

Improving Assessment Of Students Through Semantic Space Construction

Roberto Pirrone
Universita' degli Studi Palermo
Viale delle Scienze Ed. 6 P 3.
90128 Palermo (IT)
pirrone@unipa.it

Vincenzo Cannella
Universita' degli Studi Palermo
Viale delle Scienze Ed. 6 P 3.
90128 Palermo (IT)
cannella@dinfo.unipa.it

Giuseppe Russo
Universita' degli Studi Palermo
Viale delle Scienze Ed. 6 P 3.
90128 Palermo (IT)
russo@dinfo.unipa.it

Abstract—Assessment is one of the hardest tasks an Intelligent Tutoring System has to perform. It involves different and sometimes uncorrelated sub-tasks: building a student model to define her needs, defining tools and procedures to perform tests, understanding students' replies to system prompts, defining suitable procedures to evaluate the correctness of students' replies, and strategies to improve students' abilities after the assessment session.

In this work we present an improvement of our system, TutorJ, with particular attention to the assessment phase. Many tutoring systems offer only a limited set of assessment options like multiple-choice questions, fill-in-the-blanks tests or other types of predefined replies obtained through graphical widgets (radio-buttons, text-areas). This limited set of solutions makes interaction poor and unable to satisfy the users' needs. Our interest is to enrich interaction with dialog in natural language. In this respect, the assessment problem is strictly connected to natural language understanding. The preliminary step is indeed to understand questions and replies of the student.

We have reviewed the system design in the framework of a cognitive architecture with the aim to reach a double result: the reduction of the effort for the construction of the knowledge base and the improvement of the system capabilities in the assessment process. To this aim a new common semantic space has been defined and implemented. The entire architecture is oriented to intuitive and natural interaction.

Index Terms—Intelligent Tutoring Systems, Semantic Space Construction, Natural Language Interaction

I. INTRODUCTION

Intelligent Tutoring Systems (ITSs) are highly interactive software systems able to perform complex tasks with the purpose to support a student in her learning process. Interaction allows the student to achieve major learning objectives such as to attain a better comprehension of the topics under investigation, to improve her abilities in problems analysis and synthesis, to increase her explanation faculties. The interaction process is itself a complex task and different interaction modalities have been presented for ITSs. TutorJ, the system presented in this work, has a natural language module able to interact with users. A further feature in our system is the capability to mix both linguistic and graphical interaction modalities as deeper detailed in a previous work [1].

The main drawback of ITSs is that a lot of bootstrap and frequent update work is needed to obtain effective tutoring. To overcome these limitations our system has been redesigned

as a cognitive architecture. This development is intended to define a "curiosity mechanism" able to trigger the growth of the knowledge base of the system. In this scenario the knowledge base of the system is dynamic. As a consequence, the assessment has to be dynamic also, and it's related to the evolution of the dialogue. In our vision, an enhanced natural language understanding is a powerful tool to perform assessment of students' skills.

An important consideration is how to relate different learning modalities inside an interactive system in order to improve effectively the student's skills. The system has to be proactive and able to reply to students' needs with heterogenous and dynamic sources of knowledge. Educational Psychology defines self-regulated students [2] as the ones that are able to perform metacognitive learning strategies. A self-regulated tutor should be needed to reply to such students. The degree of complexity involved in such an architecture is far to be reached. Nevertheless, our approach is aimed to obtain a system that plans its activities, and is able to control its behavior and to evaluate its performance. The first seeds of this system has been implemented as a dynamic approach to knowledge base modifications and a dialogue adaption. The knowledge base of the system has its own repository and consists of structured and unstructured data. The system is composed of different modules and is goal-oriented. Each module deals with a particular task. Complex tasks are achieved through modules cooperation. The result is a layered system. We have introduced a middle stratum through a common semantic space that acts as connection between the high level of the system (the presentation one) and the low level (the repository of learning materials that can be regarded as "knowledge items"). In this work we focus on the relevance of the proposed approach with some preliminary considerations and results.

The paper is arranged as follows: next section presents a brief overview of ITSs with respect to the assessment capabilities. The third section briefly presents our system with the improvement given from the definition of the new semantic space. The fourth section presents the assessment module in more detail. The process of semantic space creation is reported in section five. The last section reports some conclusions and the future work.

II. STATE OF THE ART

The architecture of an Intelligent Tutoring System is defined according to the construction of some main modules that are devoted to peculiar operations. The main modules are the expert model, the student model, the instructional model and the learning environment.

- Expert Model: manages the knowledge base of the system and is devoted to provide solutions to students.
- Student Model: compares student's abilities and skills to the expert model to define her level of expertise
- Instructional Model: is the environment deputed to the presentation of new knowledge and activities to students.
- Learning Environment: is the entire environment. It contains all the other modules that are not directly connected to the ones mentioned above.

There are two traditional models to design ITSs: the former is based on the explicit definition of the components. This approach is defined as Model-Tracing Tutor (MTTs) and has the advantage to state explicitly the functionalities for all the different parts of the architecture [3] [4]. The latter defines the tutor activity as a constrained problem and tries to adopt solutions that best fit the problem. This approach is the Constraint-Based Model Tutor (CBMT) [5] [6]. Recent approaches try to define the whole system as a cognitive architecture [7] [8]. Cognitive Architectures refer to human cognitive processes and try to replicate the same behaviors. The Model Human Processor [9] defines several modules. The main modules are the perceptual module, the cognitive module and the motor module. The interaction between modules tries to represent cognitive processes. ACT-R [10] [11] defines an architecture with a module containing declaratively memorized information. It has some other modules to manage the current goal, the stack of goals and a visual representation of objects in the domain.

The assessment is performed differently according to the used representation model. Some systems perform bayesian assessment [12] with at least three different aspects according to the focus of the optimization: the student model, the data model, the output of the system. Another example of bayesian based assessment is presented in [13]. Bayesian networks are used to represent the student's knowledge. There are two main student models: the procedural problem solving model and the item response model. In the first one, bayesian networks are used to calculate probabilistic assessments of different kind of information. In the latter one they are used to bound the test items and the students' abilities. Other systems adopt ad-hoc mechanisms to perform assessment [14] [15]. These systems are organized in terms of elementary units of knowledge (*knowledge items*) that have to be bounded with some methodology (probabilistic, production systems and so on). Our approach follows the Knowledge Space Theory [16] that defines an expertise area as a collection of items, which can be notions, skills or abilities. To each student is associated her *Knowledge State* that is the collection of items she is able to master. Starting from this formulation, every item has its

representation in the space that can be used to calculate the gap the student has to fill to master it.

III. SYSTEM OVERVIEW

The development of our system is inspired to a cognitive model. The reader is referred to [17] for a more accurate explanation of the previous system's components. The new architecture is depicted in the above figure (See Fig 1.).

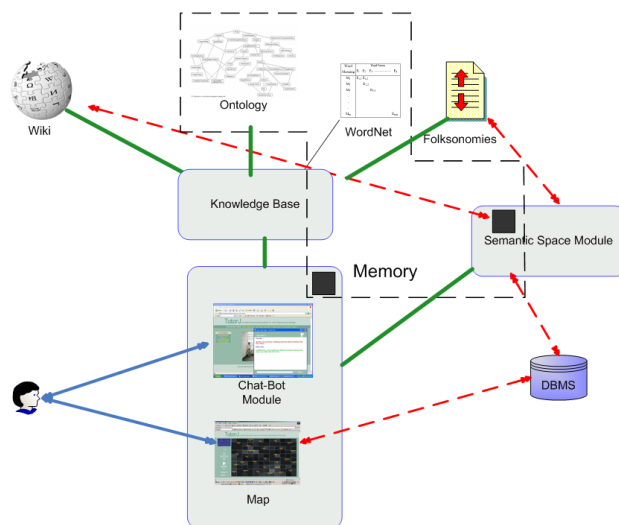


Fig. 1. The System Architecture: dotted red lines represent the data exchange from the system to external components. Green solid lines represent the internal data exchange between the different components, while blue ones represent information exchange with the user.

The system is composed of different modules and is goal-oriented. Each module deals with a single task. Complex tasks are accomplished as a cooperation between different modules. The main tasks are: interaction with the student, retrieval of learning contents and planning of a suitable learning path.

TutorJ owns a “curiosity” mechanism to increase dynamically its knowledge. According to Loewenstein [18] curiosity “occurs when an individual’s informational reference point becomes elevated in a certain domain, drawing attention to an information gap”. The two most important aspects in this theory are the awareness of what is known (and what is unknown) as a trigger for curiosity (perceptual curiosity) and the feeling of deprivation as a result of a lack of information in a particular subject (epistemic curiosity). In our framework both perceptual and epistemic curiosity are present. The former is expressed as a consequence of a communication act performed with students, the latter attains what is called *optimal discrepancy* in Education Psychology. In this case the system measures the difference between the structure (concepts and relations) of the knowledge gathered from the web and the one stored in its KB. If this difference is above a suitable threshold, the new knowledge is integrated in the long term memory (the system’s knowledge base). Awareness and curiosity enable a self regulated behavior in TutorJ. Pintrich [19] defines self regulated learning (SRL) as a process in four steps:

planning, monitoring, control and reaction/reflection. Each SRL phase involves the student as regards his/her cognitive area, the motivation to learn, the behavior while performing learning actions, and the learning context. In this respect we can think about the perceptual curiosity as affecting the system's planning and monitoring. In this scenario the learner's "Judgment of learning" (JOL) stimulates the system's "Feeling of Knowing" (FOK). On the other hand, epistemic curiosity induces TutorJ to plan new knowledge acquisition on the basis of its judgment about "Ease of Learning" (EOL). The system crawls the web to increase its knowledge, searching for repositories of semi-structured information like Wikipedia. We have adapted a crawler to integrate documents belonging to a particular subject. All the documents are used to create a common semantic space used as a sub-symbolic long term memory of the system according to the cognitive paradigm. The modified documents are stored and are projected in the semantic space that is incrementally updated. The interaction process is deputed to a novel chat-bot module that is based on a mixed conversation model performing where both linguistic and graphical interactions are seen as elementary dialogue moves.

The cognitive architecture we propose is mainly focused on how the interaction process is performed. The whole process of performing assessment has been redesigned to this aim. Interaction takes place in the Learning Environment by means of the interface modules: an enhanced chat-bot that is able to perform both linguistic and graphic interaction and a pan-zoom map that is used to show learning materials and the learning paths proposed to the learner. This module depicts the main concepts of the domain as a starred sky where constellations are related to the amount of documents related to each topic. More concepts are visible while zooming in a constellation, and documents are accessible at the maximum zoom level. The map module is directly connected to the knowledge base and to the module that manages the creation of the semantic space. In the original version of the system this module essentially was able to perform Latent Semantic Analysis (LSA) [20] on the document corpora. The LSA module computes similarity between documents and concepts to create the space. The major drawback of LSA is that the analysis of texts is based solely on statistical considerations and produces a bag-of-words representation of the space without information about the context and without the possibility to evaluate a phrase as a whole. Some studies have been proposed with the aim to have more performative results [21]. We propose an approach that uses an augmented definition of the input taking into account also the related POS (part-of-speech) tag of the preceding word in the phrase. Our approach tries to define a different space to better disambiguate the meaning of the input sentences of the students. We are implementing a different semantic space based on a tree representation of phrases. A semantic space based on the structural properties of a phrase will improve the understanding of students' sentences. The results will lead to a very rich interaction process and to enhance performances in assessment. We have defined two possible

interaction modalities with students: the first is based on the conversation so the most important information is about the subject under investigation, and the learning path planned at the current time. In the second interaction modality the system acts as an information provider. The system has to track documents provided to the user. A main feature is the ability to represent a collection of items as a possible learning path. This learning path is a personalized one and it follows the evolution of the dialog. The goal is a more effective presentation of materials. The system stimulates planning and self monitoring in the student via different arrangements of the didactical materials as regards both their browsing and the ability to present exercises or deepening texts, and so on.

IV. SEMANTIC SPACE CONSTRUCTION

The semantic space construction is the major improvement in the system definition. According to Lowe [22] a semantic space is a quadruple $\{W, L, S, R\}$. W is the set of basis elements that are functionally related and define the vocabulary. Since different representations are possible the W set can be alternatively either a set of uncorrelated words or a set of tagged words or a set of words that are related syntactically in some order. L is the lexical association function applied to the occurrences of the target element in the vocabulary. The most simple example of lexical association function is the identity function that counts raw frequencies of the basis elements in the corpora. S is a similarity measure used to compute the similarity of two samples in the space, and R is a transformation aimed to reduce the dimensionality of the input space. Our purpose is to define L and S starting from the definition of W . The reduction of the space is not the focus of our work at the present time, since some useful methodologies like Singular Value Decomposition (SVD) have been used to perform space reduction in many fields like genetic algorithms [23] or video streaming [24] and also speech recognition [25]. SVD was also used in the old LSA implementation of the Semantic Space Module.

In our formulation the basis element to define the semantic space is a complete representation of a phrase. The W set is defined according to the following steps that are performed off-line, and are used to update the space periodically (see Fig. 2).

- Web crawling to search new documents.
- Text analysis and tokenization.
- Summarization of the documents.
- Analysis of single documents and definition of the trees.

The system crawls periodically the pages in Wikipedia dealing with a particular subject. This step is performed when there is a major distance between the system knowledge and the subject treated by a student. The text analysis and tokenization is directly connected to the definition of suitable inputs for building tree structures from phrases. Some operations like text chunking, abbreviations replacement and tokenization are required. The summarization of text consists essentially in the definition and storage of persistent information about a portion of text that has been analyzed. This operation is obtained

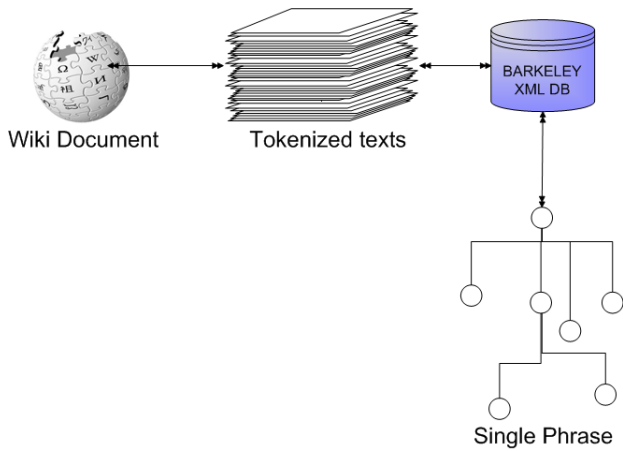


Fig. 2. The main steps of the process. Documents are manipulated and stored inside a native XML database. The result of the memorization process is a set of phrases in natural language represented as trees.

with a persistent storage inside a database. For the purpose of our analysis the chosen DBMS is the Berkeley XML DB [26], a native XML Database that optimize the usual insert and delete operations. A phrase is associated to the set of relevant words it contains. The database is arranged as a multilevel phrases clustering. Level n contains the clusters of phrases with exactly n words. Clustering is useful to reduce the computational load. After devising the number of relevant words in the input the phrase, summarization starts immediately from the corresponding level. The LSA module in the previous version of TutorJ is used also to achieve a significant computational saving. LSA allows the system to define words that are close to the input phrase directly. These simple operations are used to define a multilevel organization of words that is the combination of the relevant words in the space. After the definition of the set of phrases that have to be compared the system starts to calculate distances.

The phrases are represented topologically as trees. Tree representation of phrases can be obtained from several tools. We are comparing and integrating two different tools: the Stanford parser [27] and the FreeLing software [28]. Both are able to parse a standard text and to produce trees with POS and morpho-syntactic information. An example of the produced trees is depicted in figure 3 and 4.

Both the trees contain coded information of the single parts of the phrase. We used the Penn TreeBank notation [29] that carries out syntactic and lexical information.

As regards the definition of the similarity measure S for our space, we have defined a new metric distance between phrases that are projected in it. In this way is possible to better evaluate the correctness of the student input and is also possible to define the direction of the learning path. This is the major novelty of our approach with respect to other ones that are able to perform only static assessment. The distance measure is calculate starting from the definition of a semantic space. The distance is the starting point for the definition of a metric.

```
(ROOT
(S
(NP (DT this))
(VP (VBZ is)
(NP
(NP (DT the) (JJ first) (NN sentence))
(SBAR
(WHNP (WDT that))
(S
(VP (VBZ starts)
(ADVP (RB here))))))
(. .)))
nsubj(sentence-5, this-1)
cop(sentence-5, is-2)
det(sentence-5, the-3)
amod(sentence-5, first-4)
nsubj(starts-7, that-6)
rcmod(sentence-5, starts-7)
advmod(starts-7, here-8)
```

Fig. 3. An example of the input tree obtained from the Stanford Software.

```
The the DT 0.9998
analyzer analyzer NN 1
performs perform VBZ 1
his his PRP$ 1
task task NN 1
that that WDT 0.23911
is be VBZ 0.999707
the the DT 0.9998
analysis analysis NN 1
of of IN 0.999901
sentences sentence NNS 0.954545
. . Fp 1
```

Fig. 4. An example of the input obtained from FreeLing Software

The system computes the distance as a topological distance of trees. A common measure to define the distance between trees is the tree edit distance. This distance is able to measure how far or related are two structures in terms of the necessary steps to turn one structure into another. It is based on simple tree edit operations consisting of *deleting*, *inserting*, and *relabeling* nodes. A survey is presented in [30].

Let T is our rooted tree representing a phrase. T is a labeled tree and to each node is assigned a symbol from a finite alphabet α . T is also an ordered tree because is possible to establish a left-to-right order among siblings. The tree edit operations are defined as follows:

- *delete*: Delete a non-root node n in T whose parent node is n' , making the children of n become the children of n' . All the children are inserted in the place of n in the same order of the children of n' .

- *insert*: The complement of delete. Insert a node n as a child of n' in T making n the parent of the children of n'
- *relabel*: Change the label of a node n in the tree T

Starting from the previous operations, it's possible to define the *tree edit distance problem*. According to [30] an edit script ES between $T1$ and $T2$ is a sequence of edit operations turning $T1$ into $T2$. The cost of ES is the sum of the costs of the single operations. An optimal edit script between $T1$ and $T2$ has minimum cost and this cost is the tree edit distance. The tree edit distance problem is to compute the edit distance and the corresponding edit script. Our initial operation cost has the same unitary value for all the possible operations and is obtained starting from the methodology presented in [31].

V. ASSESSMENT PROCESS

The semantic space is used to assess the students' replies to question made by the system with the purpose to define a learning path. During the assessment procedure the system performs three kinds of actions:

- Projection of the student's sentence in the semantic space.
- Visualization of the system reply (graphical or textual)
- Definition of the learning path

The concepts under investigation are defined according to a common ontology that relies on two main levels: the structural level and the navigational one. The former defines the structural properties of the domain in terms of concepts, attributes and their possible values. The latter defines the prerequisite concepts for a particular subject and suggests a possible learning path. There are two levels of prerequisite concepts: strict and weak. Strict prerequisites has to be satisfied from the student. Weak ones represent a possible way to deepen particular subjects. A suitable threshold is defined and is used to characterize the system behavior. The system has some predefined ranges that trigger its behavior. The main actions are a second question about the topic, a question about the predecessors of the concept (defined as prerequisites concepts) or the presentation of the related materials. The assessment procedure is iteratively repeated until a suitable learning path is obtained.

VI. CONCLUSIONS AND FUTURE WORK

In this work an evolution of TutorJ has been presented. TutorJ is an ITS developed by some of the authors whose features have been extended to build a cognitive agent with a self regulated behavior and an increased capability to perform assessment. In this work we have focused our attention on assessment, and a new module for natural language understanding has been presented to achieve a more natural interaction with user. A new (more than LSA) semantic space has been devised to bridge the gap between the lowest level (the document corpora) and the highest level (the knowledge base of the system). Another important advantage is the definition of a "curiosity" mechanism that is able to help the system to enrich its knowledge with respect to the requests of users.

This in turn enriches the semantic space and the understanding ability of the system.

Future works are in the direction of further optimization of the semantic space. We are currently working on a more sophisticated solution including a new definition of S . Another interesting topic is the way to take into account the student's preferences while presenting learning materials. In this respect, we are exploiting the features of self regulated learning.

REFERENCES

- [1] R. Pirrone, V. Cannella, and G. Russo, "Gaiml: A new language for verbal and graphical interaction in chatbots," *Complex, Intelligent and Software Intensive Systems, International Conference*, vol. 0, pp. 715–720, 2008.
- [2] B. J. Zimmermann, *Theories of Self-Regulated Learning and Academic Achievement: An Overview and Analysis*. Lawrence Erlbaum Associates, 2001, ch. 1, pp. 1–37.
- [3] A. S. Gertner and K. VanLehn, "Andes: A coached problem solving environment for physics," in *ITS '00: Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. London, UK: Springer-Verlag, 2000, pp. 133–142.
- [4] B. P. Butz, M. Duarte, and S. M. Miller, "An Intelligent Tutoring System for Circuit Analysis," *IEEE Transactions on Education*, vol. 49, pp. 216–223, may 2006.
- [5] A. Mitrovic, B. Martin, and P. Suraweera. (2006) *Intelligent Tutors for All: The Constraint-Based Approach*.
- [6] C. Gonzalez, J. C. Burguillo, and M. Llamas, "Integrating Intelligent Tutoring Systems and Health Information Systems," *18th International Conference on Database and Expert Systems Applications (DEXA 2007)*, pp. 633–637, 2007.
- [7] G. M. Goh and C. Quek, "EpiList: An Intelligent Tutoring System Shell for Implicit Development of Generic Cognitive Skills that support Bottom-Up Knowledge Construction," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 37, no. 1, pp. 58–71, 2007.
- [8] A. V. Samsonovich, K. D. J. ana A. Kitsantas, E. E. Peters, N. Dabbagh, and M. Kalbfleisch, "Cognitive Constructor: An Intelligent Tutoring System based on a Biologically Inspired Cognitive Architecture (BICA)," *Artificial General Intelligence 2008: Proceedings of the First AGI Conference. Frontiers in Artificial Intelligence and Applications*, vol. 171, no. 1, pp. 311–325, 2008.
- [9] S. K. Card, A. Newell, and T. P. Moran, *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1983.
- [10] J. R. Anderson, D. Bothell, M. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "integrated theory of the mind." *Psychological Review*, vol. 111, pp. 1036–1060, 2004.
- [11] J. R. Anderson and C. L. Lebiere, "The newell test for a theory of cognition." *Behavioral and Brain Science*, vol. 26, pp. 587–637, 2003.
- [12] L. Zhang, Y. ting Zhuang, Z. ming Yuan, and G. hua Zhan, "Auto diagnosing: An intelligent assessment system based on bayesian networks," *Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual*, pp. T1G–7–T1G–10, Oct. 2007.
- [13] Z. Liu and H. Wang, "A modeling method based on bayesian networks in intelligent tutoring system," *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, pp. 967–972, April 2007.
- [14] F. Aymerich, N. Dessì, B. Pes, and A. Saba, "An automatic assessment system supporting computer science entrance examinations," in *ICALT, 2004*.
- [15] E. Guzman, R. Conejo, and J.-L. Perez-de-la Cruz, "Improving student performance using self-assessment tests," *Intelligent Systems, IEEE*, vol. 22, no. 4, pp. 46–52, July-Aug. 2007.
- [16] J.-C. Falmagne, E. Cosyn, J.-P. Doignon, and N. Thiery, "The assessment of knowledge, in theory and in practice," *Integration of Knowledge Intensive Multi-Agent Systems, 2003. International Conference on*, pp. 609–615, Sept.-4 Oct. 2003.
- [17] R. Rizzo, G. Pilato, and R. Pirrone, "A KST-Based System for Student Tutoring," *Applied Artificial Intelligence*, vol. 22, no. 4, pp. 283–308, 2008.

- [18] G. Loewenstein, "The psychology of curiosity : a review and reinterpretation," *Psychological bulletin*, vol. 116, no. 1, pp. 75–98, 1994.
- [19] P. R. Pintrich, "The role of goal orientation in self-regulated learning," in *Handbook of self-regulation*, M. Boekaerts, P. R. Pintrich, and M. Zeinder, Eds. San Diego, CA: Academic Press, 2000, pp. 451–502.
- [20] T. Landauer, P. Foltz, and D. Laham, "An Introduction to Latent Semantic Analysis," *Discours Processes*, vol. 25, pp. 259–284, 1998.
- [21] D. Kanejiya, A. Kumar, and S. Prasad, "Automatic evaluation of students' answers using syntactically enhanced lsa," in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 53–60.
- [22] W. Lowe, "Toward a Theory of Semantic Space," *Procs of 22nd Annual Conference of the Cognitive Science Society*, vol. 1, pp. 675–680, 2001.
- [23] J. Martin and K. Rasheed, "Using singular value decomposition to improve a genetic algorithm's performance," *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 3, pp. 1612–1617 Vol.3, Dec. 2003.
- [24] Z. Cernekova, C. Kotropoulos, and I. Pitas, "Video shot segmentation using singular value decomposition," *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 2, pp. II–301–4 vol.2, July 2003.
- [25] B. Lilly and K. Paliwal, "Robust speech recognition using singular value decomposition based speech enhancement," *TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE*, vol. 1, pp. 257–260 vol.1, Dec. 1997.
- [26] "Introduction to berkeley db xml - a guided tour of this native xml database system," http://www.oracle.com/technology/documentation/berkeley-db/xml/intro_xml/index.html.
- [27] D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, vol. 1, 2003.
- [28] J. Atserias, B. Casas, E. Comelles, M. Gonzalez, L. Padro, and M. Padro, "Freeling 1.3: Syntactic and semantic services in an open-source nlp library," in *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, vol. 1, 2006, pp. 184–187.
- [29] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. Macintyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: Annotating predicate argument structure," in *In ARPA Human Language Technology Workshop*. Morgan Kaufmann, 1994.
- [30] P. Bille, "A survey on tree edit distance and related problems," *Theor. Comput. Sci.*, vol. 337, no. 1-3, pp. 217–239, 2005.
- [31] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," *SIAM J. Comput.*, vol. 18, no. 6, pp. 1245–1262, 1989.