycle in cycles:

If behaviorString contains cycle:

Replace cycle in behaviorString with “CYCLE” + cycle number

substringLength ←2

While substringLength <= maxTaskRepeat:

i←0

while i<length(behaviorString)-substringLength-1

substring ← behaviorString[i:substringLength-1]

if substring contains “CYCLE”:

i←i+1

continue

count ← number of times behaviorString contains substring

if count > 2:

add substring to List of cycles

replace substring in behaviorString with “CYCLE” + cycle number

i←i+1

substringLength←substringLength + 1

Create SCMDP[state][transitionState] where the number of states and transitionStates are the number of tasks in the taskList + the number of cycles in the cycleList

Foreach behaviorString in behaviorStrings:

Foreach behavior in behaviorString:

Add 1 to the corresponding transitionState

Foreach state in SCMDP:

total ← 0

Foreach transitionState in state:

total←total + transitionState

Foreach transitionState in state:

transitionState←transitionState / total

Print SCMDP to text file

**Figure 2: MAGIC Algorithm**

The current output of the AMBL file follows the format:

[ “State name” “State to transition to” Transitional probability “State to transition to”

Transitional probability …]

The following in Figure 3 is an example of the current AMBL text output using the MAGIC algorithm and the data from the Nursing Simulation:

["other\_care" "other\_care" 0.048387 "close\_med\_box" 0.123656 "document\_post\_administration" 0.166667 "document\_med\_admin" 0.193548 "Cycle0" 0.198925 "Cycle3" 0.220430 "Cycle5" 0.258065 "login\_to\_mobile\_computer" 0.279570 "move\_computer\_out" 0.301075 "Cycle19" 0.322581 "Cycle10" 0.338710 "Cycle16" 0.344086 "Cycle15" 0.365591 "scan\_patient\_meds" 0.370968 "scan\_patient\_id" 0.430108 "perform\_assessment" 0.526882 "obtain\_meds\_pyxis" 0.537634 "review\_patient\_computer\_record" 0.580645 "explain\_meds" 0.586022 "dispose\_of\_trash" 0.596774 "review\_patient\_med\_box" 0.688172 "move\_to\_next\_room" 0.704301 "review\_information" 0.709677 "setup\_for\_med\_admin" 0.720430 "Cycle25" 0.725806 "bid\_farewell\_patient" 0.806452 "Cycle24" 0.822581 "clean\_equipment" 0.827957 "assess\_patient\_for\_other\_needs" 0.865591 "administer\_meds" 0.892473 "clean\_hands" 0.967742 "prepare\_meds\_for\_admin" 1.000000]

["close\_med\_box" "other\_care" 0.064935 "close\_med\_box" 0.069264 "obtain\_meds\_medroom" 0.082251 "document\_post\_administration" 0.134199 "document\_med\_admin" 0.242424 "prepare\_meds\_for\_admin" 0.264069 "Cycle3" 0.268398 "Cycle5" 0.311688 "login\_to\_mobile\_computer" 0.316017 "move\_computer\_out" 0.341991 "Cycle19" 0.354978 "Cycle12" 0.359307 "Cycle14" 0.367965 "Cycle15" 0.385281 "scan\_patient\_meds" 0.402597 "other" 0.419913 "scan\_patient\_id" 0.428571 "perform\_assessment" 0.571429 "obtain\_meds\_pyxis" 0.601732 "review\_patient\_computer\_record" 0.645022 "explain\_meds" 0.658009 "logoff\_computer" 0.662338 "review\_patient\_med\_box" 0.692641 "review\_information" 0.696970 "put\_on\_gloves" 0.714286 "obtain\_meds\_outside" 0.718615 "Cycle22" 0.731602 "Cycle25" 0.740260 "bid\_farewell\_patient" 0.787879 "clean\_equipment" 0.805195 "access\_computer" 0.809524 "assess\_patient\_for\_other\_needs" 0.848485 "administer\_meds" 0.926407 "clean\_hands" 0.987013 "setup\_for\_med\_admin" 1.000000]

["scan\_nurse\_badge" "review\_patient\_computer\_record" 0.090909 "perform\_assessment" 0.181818 "perform\_preadmin\_assesments" 0.272727 "Cycle6" 0.363636 "scan\_patient\_id" 0.454545 "enter\_room" 0.545455 "login\_to\_mobile\_computer" 1.000000]

**Figure 3: Data Produced Using the MAGIC Algorithm**

The MAGIC BAG (MAGIC Behavior Adjustment Graph) tool:

The MAGIC BAG tool will be created in NetLogo, which provides a simple interface that is easy for end-users to understand. The MAGIC BAG loads and parses a text file written in ABML and turns it into a graph, allowing the SCMDP’s behavior states and transitional probabilities to be adjusted more easily by those who are less familiar with programming by using buttons and simple drag-and-drop commands. It then outputs a file in ABML to be used in an actual simulation. A screenshot of the MAGIC BAG tool in its current state is shown in Figure 4 below:

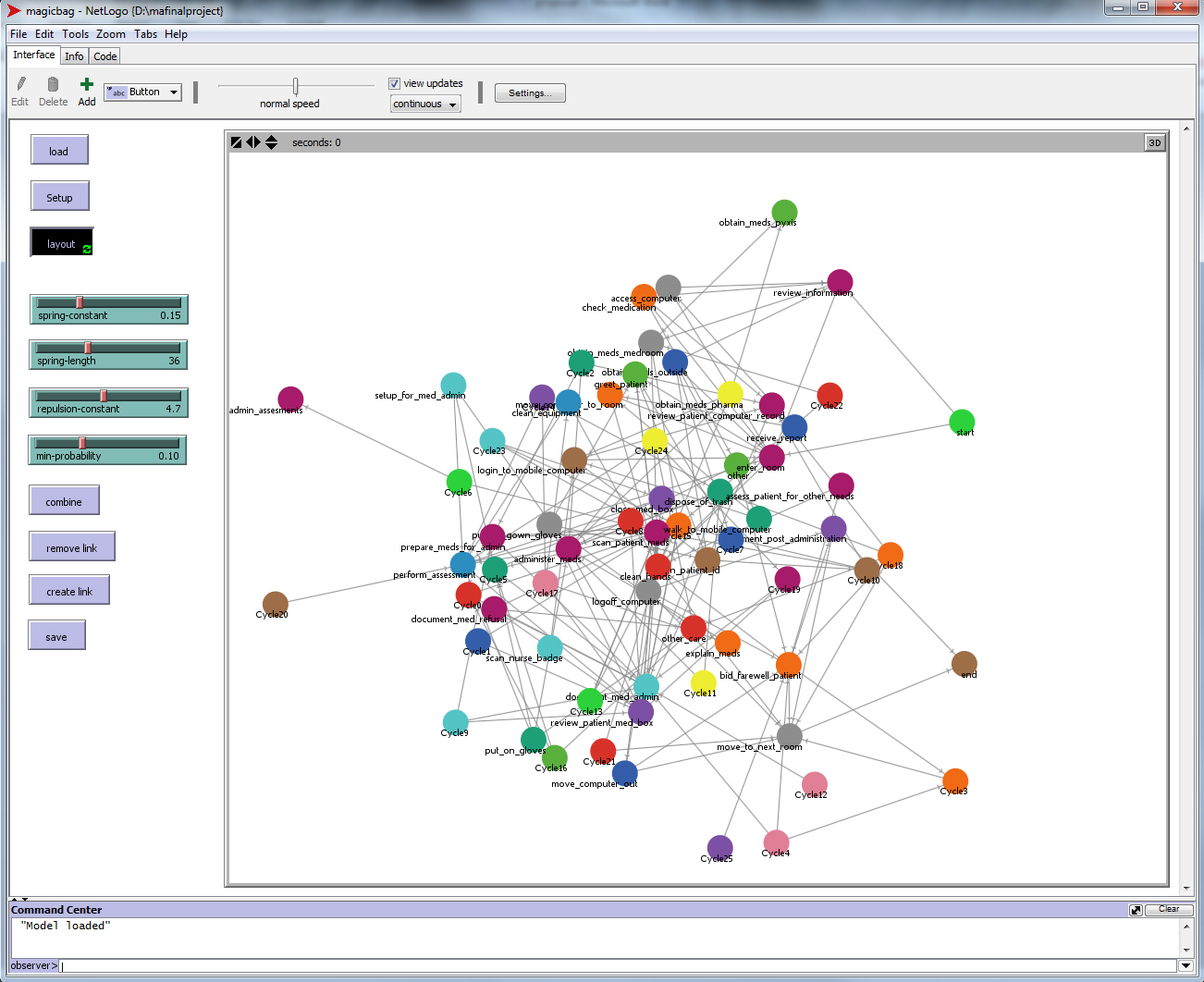


Figure 4: The MAGICBAG Tool

This tool currently allows end-users to adjust the minimum probability for a transition to be allowed, eliminating any transitions under the minimum. It also allows the creation of new transitions between states, direct elimination of transitions, and the combination of two states into one. The resulting graph can then be saved as a text file.

The Multi-MAGIC algorithm:

There are times when one collection of behavior sequence data incorporates the behaviors of individuals with different behavior styles or roles, and it would be good to have more than one SCMDP+ in order to distinguish between those roles. The Multi-MAGIC+ algorithm allows the user to specify the number of SCMDP’s to be produced by the algorithm. It uses compression on the behavior strings, removing behaviors that are repeated in sequence in order to create the most compact version of the workflow. It then creates a dummy workflow and uses edit distance (Liu et al., 2007) with the k-nearest neighbor method in order to cluster the behavior sequences. Each cluster is then processed as a separate SCMDP.

Test Phase One:

A series of approximately 100 random finite strings of characters from 5 to 15 characters in length will be generated and saved as a text file, as shown in Figure 5 below.

f,a,c,c,b,b

a,a,d,d,e,b,a,f,a,a,c,b

c,b,f,b,f,f,b,d,a,c

e,a,a,b,d,b,a,c,d,a,b,d,b

b,a,f,a,e,a,a,e,b

e,b,e,c,b,f,a

c,e,e,d,e,e,d,f,a,b,d,e,c,a,c

c,d,e,a,f,c,c,a

e,b,e,c,d,e,b,b,f,d

b,d,f,f,c,f,e

d,e,c,d,f,f,a,a,e,c,d,f,e,e,a

f,a,f,c,b,f,e,d,d

b,e,c,f,d,d,a,c,f,c,e,c

d,d,f,c,e

**Figure 5: Randomly generated character strings for Test One**

These characters will be used to represent observed behaviors. The MAGIC algorithm will then be used to create an SCMDP from this text file. The resulting SCMDP will be checked against the initial strings of characters to see if they can be reproduced using that SCMDP. The number of strings that can be reproduced will be divided by the total number of strings to give a reliability percentage for the model.

Test Phase Two:

Simulation One: Nursing Administration

The nursing administration simulation focuses on nursing behavior in a hospital setting. Data is obtained by the observation of nurses during their rounds. The simulation is a 3D version of an actual hospital floor, built using the hospital’s floor map. Each room contains only one patient. The patient has an attached script with variables for the number of medications that the patient requires, whether or not the patient requires special medications, and whether or not the patient requires some other type of care. The nurse agent uses an SCMDP logic controller to determine the most appropriate behavior given the available data.

Simulation Two: Canine Social Interaction

The canine social interaction simulation will focus on interaction between domestic dogs in a dog park setting. Data will be obtained from observations made by animal behavioral scientists. These observations will be analyzed using the Multi-MAGIC algorithm to create different styles of canine behavior. The resulting canine agent interaction will then be observed and tested for believability.

Simulation Three: Gamer Behavior

A simple shooter-style game will be created where all player actions are logged in a text file. A group of approximately 10 people will play the game, and their actions will be analyzed using the MAGIC algorithm. The resulting SCMDP will be used to create a bot that will imitate the play style of the human players. The human and bot games will both be recorded, and a different group of approximately 10 people will be asked to determine which is the player and which is the bot.

APPROXIMATE TIMELINE

* Design MAGIC Algorithm: mostly completed (completion by Dec 2013)
* Check internal validity: partially completed (completion by Dec 2013)
* Create MAGIC BAG NetLogo tool: partially completed (completion by Dec 2013)
* Create example simulations: one created but not fully implemented, plans for 2 more (completion by May 2015)
* Design of Multi-MAGIC Algorithm (completion by May 2015)

References

Adobbati, Rogelio, Andrew N. Marshall, Andrew Scholer, Sheila Tejada, Gal A. Kaminka, Steven Schaffer, and Chris Sollitto. "Gamebots: A 3d virtual world test-bed for multi-agent research." In *Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, vol. 5. Montreal, Canada, 2001.

Bechara, Antoine, Hanna Damasio, and Antonio R. Damasio. "Emotion, decision making and the orbitofrontal cortex." *Cerebral cortex* 10, no. 3 (2000): 295-307.

Bell, William H., David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger, and Floriano Zini. "Evaluation of an economy-based file replication strategy for a data grid." In *Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on*, pp. 661-668. IEEE, 2003.

Berger, Thomas. "Agent‐based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis." *Agricultural economics* 25, no. 2‐3 (2001): 245-260.

Bonabeau, Eric. "Agent-based modeling: Methods and techniques for simulating human systems." *Proceedings of the National Academy of Sciences of the United States of America 99*, no. Suppl 3 (2002): 7280-7287.

Cohen, Philip R., and Hector J. Levesque. "Intention is choice with commitment." *Artificial Intelligence* 42, no. 2 (1990): 213-261.

Cook, Diane J., Michael Youngblood, Edwin O. Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. "MavHome: An agent-based smart home." In *Pervasive Computing and Communications, 2003.(PerCom 2003).* *Proceedings of the First IEEE International Conference on*, pp. 521-524. IEEE, 2003.

Deadman, Peter, Derek Robinson, Emilio Moran, and Eduardo Brondizio. "Colonist household decisionmaking and land-use change in the Amazon Rainforest: an agent-based simulation." *Environment and Planning B* 31 (2004): 693-710.

Duffy, John. "Agent-based models and human subject experiments." *Handbook of computational economics* 2 (2006): 949-1011.

Garson, G. David. "Computerized Simulation in the Social Sciences A Survey and Evaluation." *Simulation and Gaming* 40, no. 2 (2009): 267-279.

Georgeff, Michael, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. "The belief-desire-intention model of agency." In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pp. 1-10. Springer Berlin Heidelberg, 1999.

Gigerenzer, Gerd, and Wolfgang Gaissmaier. "Heuristic decision making." *Annual review of psychology* 62 (2011): 451-482.

Ginsberg, Matthew L., and David E. Smith. "Reasoning about action I: A possible worlds approach." *Artificial intelligence* 35, no. 2 (1988): 165-195.

Gopalratnam, Karthik, and Diane J. Cook. "Active LeZi: An incremental parsing algorithm for sequential prediction." *International Journal on Artificial Intelligence Tools* 13, no. 04 (2004): 917-929.

Guralnik, Valerie, and Karen Zita Haigh. "Learning models of human behaviour with sequential patterns." In *Proceedings of the AAAI-02 workshop “Automation as Caregiver*, pp. 24-30. 2002.

Floyd, Michael W., Babak Esfandiari, and Kevin Lam. "A case-based reasoning approach to imitating RoboCup players." In *21st International Florida Artificial Intelligence Research Society Conference*, pp. 251-256. 2008.

Hintikka, Jaakko. "Individuals, possible worlds, and epistemic logic." *Nous* 1, no. 1 (1967): 33-62.

Jiang, Hong, Jose M. Vidal, and Michael N. Huhns. "EBDI: an architecture for emotional agents." In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, p. 11. ACM, 2007.

Karpov, Igor V., Jacob Schrum, and Risto Miikkulainen. "Believable Bot Navigation via Playback of Human Traces." In *Believable Bots*, pp. 151-170. Springer Berlin Heidelberg, 2012.

Konolige, Kurt. "What awareness isn't: A sentential view of implicit and explicit belief." In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge*, pp. 241-250. Morgan Kaufmann Publishers Inc., 1986.

Lam, Kevin, Babak Esfandiari, and David Tudino. "A scene-based imitation framework for robocup clients." *MOO-Modeling Other Agents from Observations*(2006).

Leon, Enrique, Graham Clarke, Victor Callaghan, and Faiyaz Doctor. "Affect-aware behaviour modelling and control inside an intelligent environment." *Pervasive and Mobile Computing* 6, no. 5 (2010): 559-574.

Liu, Jia Jie, G. S. Huang, Y. L. Wang, and R. C. T. Lee. "Edit distance for a run-length-encoded string and an uncompressed string." *Information Processing Letters* 105, no. 1 (2007): 12-16.

Martınez-Miranda, Juan, and Arantza Aldea. "Emotions in human and artificial intelligence." *Computers in Human Behavior* 21, no. 2 (2005): 323-341.

Papadimitriou, Christos H., and John N. Tsitsiklis. "The complexity of Markov decision processes." *Mathematics of operations research* 12, no. 3 (1987): 441-450.

Priesterjahn, Steffen. *Online imitation and adaptation in modern computer games*. Südwestdeutcher Verlag Für Hochschulschriften, 2008.

Railsback, Steven F., Steven L. Lytinen, and Stephen K. Jackson. "Agent-based simulation platforms: Review and development recommendations." *Simulation* 82, no. 9 (2006): 609-623.

Schrum, Jacob, Igor V. Karpov, and Risto Miikkulainen. "UT^ 2: Human-like behavior via neuroevolution of combat behavior and replay of human traces." In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pp. 329-336. IEEE, 2011.

Schrum, Jacob, and Risto Miikkulainen. "Evolving multi-modal behavior in NPCs." In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pp. 325-332. IEEE, 2009.

Seymour, Ben, and Ray Dolan. "Emotion, decision making, and the amygdala." *Neuron* 58, no. 5 (2008): 662-671.

Sharada, G., and O. B. V. Ramanaiah. "An Artificial Intelligence Based Neuro-Fuzzy System with Emotional Intelligence." *International Journal of Computer Applications* 1, no. 13 (2010): 74-79.

Sloman, Aaron. "Beyond shallow models of emotion." *Cognitive Processing: International Quarterly of Cognitive Science* 2, no. 1 (2001): 177-198.

Stanislaw, Harold. "Tests of Computer Simulation Validity: What Do They Measure?." *Simulation and Gaming* 17, no. 2 (1986): 173-91.

Umarov, Iskander, and Maxim Mozgovoy. "Believable and effective AI agents in virtual worlds: Current state and future perspectives." *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 4, no. 2 (2012): 37-59.

Van Hoorn, Niels, Julian Togelius, Daan Wierstra, and Jürgen Schmidhuber. "Robust player imitation using multiobjective evolution." In *Evolutionary Computation*, 2009. CEC'09. IEEE Congress on, pp. 652-659. IEEE, 2009.

Velásquez, Juan D. "Modeling emotions and other motivations in synthetic agents." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 10-15. John Wiley & Sons LTD, 1997.

Wahle, Joachim, and Michael Schreckenberg. "A multi-agent system for on-line simulations based on real-world traffic data." In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pp. 9-pp. IEEE, 2001.