A Conversational Agent Based on a Conceptual Interpretation of a Data Driven Semantic Space

Francesco Agostaro[1], Agnese Augello[1], Giovanni Pilato[2], Giorgio Vassallo[1], and Salvatore Gaglio[1]

[1] DINFO-Dipartimento di Ingegneria Informatica,Università degli studi di Palermo, Viale delle Scienze - Edificio 6, 90128 Palermo, Italy
{agostaro, gvassallo, gaglio}@unipa.it, augello@csai.unipa.it
[2] ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni, CNR - Consiglio Nazionale delle Ricerche, Viale delle Scienze - Edificio 11, 90128 Palermo, Italy
g.pilato@icar.cnr.it

Abstract. In this work we propose an interpretation of the LSA framework which leads to a data-driven "conceptual" space creation suitable for an "intuitive" conversational agent.

The proposed approach allows overcoming the limitations of traditional, rule-based, chat-bots, leading to a more natural dialogue.

1 Introduction

Natural language dialogue interfaces, like chat-bot systems, are simple to build and they can be used as interfaces for a large set of applications (entertainment, educational and e-learning platforms, research engines, e-commerce web-site navigation systems and so on). However the pattern matching rules they are based on are often too restrictive. Many approaches try to integrate the simple technology of chat-bot systems, with more sophisticated techniques[15,16,17,18]. In [19] a dialogue system has been implemented, which inherits from the chat-bots systems the robustness and the locality of pattern-matching based dialogue management, but uses an ontological model in order to abstract dialogue-acts from the specific inputs. Another attempt to improve the human features of chat-bots has been realized with the CyN project [13], whose aim is to link the pattern matching interpreter of a well-known conversational system, Alice[3], to OpenCyc[14], the largest commonsense knowledge base available today.

In recent years a paradigm, named Latent Semantic Analysis (LSA), useful to extract and represent the meaning of words by statistical computations applied to a large corpus of texts, has been proposed [11]. LSA is based on the *vector space method*: given a text corpus of N documents and M words, the LSA paradigm defines a mapping between the M words and the N documents into a continuous vector space S, where each word w_i is associated to a vector U_i in S, and each document d_j is associated a vector V_j in S[6]. The S vector space is a "semantic space", since semantics conveyed by the presence of the i-th word

S. Bandini and S. Manzoni (Eds.): AI*IA 2005, LNAI 3673, pp. 381–392, 2005.
© Springer-Verlag Berlin Heidelberg 2005

in the j-th document can be measured by taking the dot product between U_i and V_j. In fact, if we apply LSA to a large information conveying text corpus, segmented in N documents, we obtain a word-document co-occurrence $M \times N$ matrix A (i.e. the (i, j) entry is the count of occurrences of the i-th word within the j-th document).

The Latent Semantic methodology has been also applied to a community of traditional Alice chat-bots[1,2]. In particular, each chat-bot had a specific knowledge on one single topic[1], then a semantic space was created, in which the knowledge bases of the chat-bots have been vector-coded, allowing them to estimate their own competence about questions asked by the user.

In this paper we show that a particular interpretation of the Latent Semantic Analysis paradigm helps to better design an human-like conversational agent (here called *LSA-bot*).

It is well known that the Latent Semantic Analysis technique is capable of simulating several human cognitive phenomena (word-categorization, sentence-word semantic priming, discourse comprehension, judgments of essay quality, etc.)[11]. Besides, under specific hypotheses, the co-occurrence matrix A can be considered as a sample set, by which a word-document co-occurrence probability distribution can be inferred by means of an estimator. This can be achieved by taking a Truncated Singular Value Decomposition (TSVD) approximation of the original matrix.

In order to ensure estimator's sufficiency, a pre-processing on the original matrix A has to be performed before evaluating its approximation. We show that singular vectors of the approximated matrix can be interpreted as orthonormal basis vectors of a "conceptual" space. This "conceptual" space is entirely data driven, since it has been constructed by processing a matrix automatically arranged from the raw text corpus. Moreover we do not need to introduce any hierarchical structure, since orthonormality of basis vectors ensures independency among the vectors which generate the conceptual space, so that domains may not be included in its algebraic structure (as required, for example, in Gardenfors conceptual spaces[8]).

This approach allows to overcome the limitations of classic pattern matching based chat-bot thanks to the intuitive/associative capabilities provided by a data driven, automatically constructed, "conceptual" space. In fact this space has the same psychological basis claimed by the LSA [11] and therefore this choice allows to code somehow the intuitive/associative component of human brain. The knowledge base of the LSA-bot is sub-symbolically coded in the "conceptual" space, then the association capability is implemented by mapping the user's question in the same conceptual space and comparing the coded query with the sub-symbolically coded elements of the knowledge base. Experimental trials and comparisons have been carried out by using the cosine measure between the user query vector and each of the vectors representing the answers contained in the LSA-bot knowledge base. Answers that turn out to be closest to the query vector are then shown to the user.

The remainder of the paper is organized as follows: in section 2 it is described the proposed "conceptual" interpretation of data-driven semantic space; in section 3 it is illustrated the realization of a chat-bot with "intuitive" capabilities obtained mapping its knowledge in the created "conceptual" space, the dialogue implementation, the experimental results concerning also a comparison with the traditional Alice architecture[3]. Then, in section 4, conclusions and future work are outlined.

2 Data-Driven Conceptual Space Creation

2.1 Theoretical Background: The LSA Paradigm

In recent years a paradigm, named Latent Semantic Analysis (LSA), useful to extract and represent the meaning of words by statistical computations applied to a large corpus of texts, has been proposed [11]. LSA is based on the *vector space method*: given a text corpus of N documents and M words, the LSA paradigm defines a mapping between the M words and the N documents into a continuous vector space S, where each word is associated to a vector in S, as well as each document is associated a vector in S[6]. The S vector space is a "semantic space", since semantics conveyed by the presence of the i-th word in the j-th document can be measured by taking the dot product between the vector representing the word and the vector representing the document.

Let us consider a corpus made of N text documents, and let M be the number of words in the whole corpus (counting each word only once even if it occurs more than once). Then let A be the $M \times N$ matrix whose (i, j) entry is the count of the occurrences of the i-th word in the j-th document. Hence A_{ij} is the (not normalized) occurrence frequency of the i-th word within the j-th document. As a consequence the i-th row of the matrix A can be interpreted as representative of the i-th word's behaviour within the entire corpus, while the j-th column of the matrix A can be interpreted as representative of the j-th document within the entire corpus [11,6].

However this approach presents some drawbacks. In fact the vectors representing words and the vectors representing documents belong to different vector spaces, respectively to $\mathbb{R}^N$ and to $\mathbb{R}^M$. It's experimentally verified [10] that these vectors are very sparse, and that M and N can reach very large values. The *Truncated Singular Value Decomposition* (TSVD) technique is considered a standard technique in order to overcome these drawbacks [11,5]. First, the *Singular Value Decomposition* of the matrix A is performed, i.e. the matrix A is decomposed in the product $A = U\Sigma V^T$, where U is a column-orthonormal[1] $M \times N$ matrix, V is a column-orthonormal $N \times N$ matrices and Σ is a $N \times N$ diagonal matrix, whose elements are called *singular values* of A. The columns of U are called *left singular vectors* of A and the columns of V are called *right singular vectors* of A. We can suppose, without loss of generality, that A's singular values are

[1] i.e. the dot product of two different columns of U is 0, while each column squares to 1

ranked in decreasing order, since Singular Value Decomposition also holds if we perform permutations of singular values along with the correspondent rows in U and V [5].

Let R be a positive integer with $R < N$, and let $\tilde{U}$ be the $M \times R$ matrix obtained from U by suppressing the last $N - R$ columns, $\tilde{\Sigma}$ the matrix obtained from Σ by suppressing the last $N-R$ rows and the last $N-R$ columns and $\tilde{V}$ be the $N \times R$ matrix obtained from V by suppressing the last $N-R$ columns. Then $\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ is a $M \times N$ matrix of rank R. It can be shown [5] that $\tilde{A}$ is the best rank R approximation of the matrix A (among the $M \times N$ matrices) with respect to the metric obtained by assuming as distance among two arbitrary matrices X and Y the non-negative real number $d_F(X,Y) = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - Y_{ij})^2}$. This is the *Frobenius distance* between the two matrices X and Y. The matrix $\tilde{A}$ is said to be obtained from the matrix A by *Truncated Singular Value Decomposition* (TSVD), where the term "truncated" recalls that we obtained $\tilde{A}$ by suppressing the smallest $N - R$ singular values of A along with the corresponding columns of U and V. The $i - th$ row of the matrix $\tilde{U}$ may be considered as representative of the $i - th$ word, while the j-th word of the matrix $\tilde{U}$ may be considered as representative of the j-th document. Appropriateness of the presence of the i-th word within the j-th document can be measured by the cosine between the two vectors representing the word and the document.

2.2 A Proposal of a "Conceptual" Interpretation of the Semantic Space

In this subsection we propose a "conceptual" interpretation of the orthonormal bases of the semantic space constructed with the technique outlined above. In this way, the semantic space can be regarded as a "conceptual" space. The term "conceptual space" may be misleading, since it recalls the well known Gardenfors conceptual spaces [8]. Our conceptual spaces are substantially different, since they are automatically constructed by subsymbolic processing of the raw sample data. On the contrary, Gardenfors spaces have to be "manually" constructed by extracting from the knowledge base the quality dimensions, so they are not suitable in order to represent the knowledge base of a conversational interface.

If we normalize the matrix A described in the previous subsection, dividing it by the sum of all its elements, A can be considered as a sample set, by which a word-document co-occurrence probability distribution can be inferred by means of an estimator. Each sample can be considered as an instance of a stochastic variable. Since the number of these variables is very large, in order to infer a probability distribution from the sample set, we should use an *estimator*, i.e. a function of the aforementioned stochastic variables. We would like this estimator to be *sufficient*, i.e. it should "catch" from the sample data only the information which is relevant with respect to latent semantics, neglecting other features that are related to the particular instances of the stochastic variables. In other words, we suppose that the matrix A of the sample data can be decomposed in the sum of two matrices, $A = \Psi + N$, where the matrix Ψ contains only the sample data which are relevant to latent semantics, while N contains all the other data. We would like the estimator to give Ψ as a result if its arguments are instantiated with the actual sample data in A. A possible estimator is obtained by the TSVD technique described above.

Unfortunately, the application of TSVD does not yield a sufficient estimator. It can be shown [12] that a sufficient estimator can be obtained by evaluating the best rank R approximation to A with respect to the *Hellinger distance*, defined by

$$d_H(X,Y) = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(\sqrt{X_{ij}} - \sqrt{Y_{ij}})^2}$$

Therefore, in order to ensure estimator's sufficiency, a pre-processing on the original matrix has to be performed before evaluating its approximation, namely each entry of the A matrix has to be replaced with its square root.

Calling B the pre-processed matrix, within the R-dimensional semantic space obtained by TSVD on the matrix B, the rows of the matrix $\tilde{U}$ and the rows of matrix $\tilde{V}$ represent the vector coding of the words and of the documents respectively.

The sufficiency of this estimator allows us (by inference) to interpret the singular vectors of B as probability distributions (and TSVD is compatible with this semantic, since B's singular vectors all square to 1). We wish to point out the relationship between the orthonormality of $\tilde{U}$'s and $\tilde{V}$'s columns, and the independence between them.

The original matrix represents relationships between words and documents. The two matrices U and V obtained after decomposition reflect a breakdown of the original relationships into linearly-independent vectors[4]. This independent R dimensions of the $\mathbb{R}^R$ space can be tagged in order to interpret this space as a "conceptual" space. Since these vectors are orthogonal, they can be regarded as principal axes, and so they can be regarded as axes which represent the "fundamental" concepts residing in the data driven space generated by the LSA, and can be tagged according to this interpretation.

2.3 An Example of "Conceptual" Axis Tagging

To clarify the procedure of the conceptual axes tagging, the technique proposed in the previous subsection has been applied to the well-known example reported in [11].

In this example the documents for the matrix construction are the titles of nine technical memoranda, five concerning human computer interaction (HCI), and four concerning mathematical graph theory. These topics are conceptually rather disjoint[11].

In the table 1 the list of the titles is presented. The extracted terms are highlighted in italics.

Table 1. The document used for the matrix construction

Titles of nine technical memoranda

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system responce time*
c3: The *EPS user interface* management *system*
c4: *Systen* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *responce time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Table 2. Results of the proposed approach for the $k = 1$ axis

WORDS	SQUARE-VALUES
system	$2,753E-01$
user	$2,203E-01$
time	$1,054E-01$
response	$1,054E-01$
computer	$7,898E-02$
EPS	$6,858E-02$
survey	$6,189E-02$
interface	$4,470E-02$
human	$3,380E-02$
graph	$2,907E-03$
minors	$2,159E-03$
trees	$4,671E-04$

Table 3. Results of the proposed approach for the $k = 2$ axis

WORDS	SQUARE-VALUES
graph	$4,226E-01$
trees	$2,719E-01$
minors	$2,178E-01$
survey	$5,123E-02$
system	$1,190E-02$
EPS	$1,069E-02$
human	$6,144E-03$
interface	$5,863E-03$
user	$1,256E-03$
time	$2,359E-04$
response	$2,359E-04$
computer	$9,355E-05$

In order to consider the obtained matrix as a probability distribution each entry has been divided by the sum of all the matrix entries. Then the matrix obtained by performing the square root of each element is computed.

The truncated singular value decomposition with $R = 2$ is then performed.

The obtained tagging is reported in Tables 2, 3: as it can be seen the first three tags for the $k = 1$ axis are *system, user, interface* which reflect the HCI topic, while the first three words for the $k = 2$ axis are *graph, trees, minors* which are related to the mathematical graph theory topic. So the two concepts related to the topics turn out to be correctly separated, thus identifying the principal axes of the "conceptual" space.

3 Mapping of a Chat-Bot Knowledge in the Created "Conceptual" Space

The automatic, data driven, creation of the "conceptual" space allows to design a natural, human-like, conversational agent.

The agent, called *LSAbot*, has an "intuitive" capability, modelled as an association mechanism. This is obtained by mapping its knowledge base into the automatically built conceptual space.

In the following, the whole procedure (from the automatic creation of the conceptual space up to the realization of LSAbot) is presented. Then LSAbot's performances are compared with a traditional Alice[3] chat-bot's ones. Alice (Artificial Linguistic Internet Computer Entity) is a well-known artificial intelligence natural language chat robot based on an experiment specified by Alan M. Turing in 1950. The Alice software utilizes AIML(Artificial Intelligence Mark-up Language), an XML-like language designed for creating stimulus-response chat robots.

The Alice chat-bot knowledge base is composed of question-answer modules, called categories and structured with AIML. The question, or stimulus, is called the "pattern". The answer, or response, is called the "template". The dialogue is based on algorithms for automatic detection of patterns in the dialogue data.

3.1 Data-Driven Conceptual Space Creation

It is well known from theory that, in order to effectively apply the LSA technique, a very large corpus of documents is required[11].

Therefore a text corpus composed of 1574 English euro-parliament documents and 910 templates of the Alice[3] standard knowledge set have been used. The sentences include greetings, definition of terms knowledge, notions about artificial intelligence and computers.

After a pre-processing of the documents, which consists in removing all the stop-words (i.e. words that do not carry semantic information like articles, prepositions and so on), a set of 101424 word forms has been obtained (no stemming has been performed).

Hence a 101424×2484 terms-documents matrix has been created, whose generic entry a_{ij} is the count of occurrences of the *i-th*word in the *j-th* document.

The matrix has been then normalized and each entry has been replaced with it square root value, obtaining a new matrix B. Then the TSVD with $R = 100$ has been performed in order to find the best approximation of A according to the Hellinger distance.

Each row of the matrix $\tilde{U}$ represents the sub-symbolic coding of each of the 101424 terms, and each row of the matrix $\tilde{V}$ represents the sub-symbolic coding of the 2484 documents: the first 1574 rows represent the euro-parliament documents and the last 910 rows represent the Alice standard templates.

The 2484 vectors associated to the documents used for the conceptual space construction make up the sub-symbolic coding of the LSAbot knowledge base.

3.2 Dialogue Implementation

After mapping the chat-bot knowledge base onto the conceptual space, the conversation between the user and the chat-bots can take place. The query of the user is coded, in the same conceptual space, as a sum of the vectors representing the terms which compose the query[7], normalized with respect to the Euclidean norm.

The similarity between this vector and the vectors representing the sentences in the conversational agent's knowledge base is then evaluated using the cosine similarity measure between each sentece's vector and the user query vector.

Let q be the user query and $\boldsymbol{q}$ its associated vector, let s be one of the knowledge base sentences and $\boldsymbol{s}$ its corresponding vector; the similarity between the query and the sentence can be evaluated as:

$$sim(q, s) = cos(\vartheta) = \frac{\boldsymbol{q} \cdot \boldsymbol{s}}{\|\boldsymbol{q}\| \, \|\boldsymbol{s}\|} \tag{1}$$

where $|\boldsymbol{q}|$ and $|\boldsymbol{s}|$ are the modules of the query vector and of the sentence vector respectively.

Then, whenever the user asks a question, LSAbot answers with the sentence of its knowledge base which minimizes the cosine with the question vector.

3.3 Experimental Results and Comparison with Alice

To test the effectiveness of the approach, the LSAbot performances have been compared with the performances of the traditional Alice[3] architecture by using a sample of questions submitted to the system.

The results have been evaluated analyzing the answers given by both the LSAbot and the Alice traditional chat-bot to the user questions.

Here we call, for the sake of clarity:

- *correct*: the right answer expected for the current question;
- *coherent*: an answer that isn't expected but it is pertaining to the current question;
- *wrong*: an unexpected and not pertaining answer to the current question.

Moreover, to point out the differences between the LSAbot and Alice, the questions submitted to both the systems are, in particular, belonging to four categories:

– questions containing few words;
– ordinary questions (related to general knowledge);
– greeting questions;
– wrong questions, i.e. questions with grammatical mistakes.

In the following tables the percentage of correct, coherent and wrong answers obtained over a sample of 20 queries submitted to both systems is illustrated; a few example dialogs segments are also shown.

In table 4 and 5 some results obtained from the comparison between Alice and LSAbot are reported. Experimentals trials have been carried out by taking four sets (each composed of 35 queries) from the four aforementioned categories.

Dialogue examples. Numerical results show that in some specific cases the traditional pattern matching mechanism of Alice fails to sustain a fluent conversation, while LSAbot shows a better behaviour.

We point out that LSAbot and Alice share the same knowledge base. The only difference between the two systems is that the Alice traditional architecture is based solely on the pattern-matching mechanism, while LSAbot is based on the intuitive mechanism given by the sub-symbolic representation of the knowledge base into the automatically data driven conceptual space.

Alice tries to answer the user's query by using its pattern-matching based engine, that looks for a lexical correspondence between the user's query and the chat-bot categories. The Alice pattern matching module may fail in finding the right answer, even if its knowledge base contains categories with the right answer. LSAbot overcomes this drawback thanks to its association capability.

The cases in which Alice fails may be summed up as follows:

– when the user query is composed of only one single word, or when the query is composed of few words which don't match any pattern in the Alice knowledge base;
– when the query is grammatically incorrect and the pattern matching mechanism of Alice is not able to manage this particular mistake.

In the following a dialogue example with a query composed of one single word is reported:

Table 4. Percentage of correct, coherent and wrong answers on different 35 queries sets for the traditiona ALICE architecture

	ALICE		
	Correct	Coherent	Wrong
Queries of few words	11	0	89
Traditional queries	41	0	59
Greeting queries	100	0	0
Wrong queries	23	0	77

Table 5. Percentage of correct, coherent and wrong answers on different 35 queries sets for the LSAbot architecture

	LSAbot		
	Correct	Coherent	Wrong
Queries of few words	68	9	23
Traditional queries	63	11	26
Greeting queries	100	0	0
Wrong queries	100	0	0

Query: *Ada...*
Alice: *What is your favourite movie?*
LSAbot: *Ada is a trademark used for a structured computer programming language.*

Query: *Republic?*
Alice: *What is your real name?*
LSAbot: *Italy is a republic.*

Query: *World?*
Alice: *Can you tell me any gossip?*
LSAbot: *The world has been created by God*

Cleary, Alice is not able to answer since the question contains only few words and its form doesn't match any of its Aiml patterns. Hence, Alice tries to answer with default sentences, trying to switch the dialogue topic to another. On the contrary LSAbot can answer since it is based on a "conceptual" research within its knowledge base.

Another case in which LSAbot works better occurs when question's formulation does not comply with query patterns stored in Alice. In fact, LSAbot finds the answer conceptually related to the question regardless of its grammatical expression. In this example the question formulation complies with an Alice's query pattern :

Query: *What do you know about the C plus plus programming language?*
Alice: *It is a low-level programming language*
LSAbot: *C plus plus is the object oriented extension of the C programming language.*

while in this example the same question is expressed with other words, such that it does not comply with any Alice's query pattern.

Query: *Do you know C plus plus language?*
Alice: *Can you speak any other languages?*
LSAbot: *C plus plus is the object oriented extension of the C programming language.*

Finally, the following dialogue segments are related to the same question (complying with an Alice's query pattern): in the first segment the question is correctly written, while in the second segment it is wrongly written. LSAbot clearly shows better robustness than Alice with respect to errors in the question.

Query: *What is the Earth?*
Alice: *Earth, third planet from the sun, is our home.*
LSAbot: *Earth is a planet going around a star. Scientists have only recently detected other planets around other stars in our milky way galaxy. Do you like astronomy?*

while the following dialogue reports the wrong expression of the same query with the correspondent answers:

Query: *What is tha Earth?*
Alice: *Interesting question.*
LSAbot: *Earth is a planet going around a star. Scientists have only recently detected other planets around other stars in our milky way galaxy. Do you like astronomy?*

4 Conclusions and Future Work

In this paper an ad-hoc implementation of the LSA framework is presented, along with a conceptual interpretation of the automatically built data-driven semantic space. This interpretation allows to better design a natural human-like conversational agent, which overcomes limitations of traditional rule-based conversational agents, like Alice. The approach presented in this workregards aone-step interation butfuture works will run onthe lsa application for the whole interaction, in order to take in account the dialogue history and so the precedent contexts.

References

1. G. Pilato, G. Vassallo, M. Gentile, A. Augello, S. Gaglio LSA for Intuitive Chat-Agents Tutoring System *Proc. of International Conference on Methods and Technologies for Learning (ICMTL 2005),* Palermo 9-11 March 2005, in press
2. G. Pilato, G. Vassallo, A. Augello, M. Vasile, S. Gaglio Expert Chat-Bots for Cultural Heritage *IX Convegno della Associazione Italiana Intelligenza Artificiale Proc. of. Workshop Interazione e Comunicazione Visuale nei Beni Culturali*, 15 Sept. 2004, Perugia, Italy
3. Alice:Artificial linguistic computer entity. alice.sunlitsurf.com/alice/about.html.
4. Michael W. Berry, Susan T. Dumais, and Todd A. Letsche. Computational methods for intelligent information access. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, 1995.
5. J.K. Cullum and R.A. Willoughby. *Real rectangular matrices, in Lanczos Algorithms for Large Symmetric Eigenvalue Computations-Vol. 1, Theory.* Boston, MA: Brickhauser, 1985.

6. S. Peters D. Widdows. Word vectors and quantum logic experiments with negation and disjunction. In *Mathematics of Language, 8, Bloomington, Indiana, June 2003*, 2003.
7. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
8. P. Gardenfors. *Conceptual Spaces.* MIT Press, Bradford Books, Cambridge, MA, 2000.
9. M. Raubal. Formalizing conceptual spaces. In *Proc. FOIS 2004*, 2004.
10. R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, pages 10/187–228, New York 1996.
11. P.W. Foltz T.K. Landauer and D. Laham. An introduction to latent semantic analysis. *Discours Processes*, 25:259–284, 1998.
12. H. Zhu. Bayesian geometric theory of learning algorithms. *Proceedings of the International Conference on Neural Networks (ICNN'97)*, 2:1041–1044, 1997.
13. The Cyn Project. *http://www.daxtron.com/123start.htm?Cyn*
14. The OpenCyc Project, *http://www.opencyc.org*
15. O.Stock Language-based interfaces and their application for cultural tourism *AI Magazine*, March 22, pp 85-97, 2001
16. K.Mori, A.Jatowt, M.Ishizuka Enhancing conversational flexibility in multimodal interactions with embodied lifelike agents *Proc. of the 8th international conference on Intelligent user interfaces*, Miami, Florida, USA, pp.270-272, 2003
17. A.Koutamanis Azile: an intelligent verbal interface for the retrieval of pictorial architectural information *proc. of CIB W78 conference, Distributing Knowledge in Building.*, 2002
18. E.Mathews, G. Tanner Jackson, A.Graesser, N.Person and the Tutoring Research Group Discourse patterns in why/autotutor *proc. of Natural Language Generation in Spoken and Written Dialogue, AAAI Spring Symposium 2003*, March 24-26., 2003
19. M. Stede, D. Schlangen Dialogue Management by Topic Structure *Proceedings of Catalog '04 (The 8th Workshop on the Semantics and Pragmatics of Dialogue, SemDial04), Barcelona, Spain, July (ISBN 84-609-2205-7)*