

USING LATENT SEMANTIC ANALYSIS TO AID SPEECH RECOGNITION AND UNDERSTANDING

Lee McCauley

The University of Memphis
mccauley@memphis.edu

ABSTRACT

Generally, speech recognition engines can employ two different grammar methods, rule and dictation, to recognize an utterance. The purpose of these grammars is to constrain the search space in a way that anticipates the speaker's utterance. The research described in this paper attempts to maintain the accuracy of a rule grammar without limiting the speaker to rigorous phraseology. Latent Semantic Analysis (LSA) is used to connect specific grammar rules with the meanings underlying matching phrases resulting in utterances being matched to knowledge base elements even though the exact phrase did not match any grammar rule. A separate knowledge base is used to dynamically add or remove grammar rules in the speech recognition engine as the conversation context changes. Finally, a learning technique is used to create new regular expressions based on utterances that matched semantically through LSA.

1. INTRODUCTION

Speech recognition engines typically have two separate modes of operation. The first, called a dictation grammar, relies on complex statistical models while the second, called a rule grammar, requires simplified regular expressions that can be matched against incoming utterances. The purpose of both of these grammars is to model the speech patterns of users in a way that narrows the search space thereby increasing the likelihood of accurate speech recognition. Dictation grammars are typically less accurate than rule grammars, but allow for free-form recognition. Rule grammars, on the other hand, are highly accurate given that the user utters a phrase that precisely matches one of the regular expressions in the grammar. Described below is a method for supplementing rule grammars with Latent Semantic Analysis (LSA) in order to remove the need for rigorous phraseology. Ultimately, this enhanced recognition is only useful if the recognized text can be understood by the system. "Understanding" here is used to mean that the system can

instantiate appropriate knowledge base elements based on information gleaned from the text. In this model, Natural Language Understanding (NLU) and speech recognition are strongly interdependent. The current conversational context is maintained by the knowledge base pursuant to phrases or sentences recognized by the speech engine or LSA. The currently active rules to be used in the speech recognition engine are, in turn, provided by the knowledge base pursuant to the current conversational context.

This work is being conducted within the framework of an Intelligent Environment (IE). An IE is a space within which a user or users interact with the computer system without the encumbrance of interface devices. The primary mode of interface will, therefore, be voice input. In addition to recognizing user commands and questions, this environment is intended to be conversational in nature. There are three important aspects that must work in conjunction to allow this system to perform as intended. The first is the recognition of unstructured utterances as semantically matching to knowledge base elements. The second is the maintaining and updating of the context based on knowledge base inference. Changing the context results in the modification of grammar rules in the speech recognition engine. The final important aspect is the learning of new grammar rules.

2. BACKGROUND

In a conversational system, such as the one being described here, the natural language understanding module has one responsibility. That responsibility is to provide an analysis of the user's utterance in order to update the knowledgebase and, thereby, increase what the system understands about the user and situation.

When a system is employed in a limited domain and has only a limited number of choices of what to say or do next, it need only classify the user's utterance according to what it needs to know in order to make that decision. For instance, if a simple travel advisor has just asked, "What airport will you be departing from?" then it might classify the user's utterance according to airports, cities and other

geographic regions. The main NLU technology utilized here is based on classification.

The other main NLU technology is based on first-order logic. It generally requires a grammatical and syntactic parser. The parsing process is computationally expensive and often requires that the parsed text be grammatically and syntactically correct in order to assure an accurate reading. Because of the conversational nature of the task domain, an advanced classification method will be used instead.

Classification approaches to NLU include statistical [e.g., 1, 2], information retrieval [e.g., 3, 4] and connectionist approaches [e.g., 5]. Notable among these approaches are those that are based on word co-occurrence patterns such as LSA [6, 7] and HAL [8]. LSA is an attractive approach because it is trained on untagged texts and is a simple extension to keyword based techniques.

2.1 Latent Semantic Analysis

LSA has been remarkably successful at a number of natural language tasks. In the arena of query-based document retrieval [7, 9], LSA was compared to a large number of research prototypes and commercial systems. The performance of LSA varied from equivalent to the best other method to 30% better, with an average improvement of 16% over the competitors. The next success of LSA was in its modeling of human performance in the TOEFL test developed by Educational Testing Service [6]. The LSA model answered 64.4% of the questions correctly, which is essentially equivalent to the 64.5% performance for college students from non-English speaking countries. Another of the recent successes is the repeated demonstration that LSA can grade the essays that college students write almost as well as human graders [10-12]. Furthermore, LSA has been very impressive in accounting for (1) the developmental acquisition of vocabulary words, (2) the classification of words into categories, (3) the extent to which context activates the meaning of a word, (4) the amount of learning that occurs when students with varying degrees of domain knowledge read a text, and (5) the extent to which sentences in text are coherently related to each other

To use LSA, one must first develop an LSA space, which acts as a lexicon. The space represents the "meaning" of a word as a vector in a space of K dimensions (where K is typically 100 to 300). The space is built automatically from a training corpus. The corpus consists of a large number of "documents," where a document could be sentence, a paragraph or a longer unit of text. One computes a co-occurrence matrix that specifies the number of times that word W_i occurs in document D_j . A standard statistical method, called singular value decomposition, reduces the large $W \times D$ co-

occurrence matrix to K dimensions. This assigns each word a K -dimensional vector in the space.

Given an LSA space, the similarity (i.e., similarity in meaning, conceptual relatedness, match) between two words is computed as a geometric cosine (or dot product) between the two vectors, with values ranging from 0 to 1. The similarity between two sentences (or longer texts) is computed by first representing the meaning of each sentence as a vector that is the weighted average of the vectors for the words in the sentence, then computing the similarity between those two vectors as a geometric cosine. The match between two language strings can be high even though there are few if any words in common between the two strings. Thus, LSA goes well beyond simple keyword matches because the meaning of a language string is partly determined by the company (other words) that each word keeps.

2.2 Maintaining Context

Knowing what a user might be thinking about at any moment is important in determining what exactly a user might be asking about and how the response might be constructed to utilize the individual's currently active knowledge/context. For example, a user might point to an area of a map and ask, "what is this?" Depending on previous discussion and what the system believes the user is interested in, the system may interpret the user's question as referring to a country, a region or a city. Similarly, the responses generated by the system can be more natural if appropriate pronouns are used and commonly held or irrelevant knowledge is omitted. For instance, while looking at a map of Tennessee a user might ask, "where is Beale Street?" An appropriate response might be, "in Memphis." Note that the response did not mention the state, since that was in the shared context, nor did it mention the country, the latitude and longitude, or driving directions. Under different circumstances any of these responses might have been quite valid.

Some intelligent environments, such as the Smart Home [13], attempt to recognize individual users and represent them separately. Here we hope to represent humans in general for the explicit purpose of maintaining a common context between the user and the system. Models in cognitive psychology [14-17] give us a starting point for the creation of what a human might be thinking about during the course of their interaction with the system. The Construction-Integration model [16, 17], for example, describes the contents of a person's working memory during the course of reading. As the author points out, this model can be applied to information acquisition in general.

2.3. Task Domain

Even within such a diverse project, or perhaps especially in such a project, a specific domain must be chosen in order to focus research efforts and to allow for meaningful evaluation of the results. The domain chosen for these initial steps is a general questioning answering system. The user should be able to vocally ask the system a question from a small set of topics and receive a contextually relevant and accurate response.

Since the Intelligent Environment Lab is still in its inception stage, this domain will provide us with an opportunity to design and implement the main architecture along with voice input, voice output, and the knowledge representation structure discussed above. The main question answering knowledge base will come from the START system developed at MIT [18]. This system provides a method for entering natural language questions and receiving answers from any number of sources presented as an HTML page. It does not provide, however, for disambiguation of questions, translation of contextually relevant questions (such as, "what does that mean") or a method of directly answering the user's question in a way that is natural and contextually appropriate.

This task domain will also provide several nontrivial challenges whose solutions will significantly benefit the field of computer science. These challenges include:

- 1) the increase of the accuracy of voice recognition software based on contextual information and system expectations,
- 2) maintaining a model of the user based on the interaction and previous experience
- 3) the intelligent reformulation of a user's query, and
- 4) the creation of an English language response that is succinct and appropriate for the context.

3. HOW IT WORKS

Understanding spoken speech really requires two very different capabilities. The first is the translation of the sound patterns into textual representations. This process is referred to as speech recognition and, in our model, will contribute significantly to natural language understanding. While this research will not delve directly into the area of audio to text conversion, a primary focus is how to modify the grammar rules used as guidance to the speech recognition engine. We have chosen to use the commercially available Dragon Naturally Speaking™ software package as our speech recognition engine. Like most commercially available speech recognition products, Dragon Naturally Speaking™ can translate audio streams to text using two modes. One mode uses what is called a dictation grammar. Dictation grammars tend to use complex statistical models and a great deal of ongoing

research is being devoted to creating better models. Unfortunately, without completely replacing the dictation grammar there is no prescribed way to update or modify the grammar. The other mode used by most speech recognition engines uses a rule grammar. Rule grammars use regular expressions to constrain the possible translation of utterances. Unlike dictation grammars, rule grammars can easily be modified at runtime. The regular expression rules in this type of grammar can also be mapped to knowledge base tokens thereby providing a first step towards *understanding* the text. Understanding the text refers to translating it into the common knowledge format of the system.

Using the rule grammar described above is the most direct way to map incoming utterances to knowledge base elements. Regular expressions are matched against the text supplied by the speech recognition engine to determine if they satisfy the associated rule. If so, then any modifications to the knowledge base dictated by that rule are instigated. The speech recognition engine also provides a method for defining and dynamically activating its own pattern recognition rule sets at runtime. We will use this facility to interject, based on context, expected utterances predicted by the knowledgebase into the rule base of the speech recognition software thereby significantly increasing the likelihood of accurate recognition of those phrases. For example, suppose that the user has mentioned Albert Einstein. Expected phrases or partial phrases related to Albert Einstein, like "theory of relativity" or "speed of light," can be added to the rule base. By adding specific phrases to the list of possibilities we are actually narrowing the initial search space of interpretations of the audio stream. This both speeds up recognition, since these rules can be parsed very efficiently, and increases accuracy. Keep in mind that only a subset of the available rules is considered as possible matches for any given utterance. Valid rules might, therefore, not be active causing erroneous or incomplete knowledge to be inferred from an utterance.

The key to making this approach work is predicting key elements of upcoming utterances. Although we will start with a set of hand-coded rules, the likelihood of these rules being correct will be assessed by the system based on their actual use. This is simply a matter of updating statistical information relating whether a particular rule was used during the periods when it was considered to be in context. Statistical information will also be gathered to determine if a particular rule should have been considered in context but was not. This second measure is a bit difficult to obtain and could require a great deal more processing than is feasible for a system that must react in real time. Essentially, what needs to be gleaned is whether the user uttered a phrase that was encoded by a rule that was not active at the time of the utterance. This requires checking every utterance against every rule in the system – the active rules in context as well as all inactive

rules. Needless to say, this defeats part of the reason for using subsets of rules in the first place. The solution for this is to store utterances along with context states until the system is not actively engaged and do the comparisons during these periods. Results of these analyses would be used to determine if new rules need to be created in order to add an utterance to a given context. In addition to increasing the likelihood that the speech recognition engine will correctly recognize that pattern in that context, this also increases the system's understanding of what that utterance means by dictating knowledgebase modifications that will occur as a result of its recognition.

The second way that we will approach the problem of natural language understanding is to use Latent Semantic Analysis [6, 7] to extract and match knowledgebase elements to arbitrary utterances. LSA is a method of extracting the semantic meaning of words, sentences, or even paragraphs taken from a large corpus of text. Words and sentences can then be compared computationally based on their semantics. The basic idea is to decompose sentences into sets of knowledgebase encoded elements – the rules discussed above. This is the same task performed by text parsers but using statistical methods applied to very large texts. In the LSA approach, all domain knowledge is represented as vectors in LSA space regardless of whether it is a word, a question, a statement or a knowledgebase element. When the user speaks, each word is looked up in the LSA space, and its vector retrieved. The vector to represent the whole utterance is computed as the weighted average of the vectors representing the words. This vector is then compared to the knowledgebase-appropriate meanings (this process is described in more detail below). If the utterance vector is close enough in LSA space to the knowledgebase vector, then the system infers that the user has expressed the basic idea encoded in that knowledgebase element. Depending on what the knowledgebase rule is, the system is likely to update its model of the current user (shared knowledge, current context, etc.) and might respond to the user's query.

Overall, the process is remarkably simple. However, a few points have been glossed over that deserve fuller discussion, namely (1) constructing the LSA space, (2) calculating the match between two vectors, and (3) representing knowledgebase elements as LSA vectors.

An LSA space with K dimensions can be developed automatically given a sufficiently large corpus of texts. Since the system may be asked to converse on almost any topic, we will use one full, unabridged encyclopedia. Additional texts may be required if performance is not adequate. Analyses will be performed that assess the relationship between corpus size and performance within this domain.

There are three steps in computing the K dimensions for a corpus of texts with LSA. These steps are specified below.

(1) Preparation of a Word by Text rectangular matrix. LSA first prepares a large rectangular co-occurrence matrix that specifies the number of times that word W_i occurs in text T_j . A cell in the matrix is designated as $fr(W_i, T_j)$. The matrix is extracted from all of the words and texts in the entire corpus. It is possible to define a basic text unit (i.e., a document) as either a sentence or paragraph in the corpus. We will start out defining each sentence in the corpus as a text document.

(2) Transformation of cell values. Each cell frequency is transformed in two ways. First, the frequency (plus 1.0) is converted to its logarithm: $\log [fr(W_i, T_j) + 1]$. Second, there is a computation that estimates the relative distinctiveness of the word to a particular text, relative to the alternative texts. For example, the information-theoretic measure of entropy is computed for each word ($-\sum p \log p$) over all entries in its row and then the cell entry is divided by the row entropy value. This value increases to the extent that a word appears in a particular text and not in alternative texts.

(3) Singular Value Decomposition (SVD). SVD decomposes the large rectangular Word by Text matrix into the product of three component matrices. We refer to the large matrix as $\{X\}$ and the three component matrices as $\{W\}$, $\{S\}$, and $\{P\}$. LSA determines a best-fit set of component matrices that approximately reproduces $\{X\}$. That is, $\{X\} = \{W\}\{S\}\{P\}$. The $\{W\}$ matrix maps the set of words onto the set of K dimensions (i.e., functional features, factors). If there are N words and K dimensions, this would be an N by K matrix with each cell having a weight for a word-dimension combination (capturing the extent to which a word possesses a functional feature). $\{S\}$ is a vector with K values that weights the generic importance of each of the K dimensions. $\{P\}$ is a K by T matrix that maps the K dimensions onto the set of T texts. Therefore, the Word by Text matrix is reduced to K dimensions that serve as functional factors/features in the domain knowledge. It should be noted that there is an optimal number of dimensions that fits data in tests of LSA. For example, in Landauer's tests on the corpus of encyclopedia articles and the TOEFL data, 300 dimensions provided better fits to the data than 100 dimensions and 500 dimensions. We will need to experiment to determine the appropriate dimensions for our task.

After the LSA space is constructed, one can compute the conceptual relatedness between any two bag of words i.e., $\text{sim}(A,B)$. A bag contains one or more words, without any information about the order of the words in the text. In some LSA applications, the function words and other high frequency words (e.g., the, are, it) are removed because they are not distinctive words that discriminate texts. The results are frequently unaffected when these nondistinctive high frequency words are retained, but the role of high frequency words is still a matter of debate and research.

The two most common ways of computing relatedness are the cosine match and the dot product [19]. These two methods normally provide very similar results. Consider the relatedness of two words, X and Y. There is a K-dimensional vector for word X that is extracted from the {W} matrix: $X = (x_1, x_2, \dots, x_k)$. There is also a K-dimensional vector for word Y: $Y = (y_1, y_2, \dots, y_k)$. The dot product (XY) is the inner product of the two vectors: $XY = x_1y_1 + x_2y_2 + \dots + x_ky_k$. The dot product is a scalar (not a vector) that reflects the extent to which the two words are conceptually related. The length of vector X, designated as XX, is the square root of the dot product of vector X with itself: $[XX = x_1x_1 + x_2x_2 + \dots + x_kx_k]^{1/2}$. Similarly, there is a length of vector Y, designated as YY. The cosine match between X and Y is computed as follows: $\cos(X,Y) = XY/(XX * YY)$. When the cosine match is used, the values can vary from -1 to 1 (but values vary in practice from 0 to 1). The computations are readily extendible to cases when two or more words are in each bag of words. A bag of 2 or more words is a weighted average of the vectors of the words it contains.

Knowledgebase elements are represented internally as vectors in LSA space. Here, we need to map an arbitrary utterance to a preexisting rule that then states what knowledge needs to be updated and what actions need to be taken, if any. This is the same task as was performed by the rule-based approach described at the beginning of this section, but, with the LSA approach, the phrases spoken do not necessarily have to match precisely as long as their meanings, expressed as vectors in the LSA space, are similar enough. Unfortunately, there's not a one-to-one correspondence between a rule in the speech recognition engine and a vector in the LSA space. A given utterance may match to a combination of rules in the grammar. For example, the utterance "who is Albert Einstein," might match to a pair of rules such as the following:

- (1) <whoQuestion> = 'who is <person>'
- (2) <person> = 'Newton' | 'Einstein' | 'Albert Einstein' | 'Copernicus'

Stated loosely, these two rules recognize the words "who is" followed by any of the people listed in (2). This very simple combination recognizes four different sentences. LSA would be able to match the utterance against any of the instantiations of the rules, but there is no LSA vector that would match to (1) or (2) individually. Instead, a LSA vector is generated for each possible rule instantiation. The token bindings that went into each rule instantiation are stored along with the LSA vector. Therefore, when an arbitrary utterance is received, its LSA vector is match against the instantiations of all of the grammar rules currently in context. If a match is found then the tokens associated with that LSA vector (that particular rule instantiation) are obtained and said to the knowledgebase just as if the user had uttered the phrase exactly as prescribed in the grammar rule. It would be impractical,

however, to construct these vectors by hand. Instead, the system automatically generates all possible instantiations for a given rule. As new utterances are encountered that do not trigger a specific rule in the grammar but are matched via the LSA method, analysis is performed in order to determine how the rules might be updated to account for this new example.

4. EVALUATION

We plan on conducting two kinds of evaluations: formative evaluations and summative evaluations. Each is discussed below.

The formative evaluations are intended to guide user interface design and tool development as well as expose any flaws in the assumptions about user behavior. These evaluations will be in the form of "Wizard of Oz" sessions conducted with volunteers from outside the project. The volunteer will interact with the interface as though the system were fully functional. However, all input will be fed to an unseen human experimenter who will provide the response to the user's question. The text of the response will be input directly into a talking head module, which will output synthesized speech along with the appropriate animation. After the session is over, we will interview the volunteer user to determine how they felt about the system's performance, the talking head, the accuracy of the responses, etc.

The first thing that this type of evaluation will provide us with is feedback on what type of information we need to be able to acquire from the user. The human responding to the user will not get any additional data that the system would not get. If it turns out that the human "wizard" behind the scenes needs additional information, then we will need to provide methods for getting that information to the system as well. We will also get data on the usefulness of the talking head and other user interface elements. Finally, this evaluation will provide us with a benchmark for future tests of the complete system. By setting the standard based on human intelligence, we will get a good measure as to how well our system should be able to perform.

The summative evaluations are designed to benchmark the overall performance of the system. We will again use volunteers outside of the project. This time they will interact with the completed system. After the session is over they will be asked the same questions as were asked of the first set of volunteers in the "Wizard of Oz" tests. By comparing the results of these tests, we will get an idea of how well the system performed and what might still need to be modified. The summative evaluations as just described may occur at several stages of the project. This will not only provide guidance for future work but will also tell us how effective previous modifications had been.

5. CONCLUSION

In an effort to free users from the constraints of the desktop, intelligent environments are being created that allow for device-free interaction. A key piece to this endeavor will be the accurate understanding of natural human language. Described above is a method for the enhancement of speech recognition and natural language understanding through the coupling of highly accurate rule-based grammars with a powerful categorization technique. Together they should allow for accurate recognition and understanding without the constraints of strict phraseology. In addition to LSA, a learning mechanism will be used to infer new grammar rules. This method can do no worse than using rule grammars alone, but it remains to be seen exactly what the practical benefit will be. Emphasis will be placed on detailed evaluation regimens that will tease out this information along with a host of other important information.

6. REFERENCES

- [1] Sanker, A. and Gorin, A., "Adaptive Language Acquisition in a Multi-Sensory Device," *IEEE Transactions on Systems, Man and Cybernetics*, 1993.
- [2] Charniak, E., *Statistical Language Analysis*. Cambridge, MA: Cambridge University Press, 1993.
- [3] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. C., "Improving an Intelligent Tutor's Comprehension of Students with Latent Semantic Analysis," presented at Artificial Intelligence in Education '99, Le Mans, 1999.
- [4] Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and Group, T. R., "Using Latent Semantic Analysis to Evaluate the Contributions of Students in Autotutor," *Interactive Learning Environments*, vol. 8, pp. 149-169, 2000.
- [5] Miiikkulainen, R., "Subsymbolic Case-Role Analysis of Sentences with Embedded Clauses," *Cognitive Science*, vol. 20, pp. 47-74, 1996.
- [6] Landaur, T. K. and Dumais, S. T., "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge," *Psychological Review*, vol. 104, pp. 211-240, 1997.
- [7] Dumais, S. T., "Latent Semantic Indexing (LSI) and TREC-2," in *National Institute of Standards and Technology Text Retrieval Conference*, D. Harman, Ed.: NIST, 1994.
- [8] Burgess, C., Livesay, K., and Lund, K., "Explorations in Context Space: Words, Sentences, Discourse," *Discourse Processes*, vol. 25, pp. 211-257, 1998.
- [9] Deerwester, S., Dumais, S. T., Fumas, G. W., Landaur, T. K., and Harshman, R., "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391-407, 1990.
- [10] Foltz, P. W., Britt, M. A., and Perfetti, C. A., "Reasoning from Multiple Texts: An Automatic Analysis of Readers' Situation Models," in *Proceedings of the 18th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, 1996, pp. 110-115.
- [11] Landaur, T. K., Foltz, P. W., and Laham, D., "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, pp. 259-284, 1998.
- [12] Wolfe, M. B. W., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., and Landauer, T. K., "Learning from Text: Matching Readers and Texts by Latent Semantic Analysis," *Discourse Processes*, vol. 25, pp. 309-336, 1998.
- [13] Intille, S. S., "Designing a Home of the Future," *IEEE Pervasive Computing*, pp. 80-86, 2002.
- [14] Glenberg, A. and Robertson, D. A., "Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning," *Journal of Memory and Language*, vol. 43, pp. 379-401, 2000.
- [15] Hexmoor, H., "A Cognitive Model of Situated Autonomy," presented at PRICAI-2000 Workshop on Teams with Adjustable Autonomy, Australia, 2000.
- [16] Kintsch, W., "The Use of Knowledge in Discourse Processing: A Construction-Integration Model," *Psychological Review*, vol. 95, pp. 163-182, 1988.
- [17] Kintsch, W., *Comprehension: A Paradigm for Cognition*. Cambridge, MA: Cambridge University Press, 1998.
- [18] Katz, B., "From Sentence Processing to Information Access on the World Wide Web," presented at AAAI Spring Symposium on Natural Language Processing for the World Wide Web, Stanford University, 1997.
- [19] Rehder, B., Schreiner, M. E., Wolfe, M. B. W., Laham, D., Landauer, T. K., and Kintsch, W., "Using Latent Semantic Analysis to Assess Knowledge: Some Technical Considerations," *Discourse Processes*, vol. 25, pp. 337-354, 1998.