Chapter X

TOWARDS A NEW GENERATION OF CONVERSATIONAL AGENTS BASED ON SENTENCE SIMILIARTY

Karen O'Shea, The Intelligent Systems Goup, Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester M1 5GD. Email: k.oshea@mmu.ac.uk

Dr. Zuhair Bandar, The Intelligent Systems Goup, Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester M1 5GD. Email: z.bandar@mmu.ac.uk

Dr. Keeley Crockett, The Intelligent Systems Goup, Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester M1 5GD. Email: k.crockett@mmu.ac.uk

1. INTRODUCTION

The concept of "intelligent" machines was first conceived by the British mathematician Alan Turing [1]. The imitation game, known as the "Turing Test", was devised to determine whether or not a computer program was "intelligent". This led to the development of the Conversational Agent (CA) [ref] – a computer program that can engage in conversation using natural language dialogue with a human participant.

CAs can exist in two forms: "Embodied" agents [2] possess an animated humanoid body and exhibit attributes such as facial expressions and movement of eye gaze. "Linguistic" agents [3], [4] consist of spoken and/or written language without embodied communication. One of the earliest text-based CAs developed was ELIZA [3]. ELIZA was capable of creating the illusion that the system was actually listening to the user simply by answering questions with questions. This was performed using a simple pattern matching technique, mapping key terms of user input onto a suitable response. Further advancements on CA design led to PARRY [4], capable of exhibiting personality, character, and paranoid behavior by tracking its own internal emotional state during a conversation. Unlike ELIZA, PARRY possessed a large collection of tricks, including: admitting ignorance by using expressions such as "I don't know" in response to a question; changing the subject of the conversation or rigidly continuing the previous topic by including small stories about the theme [4]. CAs can also engage in social chat and are capable of forming relationships with a user. ALICE [5], an online chatterbot and Infobot [6] are just two such examples. By conversing in natural language these CAs are able to extract data from a user, which may then be used throughout the conversation.

Considerable research has been carried out on the design and evaluation of embodied CAs [2], [7]; however, little work appears to have been focused on the actual dialogue. This paper will concentrate on text-based CAs and the development and evaluation of high-quality dialogue.

Most text-based CA's scripts are organized into contexts consisting of a number of hierarchically organized rules. Each rule possesses a list of structural patterns of sentences and an associated response. User input is then matched against the patterns and the pre-determined response is sent as output. Infobot [6] is one such CA capable of interpreting structural patterns of sentences. However, every combination of utterances must be taken into account when constructing a script – an evidently time-consuming, high maintenance task, which undoubtedly suggests scope for alternative approaches. It is, therefore, envisaged that the employment of sentence similarity measures could reduce and simplify CA scripting by using a few prototype natural language sentences per rule.

Two successful approaches to the measurement of sentence similarity are: "Latent Semantic Analysis" (LSA) [8] and "Sentence Similarity based on Semantic Nets and Corpus Statistics" [9]. LSA is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text [8]. A word by context matrix is formed based on the number of times a given word appears in a given set of contexts. The matrix is decomposed by "Singular Value Decomposition" (SVD) into the product of three other matrices, including the diagonal matrix of singular values [10]. This dimension reduction step collapses the component matrices so that words that occurred or did not occur in some contexts now appear with a greater or lesser frequency [8]. Reconstruction of the original matrix enables LSA to acquire word knowledge among large numbers of contexts. Although LSA makes no use of syntactic relations, it does, however, offer close enough approximations of people's knowledge to underwrite and test theories of cognition. "Sentence Similarity based on Semantic Nets and Corpus Statistics" will be employed as the measure in this research and will be described in further detail in Section II.

This chapter is organized as follows: Section II will describe and illustrate the sentence similarity measure; Section III will describe two CAs and their scripting methodologies; Section IV will present an experimental analysis of the two approaches; Section V will evaluate the results and Section VI will conclude and highlight areas for further work.

2. SENTENCE SIMILARITY MEASURE

"Sentence Similarity based on Semantic Nets and Corpus Statistics" [9] – should this be in quotes ? is a measure that focuses directly on computing the similarity between very short texts of sentence length. Through the use of a lexical/semantic knowledge-base such as WordNet [11], the length of separation between two words can be measured, which in turn, can be used to determine word similarity. The synset – a collection of synonyms – at the meeting point of the two paths is called the subsumer. The depth of the subsumer is similarly measured by counting the levels from the subsumer to the top of the hierarchy. Li *et al.* [9], [12] proposed that the similarity between two words be a function of the attributes: path length and depth. The algorithm initiates by combining the two candidate sentences (T1 and T2) to form a joint word set using only distinct words. For example:

T1 = Mars is a small red planet
T2 = Mars and Earth orbit the sun

A joint word set 'T' is formed where:

T = Mars is a small red planet and earth orbit the sun

As a result, each sentence is represented by the use of the joint word set with no surplus information. Raw semantic vectors are then derived for each sentence using the hierarchical knowledge-base WordNet [11], in order to determine the separation between words. Taking a non-linear transfer function as an appropriate measure, the following formula denotes a monotonically decreasing function of l, where l = path length between words and α is a constant.

$$f(l) = e^{-\alpha l} \quad (1)$$

As for the depth of the subsumer, the relationship of words at varying levels of the hierarchy must be taken into consideration. For example, words at the upper layers are far more general and less semantically similar than words at lower layers [9]. Therefore, subsuming words at upper layers must be scaled down whereas words at lower layers must be scaled up, resulting in a monotonically increasing function of h, where h = depth of subsumer and β is a constant.

$$f(h) = (e^{\beta l} - e^{-\beta h}) / (e^{\beta l} + e^{-\beta h}) \quad (2)$$

As such, the raw similarity $s(w1, w2)$ between two words is calculated as:

$$s(w1, w2) = e^{-\alpha l}. (e^{\beta l} - e^{-\beta h}) / (e^{\beta l} + e^{-\beta h}) \quad (3)$$

where $\alpha = 0.2$ and $\beta = 0.45$.

Each word is then weighted, that is, assigned an information content value, based on its significance and contribution to contextual information. By combining the raw semantic vector $s(w1, w2)$ with the information content of each word, $I(w1)$ and $I(w2)$, semantic vectors are created:

$$s_i = s(w1, w2) . I(w1) . I(w2) \quad (4)$$

Finally, the semantic similarity Ss between two sentences, $s1$ and $s2$, is calculated as:

$$Ss = si1.si2 / \sqrt{si1} / \sqrt{si2} \quad (5)$$

where $si1$ is the resultant semantic vector of sentence 1 and $si2$ is the resultant semantic vector of sentence 2.

Word order also plays an active role in sentence similarity. Each word is assigned a unique index number which simply represents the order in which the word appears in the sentence. For example, take the following sentences denoted T1 and T2:

T1 = The cat ran after the mouse

T2 = The mouse ran after the cat

A joint word set 'T' is formed where:

T = The cat ran after the mouse

Each sentence is than compared to that of the joint word set. If the same word is present – or if not, the next most similar word – then the corresponding index number from T1 will be placed in the vector, r1. As such, the word order vectors r1 and r2 for the example sentence pair T1 and T2 would be formed as follows:

r1 = {123456}
r2 = {163452}

Therefore, word order similarity *Sr* is calculated as:

$$Sr = 1 - \sqrt{(r1 - r2)} \,/\, \sqrt{(r1 - r2)} \qquad (6)$$

Finally, the sentence similarity is derived by combining both semantic similarity and word order similarity. The overall sentence similarity between two sentences *S(T1*, *T2)* is calculated as:

$$S(T1, T2) = \delta Ss + (1 - \delta)\, Sr \qquad (7)$$

where δ takes into account that word order plays rather a less significant role when determining sentence similarity.

3. SCRIPTING METHODOLOGIES

Two types of CA and their scripting methodologies will now be described. First, the traditional approach [6] employing structural patterns of sentences and second, the new proposed approach employing natural language sentences. The first approach requires considerably more human intervention and skill in contrast to the opposing second approach, which will be highlighted in the subsequent sections.

3.1 Traditional Approach

Traditional approaches [6] interpret structural patterns of sentences by using scripts consisting of rules organized into contexts. A context may be described as a collection of rules relating to a particular topic. Each context contains a number of hierarchically organized rules each possessing a list of structural patterns of sentences and an associated response. A user's utterance is then matched against the patterns and the associated response is "fired" (selected) and sent as output. The following steps 1-3 illustrate the procedure.

1. Natural language dialogue from the user is received as input and is matched to a pattern contained in a rule.
2. Match-strength is calculated based on various parameters, including the activation level of each rule.
3. The pattern with the highest strength is thus 'fired' and sent as output.

Scripts are constructed by first assigning each rule a base activation level, a number between 0 and 1. The purpose of the activation level is to resolve conflicts when two or more rules have patterns that match the user's input [13]. The scripter must then decide which patterns a user may send in response to output. Each pattern is assigned a pattern-strength value, typically ranging between 10 and 50. For example, a rule may be constructed as follows:

```
<Rule_01>
a:0.5
p:50 *help*
p:50 I do not *<understand-0>*
r: How can I help you
```

where a = activation level, p = pattern strength/pattern, r = response.

Patterns can also contain wildcard elements "*" which will match with one or more consecutive characters. In addition, the macro "<understand-0>" enables the scripter to incorporate stock patterns into a rule [6]. Writing such scripts is a time-consuming and highly skilled craft [14]. For example, a script typically consists of a number of contexts each denoting a particular topic of conversation. Each context contains a hierarchically organized list of rules each possessing a collection of structural patterns of sentences. However, modifying one rule or introducing a new rule into the script invariably has an impact on the remaining rules. As such, a reassessment of the entire script would be warranted, without which would render the CA futile. The scripter is, therefore, required to remember the rankings of the rules and predict how the introduction of new rules will interact with existing rules [13]. The huge overhead and maintenance of this type of scripting undoubtedly suggests scope for an alternative approach.

3.2 Sentence Similarity Approach

The new proposed approach will maintain the same script as that of the traditional approach; however, all patterns will be replaced with natural language sentences. This considerably reduces the burden and skill required to produce CA scripts. Through the use of a sentence similarity measure [9], a match is determined between the user's utterance and the natural language sentences. The highest ranked sentence is fired and sent as output. The following steps 1-3 illustrate the procedure.

1. Natural language dialogue is received as input, which forms a joint word set with each rule from the script using only distinct words in the pair of sentences. The script is comprised of rules consisting of natural language sentences.
2. The joint word set forms a semantic vector using a hierarchical semantic/lexical knowledge-base [11]. Each word is weighted based on its significance by using information content derived from a corpus.
3. Combining word order similarity with semantic similarity the overall sentence similarity is determined. The highest ranked sentence is 'fired' and sent as output.

The proposed scripts are simply constructed by assigning a number of prototype natural language sentences per rule. For example, one such rule may be constructed as follows:

<Rule_01>
I need help
I do not understand
r: How can I help you

where s = sentence and r = response.

The precise number of sentences per rule will start at one and increase to "n" where "n" is determined by experimental analysis. However, it is expected that the value of "n" will be small and significantly less than the number of patterns used in traditional scripting methodologies.

4. EXPERIMENTAL METHODOLOGY

4.1 Domain

The real world domain is concerned with advising students at University on debt management and the payment of tuition fees. For the purpose of experimentation, one script, which consists of 18 rules, was taken from a substantially extensive script developed by Convagent Ltd. [6]. This sample script was selected purely for its size, suitability and relevancy.

4.2 Experiments

Two sets of experiments were undertaken to compare the traditional scripted CA and the sentence similarty based CA. The first experiment examined the traditional approach using structural pattern of sentences [6]. The rules consisted of patterns, which were in some cases consolidated with macros. This accumulated the count of patterns into the 100s. In comparison, the second experiment examined the new proposed approach, re-structured using natural language sentences. Through the use of a sentence similarity measure, the level of scripting was reduced to a couple of generic prototype sentences. Table 1 illustrates the scripting by the two approaches for the same rule.

Table X-1. Example scripting by two approaches to CA design

Approach One Traditional Pattern Scripting	Approach Two New Proposed Scripting
<Rule_01>	<Rule_01>
a:0.5	s: I need help
p:50 *<confused-0>	s: I do not understand
p:50 *<confused-0>*	s: This is confusing
p:50 *<sure-neg-0>*	r: How can I help you
p:50 *<sure-neg-1>*	
p:50 *help*	
p:50 *not *<understand-0>*	
r: How can I help you	

Approach one consists of structural patterns of sentences consolidated with macros. The macro “<confused-0>” contains 16 patterns. Similarly, the macros “<confusing-0>”, “<sure-neg-0>”, “<sure-neg-1>” and “<understand-0>” contain a further 8, 21, 10 and 13 additional patterns respectively. This accumulates the final number of patterns, including the patterns “*help*” and “*not*” to 70. Approach two, however, replaces the above patterns for three generic natural language sentences: “I need help”, “I do not understand” and “This is confusing”.

5. RESULTS AND DISCUSSION

The first experiment examined the traditional approach using structural patterns of sentences [6], while the second approach examined the new proposed approach using natural language sentences. The experiments entailed sending as input 18 domain-specific user utterances. The 18 selected user utterances were deemed representative of the domain. The resulting output, that is the fired pattern/sentence, for the 18 cases are displayed in table 2.

Table X-2. Results of user input for two approaches to CA design

Utterance User Input	Approach One Traditional Pattern Scripting Fired Pattern	Approach Two New Proposed Scripting Fired Sentence
1. I am having trouble with my benefactor	*	I have a problem with my sponsor
2. I want assistance	* will pay *	I need help
3. I have not quit my course	*I* not *quit* course	I have not received my funding
4. Could I pay a tiny quantity of the cost	Could I *	I would like to pay a small amount of the fee
5. I have no finance	* no *	I have no funding
6. I have already paid the fee	* have paid *	I could pay part of the fee
7. I have a different reason	* have a *	It is none of those reasons
8. I have not sent any payment	* not sent * payment *	Payment has not been sent
9. I am no longer studying at the University	* no *	I am still attending my course
10. I have to wait for my career development loan draft	* wait * loan	I am still waiting for my loan
11. I have not sent any payment however I have not quit	* however *	Payment has not been sent in the post
12. Could you repeat the choices	*	Please repeat the option
13. I have not yet obtained my student loan	* student loan *	I have not received my student loan
14. My local education authority appraisal has been	*	I have not received my local education authority

Utterance User Input	Approach One Traditional Pattern Scripting Fired Pattern	Approach Two New Proposed Scripting Fired Sentence
delayed		assessment
15. My hardship finance has failed to arrive	* hardship *	I have not received hardship funding
16. I am having trouble with my direct debit	* direct debit *	I have direct debit problems
17. I am broke	*	I am not at the University
18. I sent you the cash weeks ago	* sent *	Payment was sent in the post to the University last week

The results of the user utterances are as follows: The outputs generated after the input of user utterances 3, 6, 8, 10, 13, 15, 16, and 18 indicate a correct firing by approach one. As a result, approach one appears to have found a structurally comparable match. The outputs generated after the input of user utterances 1, 2, 4, 5, 7, 8, 10, 12, 13, 14, 15, 16, and 18 indicate a correct firing by approach two. As a result, approach two appears to have found sufficient semantic similarity between the user utterances and the corresponding natural language sentences.

The outputs generated after the input of user utterances 1, 2, 4, 5, 7, 9, 11, 12, 14, and 17 indicate a miss-firing by approach one. As a result, approach one appears to have failed to find an identical or comparable match to that of the user utterance. The outputs generated after the input of user utterances 3, 6, 9, 11, and 17 indicate a miss-firing by approach two. As a result, approach two appears to have failed to identify sufficient semantic similarity between the user utterances and the natural language sentences.

In the cases where approach one miss-fired, this was due to the script not possessing an identical or comparable structural match. This, however, may be rectified by incorporating the missing patterns into the script. In the cases where approach two miss-fired, this was invariably due to the user utterance containing an adjective or verb. The sentence similarity measure employed in this paper considers only one part-of-speech, in this case, nouns. As a consequence, input, other than that of nouns, will be disregarded and thus, somewhat hinder the measures performance. This, however, may be rectified by incorporating additional natural language sentences into the script. Furthermore, the sentence similarity measure could be adjusted so as to consider other parts-of-speech.

In totality, approach one correctly matched 8 out of 18 user utterances, whereas approach two correctly matched 13 out of 18 user utterances. Typically the number of patterns per rule for the traditional pattern script was between 50 and 200. In contrast, the average number of sentences per rule for the natural language script was three.

6. CONCLUSIONS AND FURTHER WORK

Most CAs employ a pattern-matching technique to map user input onto structural patterns of sentences. However, every combination of utterances that a user may send as input must be taken into account when constructing such a script. This paper was concerned with constructing a novel CA using sentence similarity measures. Examining word meaning rather than structural patterns of sentences meant that scripting was reduced to a couple of natural language sentences per rule as opposed to potentially 100s of patterns. Furthermore, results indicate good sentence similarity matching with 13 out of 18 domain-specific user utterances as opposed to that of the traditional pattern matching approach.

Further work will entail considerable development of the new proposed approach. The aim will be to incorporate the use of context switching whereby each context defines a specific topic of conversation. This would assist the approach to cope with negation of sentences, such as "I have paid" and "I have not paid". The CA will be robust, capable of tolerating a variety of user input. It is intended that a user evaluation of the two approaches to CA design will be conducted. Firstly, each approach would be subjected to a set of domain-specific utterances. Each CA would then compute a match between the user utterance and the rules within the scripts, firing the highest strength pattern/sentence as output. A group of human subjects would evaluate the scripts and their corresponding outputs in order to judge whether the correct pattern/sentence had been fired. This would provide a means for evaluating the opposing approaches and their scripting methodologies.

REFERENCES

[1] A. Turing, "Computing Machinery and Intelligence" Mind, Vol. 54, (236), 1950, pp. 433-460.

[2] D. W. Massaro, M. M. Cohen, J. Beskow, S. Daniel, and R. A. Cole, Developing and Evaluating Conversational Agents, Santa Cruz: University of California, 1998.

[3] J. Weizenbaum, ELIZA – "A Computer Program for the Study of Natural Language Communication between Man and Machine", Communications of the Association for Computing Machinery, Vol. 9, 1966, pp. 36-45.

[4] K. Colby, Artificial Paranoia: A Computer Simulation of Paranoid Process. New York: Pergamon Press, 1975.

[5] R. S. Wallace. (2008, February 01). ALICE: Artificial Intelligence Foundation Inc. [Online]. Available: http://www.alicebot.org

[6] D. Michie, and C. Sammut, Infochat™ Scripter's Manual. Manchester: Convagent Ltd, 2001.

[7] G. A. Sanders, and J. Scholtz, "Measurement and Evaluation of Embodied Conversational Agents", in Embodied Conversational Agents, Ch 12, J. Cassell, J. Sullivan, S. Prevost and E. Churchill ed., Embodied Conversational Agents, MIT Press, 2000.

[8] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis". Discourse Processes, Vol. 25 (2-3), 1998, pp. 259-284.

[9] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics", IEEE Transactions on Knowledge and Data Engineering, Vol. 18, (8), 2006, pp. 1138-1149.

[10] D. Landauer, D. Laham, and P. W. Foltz, "Learning Human-Like Knowledge by Singular Value Decomposition: A Progress Report", in Advances in Neural Information Processing Systems 10, M.I. Jordan, M. J. Kearns, and S. A. Solla, eds., Cambridge, Mass.: MIT Press, 1998, pp. 45-51.

[11] G. A. Miller, "WordNet: A Lexical Database for English", Comm. ACM, Vol. 38, no. 11, 1995, pp. 39-41.

[12] Y. Li, Z. A. Bandar, and D. Mclean, "An Approach for Measuring Semantic Similarity between Words using Multiple Information Sources", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, (4), 2003, pp. 871-881.

[13] C. Sammut, "Managing Context in a Conversational Agent", Electronic Transactions on Artificial Intelligence, Vol. 3 (7), 2001, pp. 1-7.

[14] D. Michie, "Return of the Imitation Game", Electronic Transactions in Artificial Intelligence, 2001.