EVALUATION OF SIMPLE (SEMANTICALLY IMPROVED MATCHING PROCEDURE FOR LANGUAGE ENABLED) CONVERSATIONAL AGENTS

by

ANIMESH ASHOK THAKRE

(Under the Direction of Dr. Kyle Johnsen and Dr. Eileen Kraemer)

ABSTRACT

The effectiveness of user interactions with a conversational agent largely depends on the relevance of responses that the system is able to generate for a user utterance. Existing systems primarily employ syntactic templates (*i.e.* grammar rules and word matching) to indirectly extract meaning from user input. A different method of extracting meaning from input is to determine the semantic distance between the words in one sentence and the words in another. Such a distance metric is made possible by semantic networks, such as WordNet, that link words on relatedness. We present an approach to match user input utterances to agent responses based on a semantic distance metric using the WordNet lexical database and propose a number of uses for our approach in developing conversational agent systems.

INDEX WORDS: Interactive Session, Trigger, Response, Virtual Agent, Semantic Score, False Positive, False Negative, WordNet.

EVALUATION OF SIMPLE (SEMANTICALLY IMPROVED MATCHING PROCEDURE FOR LANGUAGE ENABLED) CONVERSATIONAL AGENTS

by

ANIMESH ASHOK THAKRE

BE, NAGPUR UNIVERSITY, INDIA, 2008

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2011

EVALUATION OF SIMPLE (SEMANTICALLY IMPROVED MATCHING PROCEDURE FOR LANGUAGE ENABLED) CONVERSATIONAL AGENTS

by

ANIMESH ASHOK THAKRE

Major Professor:	Kyle Johnsen
	Eileen Kraemer
Committee:	Khaled Rasheed
	Robert Robinson

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2011

DEDICATION

I would like to dedicate this thesis to my parents, Ashok Thakre and Veena Thakre, for their unconditional love and support, and to my loving sisters Mona didi and Rashmi didi, who made my footing more secure along this difficult path.

ACKNOWLEDGEMENTS

I would like to acknowledge the help of my major professor Dr. Kyle Johnsen for his support and guidance. He has been a major figure for my education at UGA. I thank Prof Kraemer and Dr. Rasheed for helping me through my research by sharing their expert knowledge. And my special thanks to Prof Robinson for being my mentor. I would also like to thank my colleague Tyler Niles who helped me during my research in many ways.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Early in the history of modern computing, Alan Turing introduced his famous test, to determine when an artificial intelligence had reached human intelligence. The Turing test [Turing, 1950]centers around the concept that when a computer program, an intelligent agent, could fool a human being into believing that it was a real person, then the agent had achieved human-level intelligence. As of 2011, no conversational agent has passed this test for more than a few minutes. Despite this limitation, conversational agents have still been employed effectively in professional fields for education and training purposes. They are effective because they teach procedure, and are not necessarily designed to prove human-intelligence. In these systems, the general goal is to produce *reasonable* responses to user input, in order to allow trainees to practice the performance of a common interaction, such as a medical interview or interrogation. In such interactions, the user is essentially following a non-linear script; and as such, the conversational agent must also follow the script. In other words, when the user says "hello", the script might demand that the conversational agent respond with "hello", or "hello, how are you". This thesis presents SIMPLE (Semantically Improved Matching Procedure for Language Enabled), an algorithm that uses sentence similarity measures to improve the effectiveness of script-oriented conversational agents, and proposes a number of uses for the approach.

SIMPLE is directed towards linguistic agents that must match natural language user input to a highly non-linear script (*i.e.* one where possible inputs are numerous and may appear in any

order), to produce a predetermined response as output. Most conversational agents use a syntax matching technique to map each user utterance to an existing script and generate a response. However, users might provide utterances that are different from the exact utterances in the script by changing the structure (syntax) of a given sentence. As a result, a large number of possible variations of an utterance must be taken into account. A common approach to deal with utterance variations is to maintain a substantial corpus with possible user utterances and utterance variations. Then, as a new utterance is encountered, the syntactically closest corpus utterance is selected, and the predetermined response to this utterance is given. This approach does not scale well, as syntax can vary considerably with very little semantic variation. A potentially superior approach would be to augment a purely syntactical matching algorithm with semantic information. For example, "pain" and "hurt", while syntactically dissimilar are close semantically. This is the approach taken by SIMPLE. SIMPLE matches user input utterances to agent responses based on a semantic distance metric that leverages the WordNet lexical database.

User utterances act as an input for the system. For a given input utterance and each standardized response from the database, the system finds the most appropriate sense for each word by accessing a dictionary where senses are arranged in a hierarchical order (WordNet [Princeton]). The semantic similarity between the input utterance and each response is computed based on the semantic similarity of the word senses.

The system is targeted for use mainly in the field of education where conventional training methods have practical limitations. For example, healthcare professionals, such as medical students, must be trained for conversational scenarios, e.g. conducting a doctor-patient interview. Modern training techniques include lectures, textbook models, and often the hiring of real human subjects acting as patients [Dickerson, 2005]. The limitations with such training

models include time constraints, standardization, availability of the actors, actor diversity, monetary cost, and overhead of maintaining training facilities.

In our experimental studies, we have evaluated several techniques including only syntax matching, only semantic matching, and an approach that combines syntax as well as semantic matching. Results indicate that the combination of syntactical and semantic information provides greatly improved performance and accuracy in terms of generating relevant responses.

CHAPTER 2

LITERATURE REVIEW

2.1 History of Conversational agents

"Embodied" agents [Sanders, 2000] are agents that possess an animated humanoid body and possess attributes such as facial expressions and movement of eye gaze while Linguistic Agents [O"Shea K, 2008] [Cole, 1999] consist of spoken and/or written language without embodied communication. Our focus is mainly on linguistic agents, also called conversational agents. A conversational agent is a computer program that can participate in conversation using natural language dialogue with a human participant. ELIZA [Weizenbaum, 1996] is an example of the earliest text-based conversational agents. ELIZA"s purpose was to dupe the user into believing that the system was actually listening to the user like any other human. The system simply responded to questions by asking another question, which was an effective way of making the user believe that he was conversing with a real person. Eliza employed a pattern matching technique, mapping key terms of user input onto a suitable response. As years went by, conversational agents went through a complete transformation from being a question answering system, to agents capable of exhibiting personality, character and even paranoid behaviour. PARRY [Colby, 1975] is an example of such an evolved conversational agent that could track its own internal emotional state during a conversation and portray a distinctive personality. PARRY improved up on ELIZA with some distinguishing features, including: admitting ignorance by using expressions such as "I really don"t know" in response to a question; changing the subject of the conversation or rigidly continuing the previous topic by adding small stories about the

theme. The next milestone in the history of Conversational agents came with the emergence of the Internet era when conversational agents were made to engage in social chat and were able to form relationships with online users. ALICE [Wallace, 2008], an online chatterbot and Infobot [Michie, 2001] are just two such examples. These Conversational agents would extract data from users by conversing in natural language and then use this data to drive the conversation forward. Such agents would follow a predetermined script by asking questions that would reveal personal interests and likings of the user. These would then be matched with general topics of interests and the conversational agent would then continue the conversation by asking questions about these topics. Neither Eliza nor ALICE could pass the Turing test, yet ELIZA remains a milestone for being the first attempt at human-machine interaction while ALICE is one of the most evolved conversational agents that has won the Loebner prize, which is given to the most accomplished humanoid, talking robot. Alternative approaches to natural language conversational agents are multiple-choice agents [Brent Rossen, 2009 – Dickerson, 2005]. While restrictive in terms of experience variability, multiple-choice agents are extremely robust, content rich, and rapidly constructed. Moreover, multiple-choice systems are often authorable by end-users without the need for extensive programming support [Dickerson, 2005]. As a result, multiple choice systems have seen widespread use, where natural language systems have not. The goal of SIMPLE is to provide characteristics similar to multiple-choice agents in terms of robustness, ease-of-use, and authoring process, while enabling richer experiences through a natural language interface.

2.2 Conversational agents as trainers

Recently, conversational agents have been employed as virtual assistants in the fields of education and training. Roda *et al*. [Roda, 2001] highlighted some approaches for integrating conversational agents in a learning environment; the most fundamental approach was to use

conversational agents as advanced help assistants associated with a specific learning environment or tool. Conversational agents can perform as trainers that are equipped with specific domain knowledge or they could be used as role-playing actors in simulated experimental learning environments, such as the virtual patients created in the online tool Virtual People Factory [Brent Rossen, 2009]. Roda *et al.* applied Conversational Agents to operate in domains such as helping people learn to manage and share knowledge in organizations by continuously observing the actions of the user and then providing customized guidance and mentoring. Crocket *et al.* [O'Shea K, 2008] suggest that conversational agents can provide "tailored experiences" for teachers and students that enable curricula to meet new demands and ever-increasing scope. Conversation agents can build upon each individual student's strength, abilities and learning skills to provide support and improve overall learning experience. Teachers need to be trained for effective communication with students. Conversational agents can provide sustained motivation for learning these skills.

Conversational agents have also been employed for training in many professional fields (e.g. medicine, military, law-enforcement) [Brent Rossen, 2009] [Dickerson, 2005] [Kenny, 2007]. The vast majority of such systems rely on multiple-choice input instead of natural language input [Kenny, 2007]. While evidence [Kerly, 2007] suggests that cognitive domain learning does occur with these types of systems, from a training standpoint, multiple-choice input systems do not allow a person to practice the task of crafting utterances in a conversation. This is a higher-level skill than choosing from a list of possible utterances and is more difficult to train. However, in order to train students in such open dialogue, a conversational agent must, at a minimum, be able provide relevant responses to user utterances in a conversation. From our experience, response relevance is the most important factor in user trust of an open dialog

Embodied Conversational Agent training system. As discussed in the following chapter, SIMPLE agents are designed to specifically improve response relevance of conversational agents deployed conversational in training scenarios where high-quality dialogue is essential for learning.

CHAPTER 3

APPROACH

3.1 Introduction

The goal of any conversational agent is to engage the user in a sustained dialogue. This process depends on the ability of the conversational agent to produce quality responses for given user utterances. A conversational agent should be able to create an illusion that the user is interacting (however brief) with a real human. Maintaining a corpus of anticipated user utterances for improving the quality of responses is an approach that leads to a redundant system. We now discuss an approach that aims at improving the quality of generated responses while keeping the corpus size unchanged.

A trigger is a syntactic instance of either a question or a statement. Having all possible syntactic variations of a trigger present in a script introduces redundancy and degrades performance of the matching procedure. Also the task of scripting triggers corresponding to the standardized responses is cumbersome and both logistically difficult and time consuming [Kenny, 2007].

For example a trigger script may contain the following triggers mapped to the response "Hello",

Trigger 1: Hello

Trigger 2: How are you

A user may choose any of the following sentences as a possible utterance and expect the same response "*Hello*" from the system,

Utterance 1: Hi

Utterance 2: Hello

Utterance 3: How are you doing?

Utterance 4: How are you feeling today?

No matter which utterance is selected by the user, a system is needed that matches an utterance to one of the existing triggers and returns "Hello" as a response. The system is expected to match utterances "Hi" and "Hello" with trigger "Hello" while utterances "How are you doing" and "How are you felling today" must be matched with trigger "How are you". Therefore, we need an algorithm that could capture subtle semantic variations between utterances and existing triggers and produce a relevant response.

Most of the existing conversational systems use a syntactic measure to match user utterances using the corpus. Such systems fail to capture the semantic relationships among words and are thus more prone to generate false positive and false negative responses. For e.g. a false positive event occurs when the system reports a response for an utterance even though no valid response for that particular utterance has been defined in the system. Similarly, a false negative is an event occurs when the system fails to generate a relevant response for an utterance for which a valid response exists in the system. Sentence similarity is a new dimension that provides a semantic measure for computing relevance among given words. This semantic similarity approach could be extended to compute relevance of utterances with triggers, thus improving the response relevance rate of the system.

The SIMPLE conversational agent system leverages existing sentence similarity approach that uses WordNet as a semantic database and receives an utterance as a text input from the user. The purpose of the system is to match the user utterance to the most relevant trigger and return a standardized response that is mapped with the matched trigger. It is essential that the

generated response be relevant to the utterance because the generated response will lead the conversation in a particular direction and will have a significant impact on future utterances.

3.2 Dataset

SIMPLE is designed to work with a script structured according to the following description: (This format is used in the popular Virtual People Factory authoring system).

Script: A list of Sentences, each of which are designated as a trigger or response.

Trigger: A sentence that generates one of the possible responses from a Response script. Each trigger is linked to exactly one response.

Response: An output (generally a statement) provided by the conversational agent in response to an utterance. Each response may be linked to multiple triggers.

Utterance: An input from the user that is intended to generate a response from the agent.

Figure 3.1 illustrates the many to one relationship between triggers and responses.

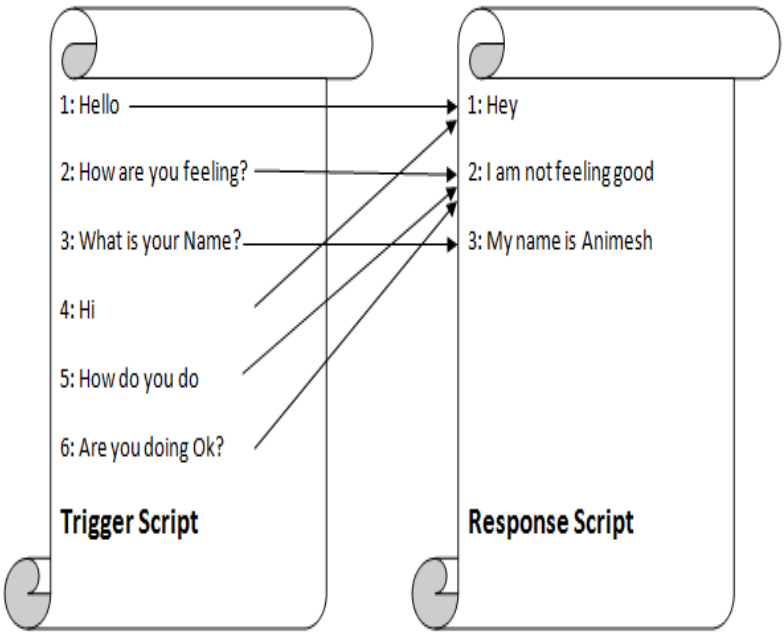

Fig 3.1: Script structure used by SIMPLE

A large corpus of scripts can be downloaded from the Virtual People Factory servers. These scripts have been tediously obtained from domain experts and an iterative scheme by which users test the system and provide feedback that is used to improve the script [Rossen, 2009]. For the purposes of evaluating SIMPLE, we have chosen a script created by medical professionals that simulates the interaction between a doctor and patient complaining of stomach pain, which ultimately turns out to be Dyspepsia. However, any of the available scripts are applicable to SIMPLE.

As a result of the many to one relationship between triggers and responses and the iterative design of the scripts, each response is essentially linked with a cluster of related sentences, as illustrated in Figure 3.2. A *trigger cluster* is composed of at least one or more Triggers that are mapped to a particular response.

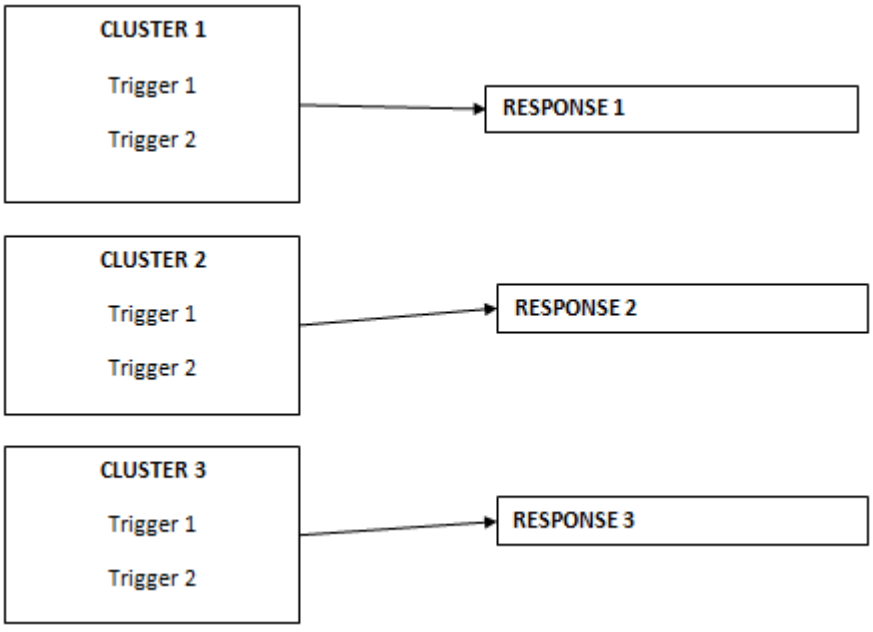

Fig 3.2: Mapping between Trigger cluster and Response

For example, consider the trigger *cluster* of Figure 3.3, which is mapped with response R, such that if an utterance U is matched with any trigger from the cluster it will generate R as a response – provided that it is not matched with any better cluster.

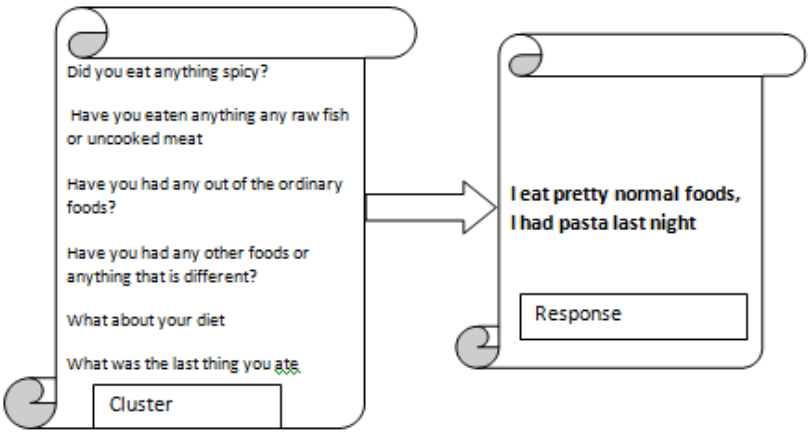

Fig 3.3: Mapping a single cluster to a Response "R"

Figure 3.4 depicts the system domain for SIMPLE. The goal of SIMPLE is to improve script-oriented conversational agents by incorporating semantic relatedness into script-related processes such as retrieving a relevant response for a particular utterance, or crafting scripts efficiently. A conversational agent system relies on a corpus of user utterances to generate standardized responses. To ensure that the system is able to converse naturally with a user, a large corpus (script) with many possible user utterances that are matched to system responses needs to be developed. Naïve users add utterances in the form of simple English sentences to develop such a corpus. The size of the corpus continues to grow as users add new utterances in the system; this may lead to a redundant system. Users produce utterances that are often syntactic variations of sentences already existing in the corpus. The semantic information of such utterances remains consistent across the syntactic variants. A user utterance should be incorporated in a corpus only if it provides new semantic information that is not contained in

existing triggers. A corpus having large number of triggers with similar semantic content that are subtle syntactic variants of a given utterance reduce overall system response time.

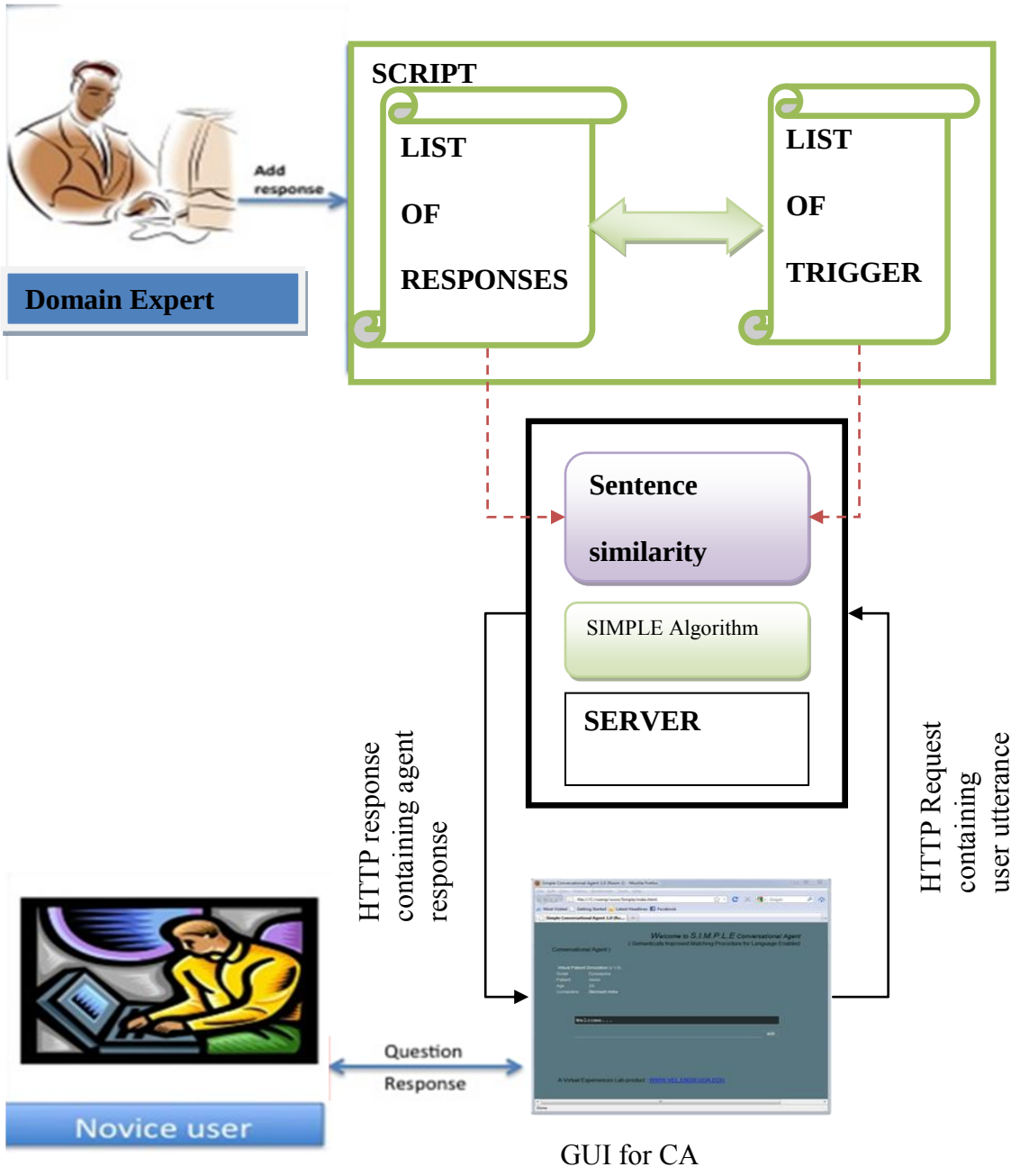

Fig 3.4: Overview of Conversational agent System. Medical professional entering standardized responses in the system (Top Left). A medical student having a typed interaction with the system (bottom)

3.3 Fusion of Semantic and Syntactical matching for measuring Sentence Similarity

A cluster within a corpus is composed of triggers that are mapped to the same standardized response. These clusters enforce a semantic relationship among triggers that are not necessarily syntactically similar. For example, a standardized response like “My parents are not alive” could be mapped with a cluster having triggers “Is your mother alive?” and “Is your father alive?” Both triggers are syntactically different with respect to the words “mother” and “father”, yet these words share a semantic relevance. Sentence similarity captures such semantic relevance among sentences to determine a suitable response for a given utterance.

We have established that a trigger can be expressed using several distinct syntactic variations, while the semantic meaning of the trigger remains unchanged. We exploit this fact to capture the semantic content of a sentence (trigger) and use this information during the process of matching. For example, the words „Light“ and „Brightness“ are syntactically distinct but share the same semantic meaning in certain senses of the words. Therefore, we make use of semantic content that is inherent to both utterances and triggers to generate the most relevant response.

While syntactic relevance plays a key role in determining similarity of two sentences, we also include semantic information about utterances and triggers, the motivation being that often utterances and triggers both contain words that are syntactically similar but differ only in few words. For e.g. the sentences “I went to the store” and “I went to the pool” are almost syntactically similar except for the nouns “store” and “pool” which make them semantically quiet different. For such cases, if we include semantics then we may be able to identify words that differ and determine if they are significant enough to prevent a match.

The combined approach establishes the semantic meaning of each word by performing a dictionary lookup for determining the part of speech and a specific meaning of the word.

SIMPLE performs a preprocessing step for gathering semantic information about each trigger by doing a dictionary lookup using WordNet.

3.4 WordNet

To determine the semantic meaning of words, we use a dictionary called WordNet [Princeton]. WordNet is a lexical database which establishes connections between nouns, verbs, adjectives and adverbs. A specific meaning of a word is represented by an object within WordNet called „Synset". A Synset contains information about the definition of the word, its synonyms and explanation about the uses of the word. A word can have more than one part of speech. For example the word light can be used both as a noun and as an adjective. Each part of speech is called a „Sense" in WordNet. Each „sense" is contained in a different Synset. A Synset also contains a "gloss" that defines the concept it represents. For example, the words "night", "night-time", and "dark" constitute a single synset that has the following gloss: "The time after sunset and before sunrise while it is dark outside". Synsets are connected to each other via explicit relations. WordNet defines relations between synsets and relations between word senses. A relation between synsets is a semantic relation, and a relation between word senses is a lexical relation. The difference is that lexical relations are relations between members of two different synsets, but semantic relations are relations between two whole synsets. WordNet defines relations between synsets and relations between word senses. Synsets are organized as "Is-a-kind-of" and "Is-a-part-of" hierarchies. For example, "car is a kind of a vehicle" and "trunk is a part of a tree". Semantic similarity uses both IS-A relationships. Figure 3.5 shows a synset hierarchy in WordNet. The root describes the most general concept. The leaf nodes represent synsets containing specific meaning of the word.

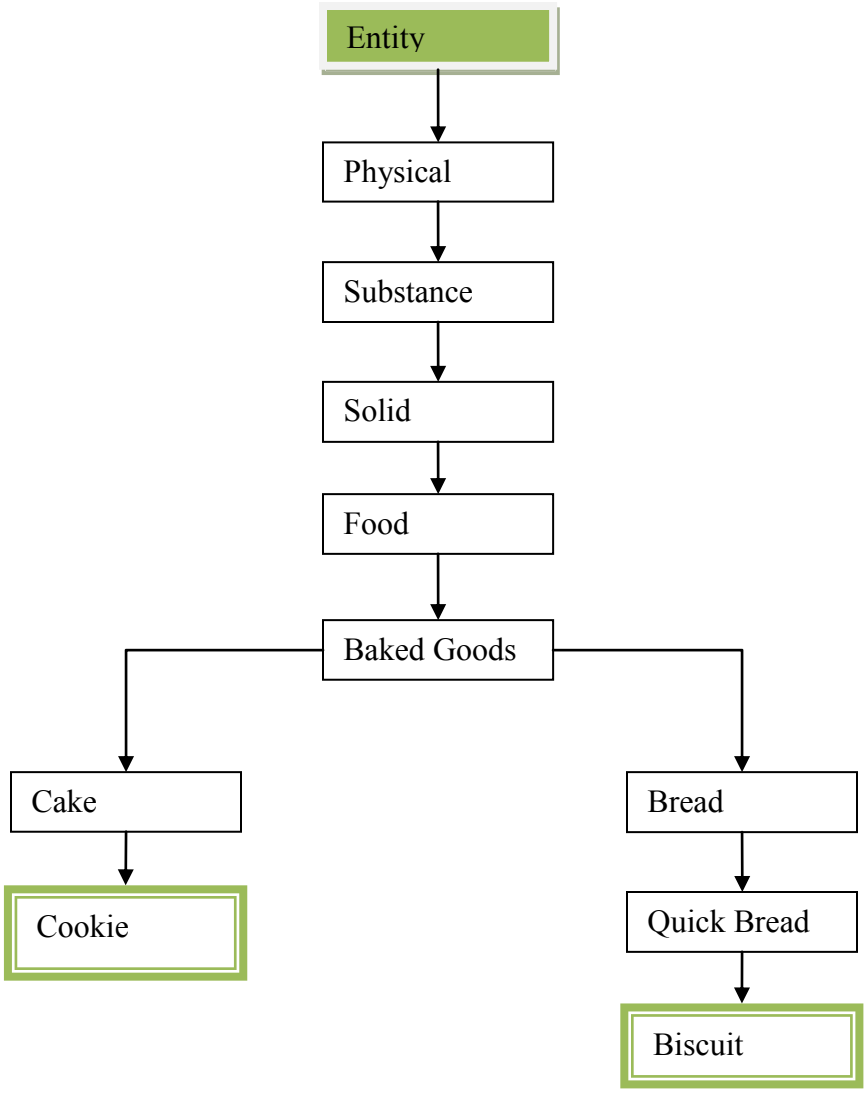

Fig 3.5: Synset hierarchy in WordNet representing IS-A relationship

3.5 Measuring semantic similarity using path length distance

Given two sentences, we have created a measurement that determines how similar the meaning of two sentences is. A high score indicates more similarity between the meanings of the two sentences. The steps for computing semantic similarity between two sentences follow. Note, Parts 1 through 6 of the semantic similarity measurement are derived from [Crowe and Simpson 2011], including source code.

1. Tokenization

The semantic content of a sentence is determined by the meaning of each word in the sentence. The meaning of each word is established by performing a lookup operation using WordNet. For this purpose each sentence is partitioned into a list of tokens, for example the sentence, “How are you feeling today?” will have tokens as “How”, “are”, “you”, “feeling”, “today”.

2. Determine Part of speech

WordNet is organized into different taxonomies where each sense (part of speech) of a word can be found in a different taxonomy. Hence we need to determine the most appropriate part of speech for each word by performing tagging. This step performs part of speech tagging by assigning part of speech to each token. Every token is assigned one of the available parts of speech *i.e.* Noun, Adjective, Verb and Adverb. This algorithm uses Eric Brill‟s tagger [Brill, 1992] which is a rule based tagger. Brill‟s algorithm takes a token as input and outputs the most appropriate part of speech for the particular token by using predefined rules for classification. Brill‟s tagger is error driven because it uses supervised learning techniques for improving accuracy. It performs transformations by assigning and changing tags to each word based on predefined rules. After changing incorrect tags by applying rules over and over again, Brill‟s tagger is able to achieve high accuracy.

3. Perform stemming of words

Sentences are often composed of words which contain common morphological and inflexional endings. Such words are needed to be converted to their base forms for ensuring successful dictionary lookup operations. The stemming process involves removing common morphological and inflexional endings of words. For example Boxes -> Box + s -> Box. We use the Porter stemming algorithm [Porter, 1980] for this purpose. The algorithm is similar to a lexicon finite state transducer [Mohri, 2008], which performs following operations: splitting a

word into possible morphemes then getting intermediate form of the word and finally map stems to categories and affix to the meaning of the underlying form.

4. Build Similarity matrix

WordNet returns the most frequently used word sense for a given word as a part of its lookup operation. A semantic similarity relative matrix, R[m, n] of each pair of word senses, is formed where R[i, j] is the semantic similarity between the most appropriate sense of word at position i of sentence 1 and the most appropriate sense of word at position j of sentence 2. Thus, R[i,j] is also the weight of the edge connecting from i to j. The words must belong to the same part of speech *i.e.* either noun or verb, if not then semantic similarity for the pair of words is not determined.

5. Compute a similarity score

Following equation is used for determining semantic relevance score for a given pair of words [Crowe and Simpson 2011].

$$\text{Score} = \frac{2 * \text{least common ancestor depth}}{\text{depth of word 1} + \text{depth of word 2}} \quad (1)$$

Where, the least common ancestor depth is the distance of the least common ancestor synset from the root node. From Equation 1, the distance of least common ancestor (Bread) is 7. Also the depths of words “bread” and “biscuit” from the root are 7 and 9 respectively. Hence, Semantic Score= 2*7/ (7+9) =14/16=0.875. Figure 3.6 shows Least Common Ancestor for words “bread” and “biscuit”.

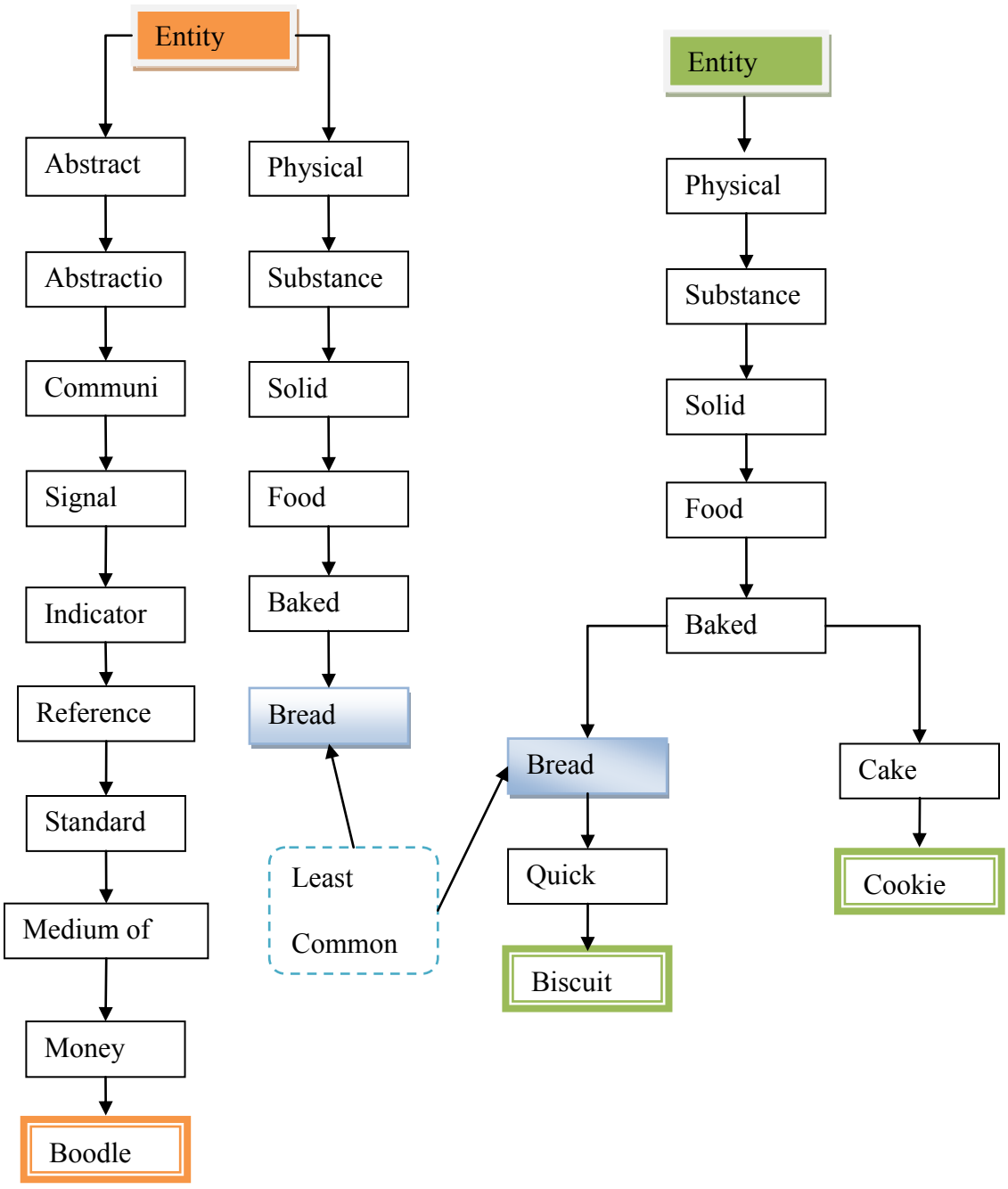

Fig 3.6: Finding Least Common Ancestor (LCA) for words "Biscuit" and "Bread"

3.6 Measuring Syntactic similarity using Edit distance measure

Levenshtein distance [Levenshtein, 1966], also called the edit distance, and measures the minimum number of operations (insert/delete/substitute) that are needed to transform string 1 into string 2. For example the edit distance for strings “Cat” and “Rat” is 1 since only one substitution operation (substitute „C‟ with „R‟) is required to transform “Cat” into “Rat”. A 0 distance indicates that the strings are identical and no operation is required for the transformation. Levenshtein distance is zero if and only if strings are identical. The lower bound for Levenshtein distance is always at least the difference of the sizes of the two strings and at most the length of the longer string.

3.7 SIMPLE Algorithm

SIMPLE algorithm leverages sentence similarity for determining semantic relevance score for each cluster. These clusters are evaluated and compared based on semantic as well as syntactic metrics.

I. For each trigger in the trigger script, perform a dictionary look up operation by accessing WordNet and retrieving semantic information including the meaning of each word, part of speech, total number of senses and distance of synset from root node.

II. User submits an utterance as a text input via user interface as in Figure 3.7.

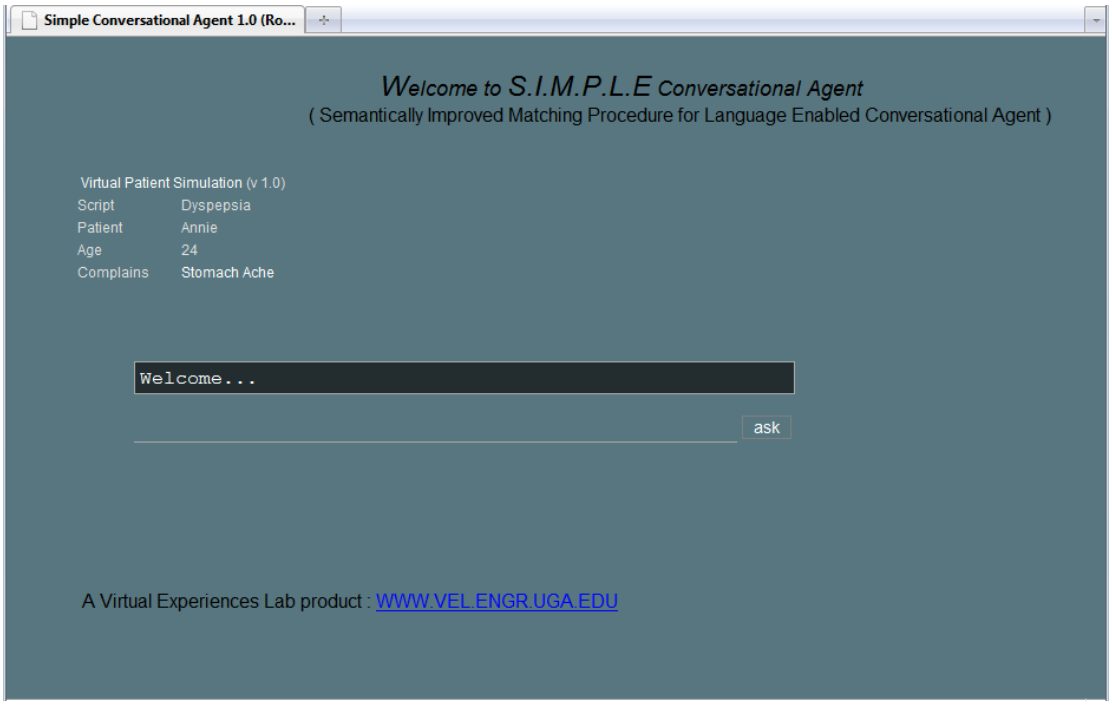

Fig 3.7: User Interface for interacting with SIMPLE conversational agent

III. Perform a dictionary look up by accessing WordNet and gather semantic information for the user utterance.

IV. The user utterance is then matched with each trigger in the script using path length measure (See section 3.5) based on the WordNet taxonomy and semantic relevance scores are computed.

V. Choose the trigger with the global maximum semantic similarity score to the user utterance.

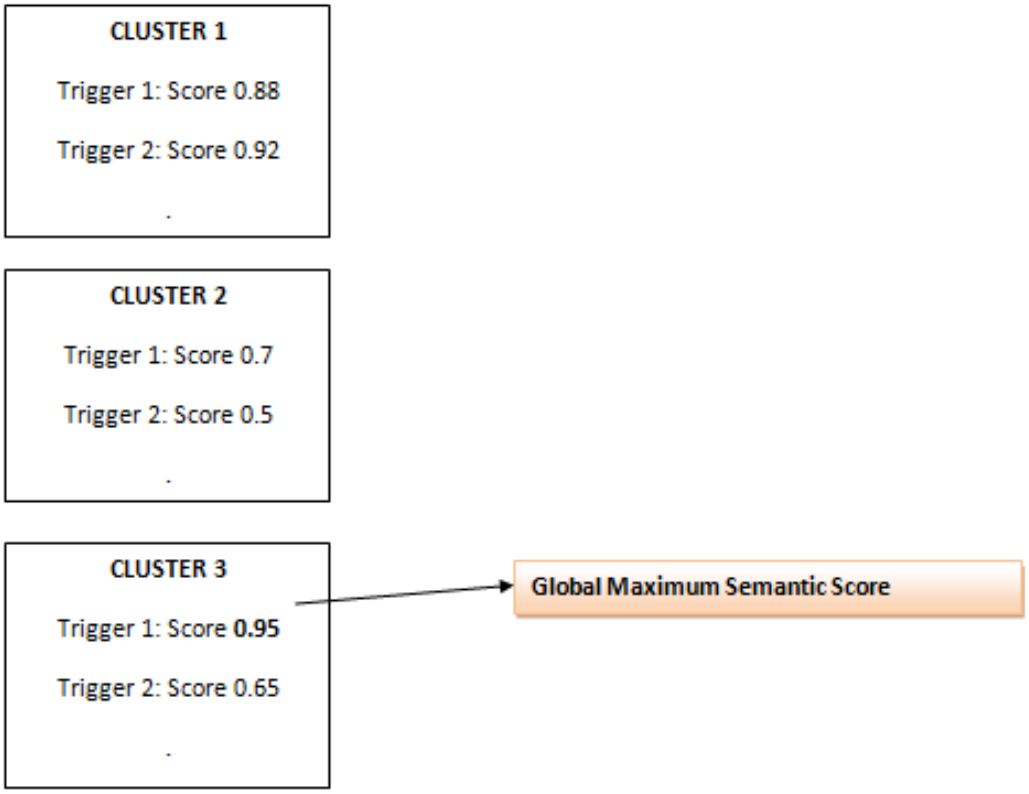

Fig 3.8: Cluster with Max Semantic Relevance Score

This semantic score is the maximum semantic relevance score associated with a trigger within the entire script. If multiple triggers with equal global maximum values are discovered then we choose the trigger that belongs to the cluster having the maximum average cluster score. The cluster that contains a trigger with global maximum semantic score is denoted as Cluster 3 in Figure 3.8.

VI. Compute the Local Cluster Average Score for each cluster in the trigger script.

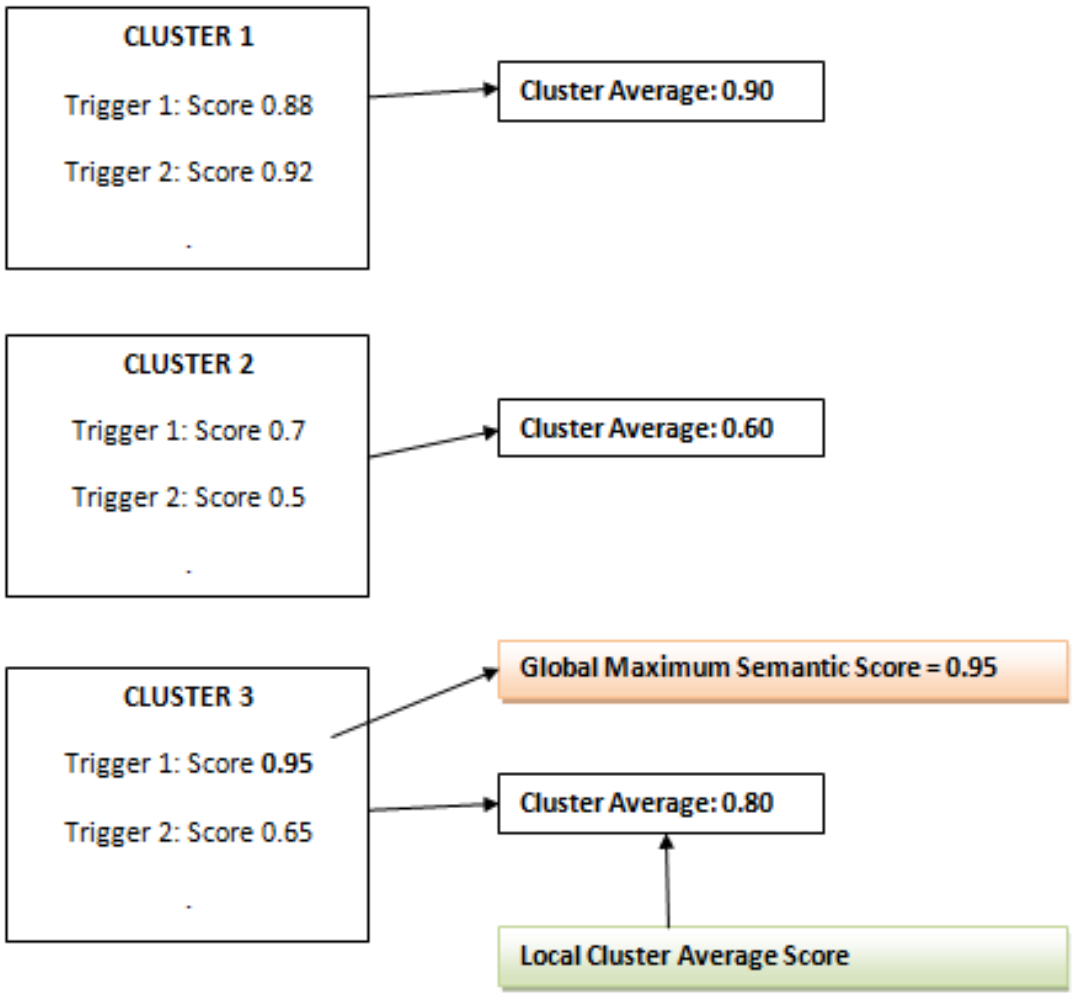

Fig 3.9: Finding Cluster Average score for each Cluster

The Local Cluster Average Score is the average semantic relevance score of the cluster that contains the trigger having the Global Maximum Semantic Score.

The significance of average cluster score is that it represents how similar triggers are within a given cluster with respect to the user utterance. For example, from Fig 3.9 a cluster average score of 0.9 indicates that cluster 1 contains triggers that are more relevant to a given utterance as compared with cluster 2 and cluster 3 which have comparatively low Local Cluster Average Score, and thus are less likely to contain relevant triggers.

VII. **Choose the cluster having maximum Local Cluster Average Score**

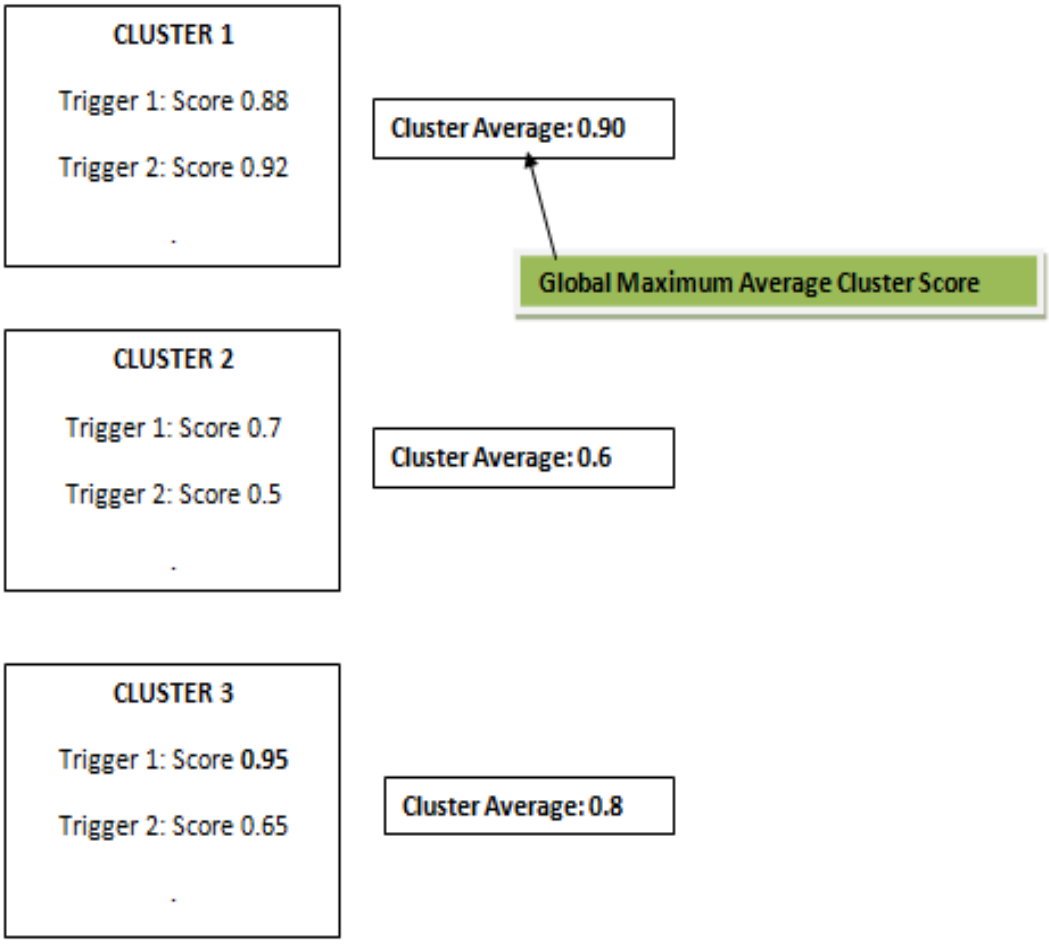

Fig 3.10: Finding Global Maximum Cluster Average

We calculate the average semantic relevance score for each cluster in the trigger script. The Global Maximum Average Cluster Score is the maximum average score of a cluster within the script. In above figure Cluster 1 has the maximum average semantic cluster score of 0.9.

VIII. Find local cluster score for the cluster identified in step V

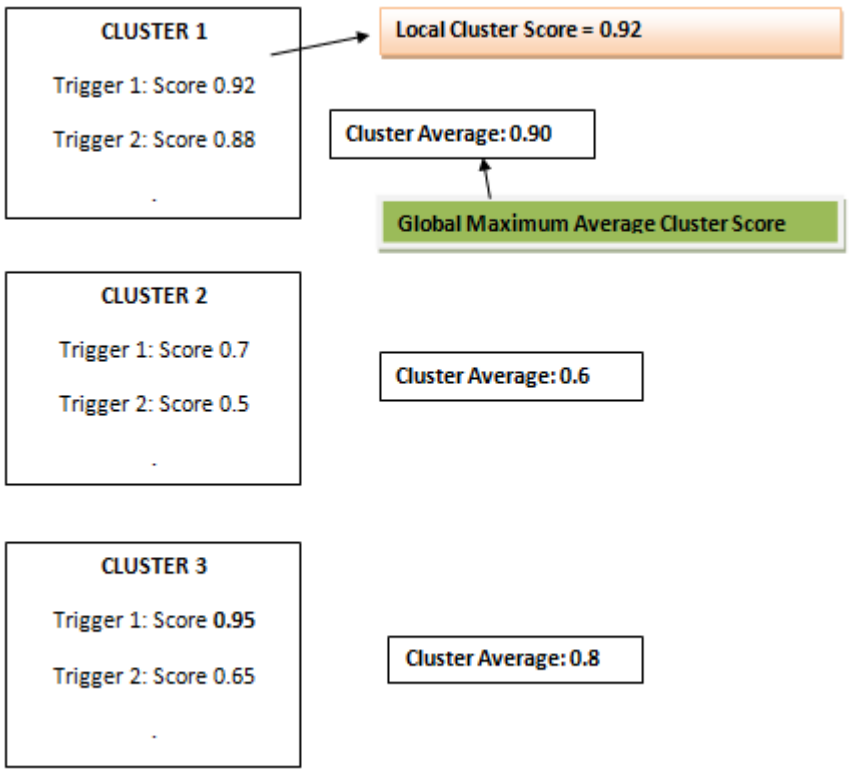

Fig 3.11: Finding the Local Cluster Score

Local cluster score is the maximum semantic relevance score within the cluster which has the global maximum average cluster score.

IX. Calculate Syntactic relevance score for cluster A and cluster B, which have been identified in STEP V and STEP VII respectively.

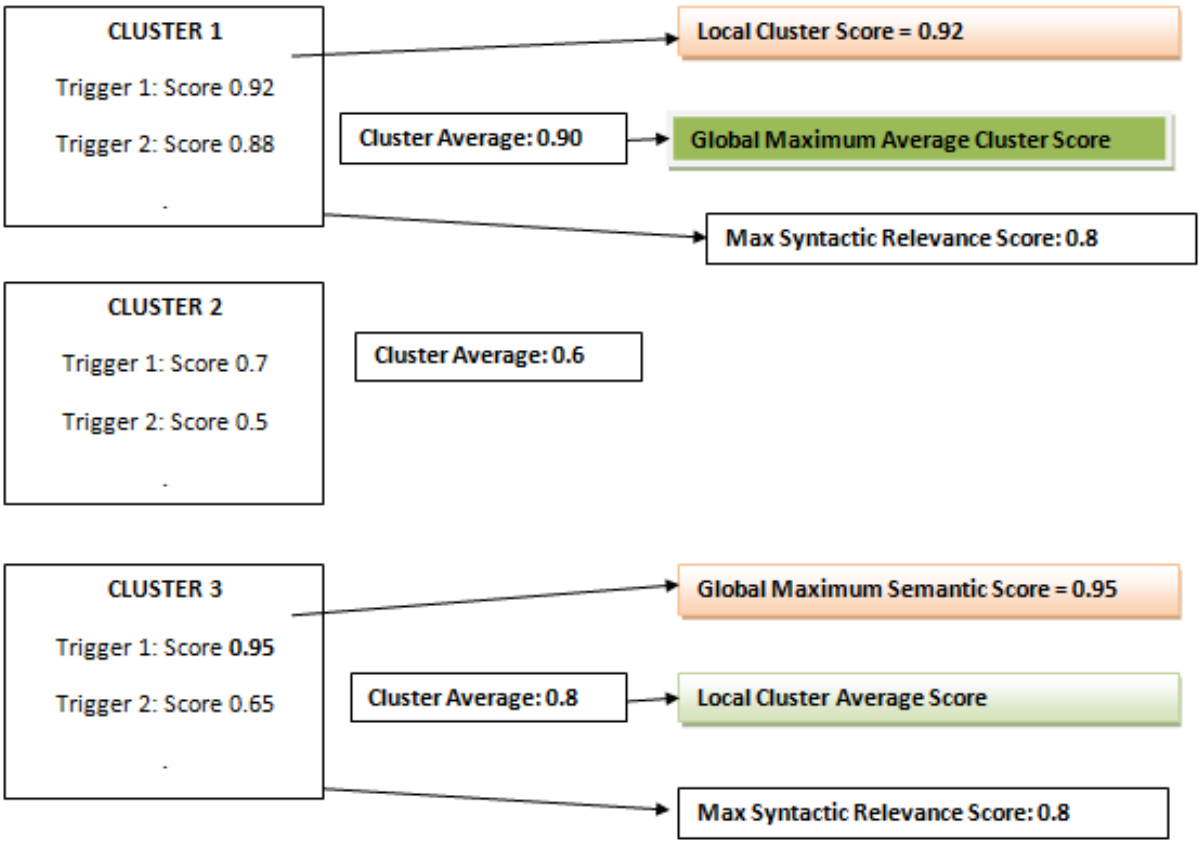

Fig 3.12: Semantic relevance scores for each cluster

The Syntactic Relevance Score is computed by matching the user utterance with each trigger from the script. Levenshtein distance measure is used for computing syntactic relevance. Note: The Levenshtein algorithm is described in details in previous section 3.6.

X. Compute:

Score difference = (global_max_Semantic_score - local_cluster_score)

Average difference = (global_max_avg_Cluster_score - local_cluster_avg_Score)

Score difference = 0.95-0.92 = 0.03

Average difference= 0.9-0.8 = 0.1

CASE 1: If Score difference==0 AND Average difference==0

The difference between semantic relevance scores will be 0 if the dictionary look up operation for the utterance returned no semantic information or every trigger in the script is the same. In such a case we consider syntactic relevance scores of the triggers and return a response as output that is mapped to a trigger with the highest syntactic relevance score. For example, an utterance such as “how are you” returns no semantic information when a dictionary look up in WordNet is performed. As a result, both the maximum semantic relevance score and maximum average cluster scores are returned as 0. If this occurs, the syntactic relevance score is used to select a response.

CASE 2: If Score difference <= Average difference

This case indicates that there are two clusters such that the difference between the cluster average scores is much higher than the difference between individual trigger relevance scores. A large difference in cluster average score suggests that the cluster with max cluster average score is composed of triggers that are semantically more relevant to the utterance than the triggers contained in the other cluster.

We select the response associated with the cluster having the maximum cluster average score only if the syntactic relevance score for this cluster is greater than the syntactic relevance score of the other cluster (Cluster with max semantic relevance score). Otherwise we select the response associated with the cluster that has the trigger with maximum semantic relevance score. The syntactic relevance score is checked

because the goal is to match a trigger that is both syntactically and semantically relevant to a given utterance.

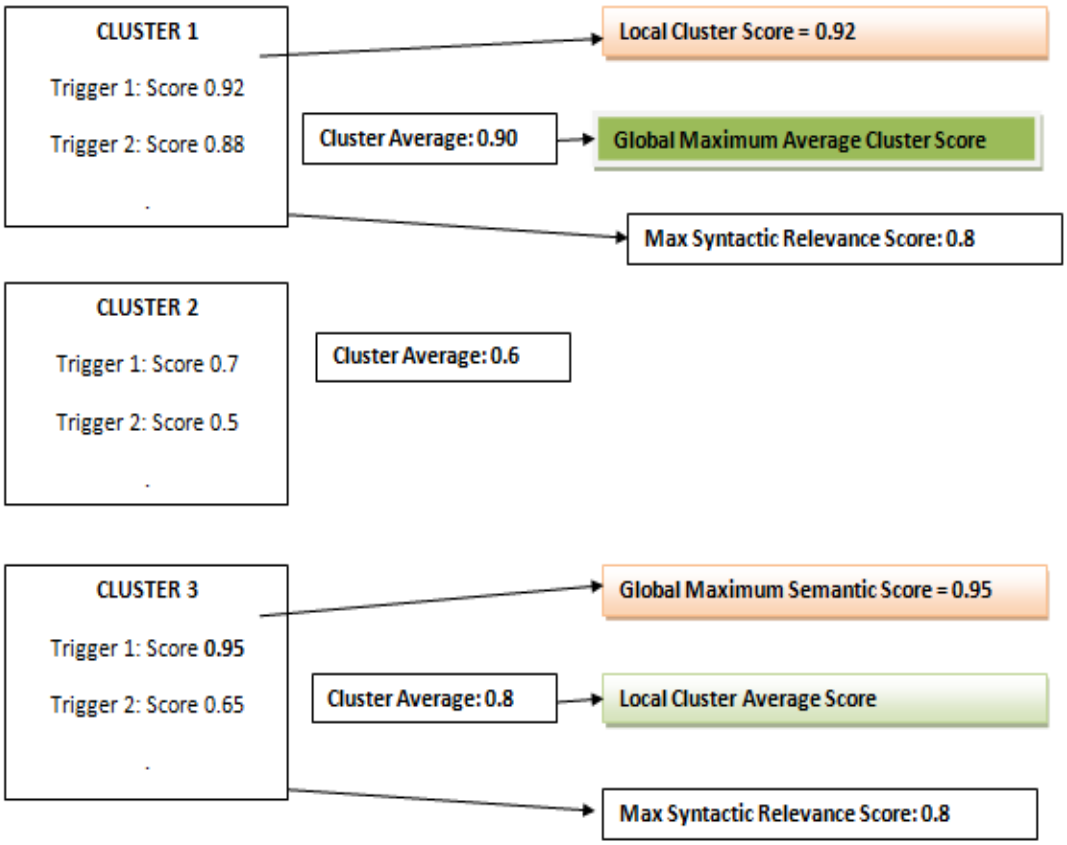

Fig 3.13: Cluster having the maximum average semantic relevance of all clusters

CASE 3: if Score difference > Average difference

This case indicates that there are two clusters such that the average semantic scores are very close to each other. In other words, both clusters contain a collection of triggers that are semantically related to the given utterance. However, we know that one of these

clusters has a trigger that has the maximum semantic relevance score. Hence, we select the response associated with this cluster.

Fig 3.14: Cluster containing trigger with maximum semantic relevance

CHAPTER 4

EVALUATION

4.1 Overview

We now describe the evaluation of different approaches that were used for matching user utterances with triggers to produce a standardized response. Each approach was evaluated on the basis of accuracy of the generated response. The accuracy of a response is determined by a human subject who classified the generated response as either “Relevant” or “Irrelevant” based on its relevance with the given utterance. Standardized scripts [Table 4.1] were used from Virtual People Factory [Rossen, 2009] an online portal which serves as an information gathering tool for improving conversational agents. These scripts included responses created by domain experts and triggers that were submitted by users during iterative testing of the scripts. Users have access to Virtual People Factory web portal where they either have a typed interaction with a conversational agent or add triggers to existing scripts. The transcripts from these interactions are logged by Virtual People Factory. User utterances from these transcripts were extracted and served as input data for our tests.

Table 4.1: A Partial Transcript from Virtual People Factory

Utterance	Response
Hello	hello doctor
Goodbye	Bye
to do you have any diarrhea	I‟ve been constipated for the last week
Good and comeback	Yeah
how much aspirin do you take	i usually take 2 pills, 2 or 3 times a day.
you also take Zestril for high blood pressure	I take zestril once a day, in the morning
right and it's good that you came in at with a	Yeah
Do you work at the post office	i work at the post office

4.2 Experimental design

4.2.1 Experiment 1

We evaluated following two approaches that use semantic similarity matching algorithm for producing a relevant response.

- Semantic only approach: Approach that uses semantic matching only
- Syntax and Semantic approach: Combined approach that uses syntax as well as semantic matching

Step 1: Utterance: A total of 2635 sentences from Dyspepsia script were used as utterances such that, at any given instant, a sentence could be either a trigger or an utterance but not both, *i.e.* a “leave-one-out” approach to validation. We removed a trigger one by one from a cluster and used it as an input utterance. When a trigger is removed from a cluster, the cluster average changes and we perform the matching operation using new cluster average scores. Since

the trigger is removed from the cluster before using it as an utterance, it functions as an unknown or a completely new utterance that the system has not matched before; this ensures that the test environment is able to consistently provide unique input for each test run.

Step 2: Algorithm: The semantic matching algorithm receives an utterance as input and performs matching based on the specified approach (Semantic only or combined approach *i.e.* Syntax and Semantic) and generates a response as output.

Step 3: Validation: In the validation phase, the cluster id of the generated response is compared with the cluster id of utterance. If both the utterance and response have the same cluster id then the response is categorized as accurate otherwise it is reported as an error.

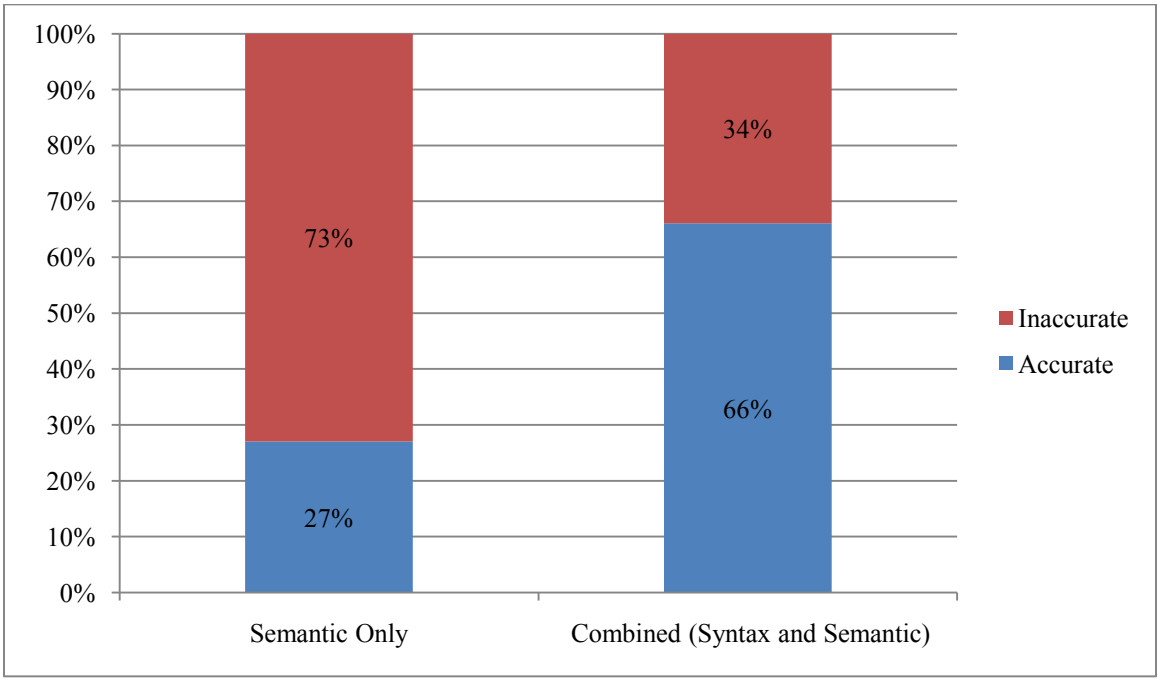

Fig 4.1: Accuracy results for approach A and approach B

According to experimental results sown in Figure 4.2, approach A that employed semantic matching only reported an accuracy of **26.99 %** while Approach B that used a combination of semantic and syntactic matching reported an accuracy of **66.09 %.** A significant gain in accuracy confirms that the proposed approach is capable of generating more relevant

responses for new utterances that are not part of the existing trigger script. We also observed that if an utterance was present in a script as a trigger then both the approaches report a perfect match and produce the expected response.

4.2.2 Experiment 2: Comparing SIMPLE and VPF

The second experiment was designed to compare the proposed approach (combination of syntax and semantic matching) with traditional approach (syntax matching only) *based on real transcript data*. Experiment 1 was designed to evaluate cluster based approach using an automated testing procedure. The automated testing process required eliminating a trigger from the script which was then used as an input for the system. The VPF algorithm is tightly coupled with both trigger and response scripts as a result eliminating a trigger from the script and using it as an input for each test run was not feasible .Virtual People Factory uses a traditional approach *i.e.* syntax based matching procedure for matching user utterances with standardized responses. A transcript from VPF containing 2725 utterance-response pairs was generated. Each utterance-response pair was classified into one of the categories in Table 4.2, based on the accuracy of the generated response. Note: Timed triggers were mapped to responses that were generated only if no user input was received after an elapsed period of time. Also, the category "N/A" represents the triggers that did not had any relevance with the current response script. These were mostly triggers from different scripts and related to different topics.

Table 4.2: Table lists the number of Responses in each category as classified by VPF users.

Category	Accuracy Category	Occurrences	Category %
1	No Response	1	0.036 %
2	Misleading	1	0.036 %
3	Inaccurate	480	17.61 %

4	Accurate	1744	64 %
5	Requires Context	4	0.14 %
6	Problematic Input	348	12.77 %
7	Timed Trigger	70	02.56%
8	N/A	77	02.82 %
	TOTAL	2725	

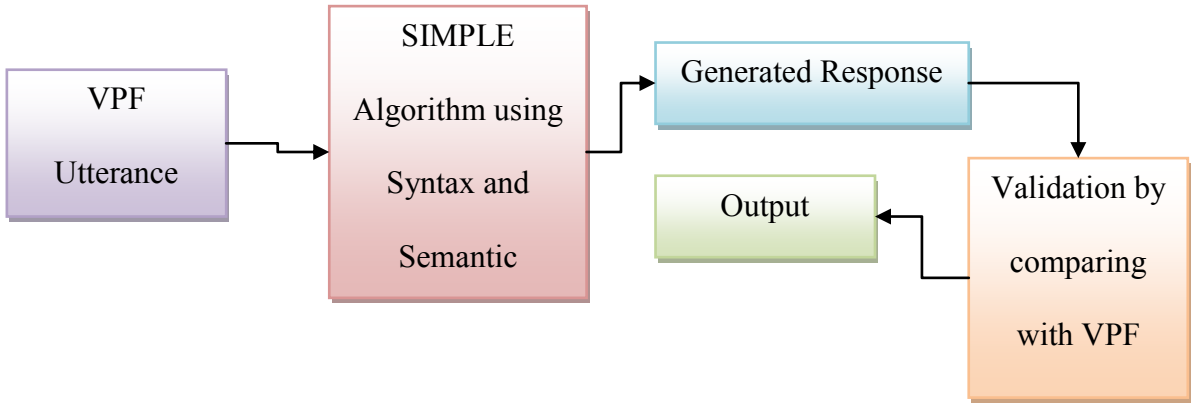

Fig 4.2: Experiment design for comparing SIMPLE and VPF

Step 1: Utterance: A total of 2725 sentences from the dyspepsia transcript were used as utterances.

Step 2: Algorithm: The SIMPLE algorithm (Syntax + Semantic) was used to process each utterance and produce a response as output.

Step 3: Validation: Each response was manually classified into one of the categories in Table 4.3 by the experimenter. Validation of generated responses is necessary for comparing response relevance of VPF and SIMPLE.

The following table shows results obtained by SIMPLE

Table 4.3: Response categories for SIMPLE

Category	Accuracy Category	Occurrences	Category %
1	No Response	0	0 %
2	Misleading	0	0 %
3	Inaccurate	494	18.12 %
4	Accurate	1529	56 %
5	Requires Context	2	0.07 %
6	Problematic Input	553	20.29 %
7	Timed Trigger	0	0 %
8	N/A	147	5.39 %
	TOTAL	2725	

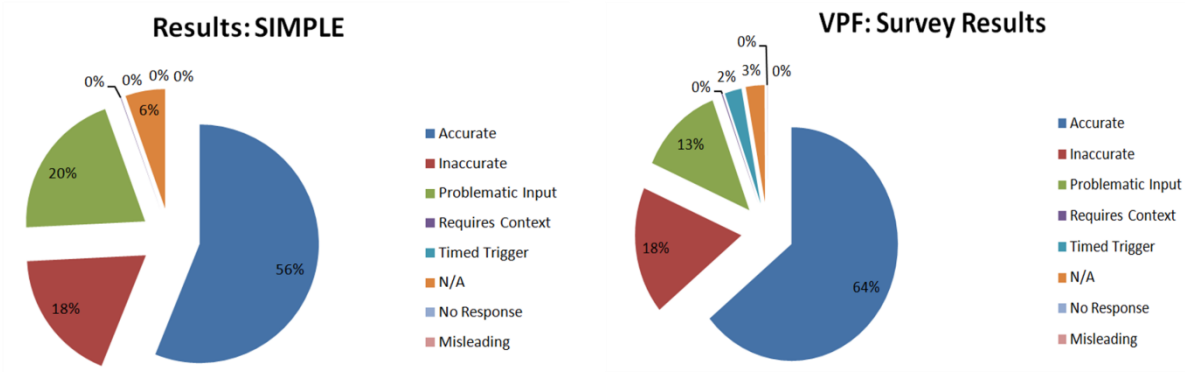

Fig 4.3: Comparison of SIMPLE Vs VPF

CHAPTER 5

DISCUSSION & CONCLUSION

5.1 Discussion

Experiment 1 compared both semantic only approach and combined *i.e.* syntax and semantic approach in terms of accuracy of the generated responses. The semantic only approach generated 711 relevant responses at an accuracy rate of 26.99 % while the combined approach generated 1741 relevant responses with an accuracy rate of 66.09 %. The semantic only approach reported low accuracy because user utterances that contained little or no semantic content could not be matched to any of the existing triggers. For example, an utterance, "How are you", does not carry semantic information as a result a match was not found leading to an error. In such cases the combined approach was able to leverage the syntactic relevance among the utterance and triggers and selected a match based on the maximum syntactic relevance score only.

Experiment 2 compared VPF that is based on syntactic only approach with SIMPLE that used a combination of syntax and semantic measures. SIMPLE conversational agent reported an inaccuracy rate of 18.12 % which was very close to the inaccuracy rate of VPF conversational agent that had an inaccuracy rate of 17.61%. The semantic information for user utterances having typographical errors cannot be established by WordNet, hence SIMPLE was unable to match such utterances with triggers. VPF was found to have a higher accuracy rate (64%) than SIMPLE (58%). This is reflective of the relative maturity of the VPF syntactical approach, which uses a multitude of syntactical distance metrics that have been optimized over the course of many years. SIMPLE, by comparison, is still in the early stages, and requires significant user

testing to optimize. In addition, by incorporating the syntax matching approach of VPF, SIMPLE should be able to be at least as accurate as VPF, with the new semantic information offering performance improvements.

5.2 Conclusion

It is widely accepted that the effectiveness of user interactions with a conversational agent largely depends on the relevance of responses that the system is able to produce given a user utterance. With the use of semantic distance metric SIMPLE is able to capture both the semantic as well as syntactic relatedness between utterance and responses, thus improving the relevance of responses generated by a Virtual Conversational agent. SIMPLE is still in infancy stage and yet its performance and accuracy are far more promising than syntax based conversational agents which leaves room for improvements. SIMPLE can be effectively used in the field of training and education for teaching domain specific procedures using standardized scripts.

5.3 Future Work

The addition of semantic information to conversational agent problems provides a number of exciting possibilities beyond those presented in SIMPLE. Using semantic similarity measures for maintaining the context of current conversation or automatically changing the subject or guiding the conversation in a particular direction by a conversational agent is a promising domain to be explored. We discuss few such future improvements for conversational agent systems that could be achieved using syntax as well as semantic metrics.

Avoiding Redundancy: A virtual conversational agent system relies on a corpus of user utterances to generate standardized responses. To ensure that the system is able to carry a natural conversation with a user, a large corpus with all possible user utterances needs to be developed.

Naïve users add utterances in the form of simple English sentences to develop such a corpus. The size of the corpus continues to grow as users add new utterances in the system; this may lead to a redundant system. There seems to be a simple solution to avoid this problem, as novice users submit a trigger to the corpus, for each new trigger we calculate the Script Effectiveness Index which is the average semantic relevance score of the script calculated by matching each trigger with all triggers within the script. If the Script Effectiveness Index of the trigger script increases, then it indicates that a similar trigger already exists in the system.

Session Oriented User Performance: During an interactive session the program logs the triggers or questions asked by the user. After each new entry in the log, a program measures the similarity score of the new trigger with all of the existing triggers in the log. At the end of the session, the average similarity score of the entire log is computed. This average score could be used as a metric to evaluate users‟ performance during the session. For example, a low average similarity score would indicate that the user asked a series of questions that were semantically distinct and hence the user was able to extract more information from the system with each utterance.

Avoiding False Positives/ False Negatives: During an interactive session the user may encounter events in which the system generates either a false positive or a false negative response. Such events are not desirable because they reduce the reliability and user satisfaction of the system. During such an event the system will allow the user to view the top „x‟ number of matches with high scores and select the most appropriate response for that specific utterance. This action will be saved for future events and the utterance will be flagged, at the end of the session all the utterances that have been flagged will be stored in a separate file. The flagged utterances will then be matched with existing triggers in the script for that scenario and a list of

relevant triggers will be produced, this list will contain specifically those triggers which are most likely to cause false positive/negative responses in future interactions. The system can then learn from such past events and adapt its matching to avoid responses that have been flagged for certain utterances.

Creating Level based Scenarios: One of the interesting parts of a dialogue training system is that if the conversational agent is less responsive, or is more particular in terms of providing information in a scenario, the scenario is more difficult. In some situations, this could be desirable. Semantic similarity provides a mechanism to achieve this difficulty adjustment automatically.

User interaction with a conversational agent in a particular scenario can be evaluated using existing test scripts. However, once a user completes the scenario, there is not much he can do to improve his skills on that particular scenario because every time the user interacts with the virtual patient the user will be aware of the question that triggers a specific response. We would like to create such an experience for the users, in which users have an option of selecting varying difficulty levels; these difficulty levels would reflect real life behaviour of humans. One approach would be to categorize scripts based on their Script Effectiveness Index. A script with low Script Effectiveness Index will be placed at a lower difficulty level because the responses would be straightforward and discrete, similarly a Script with a high Script Effectiveness Index would indicate that the responses are closely related to each other and the user must generate specific triggers to get the desired response.

Mapping User Utterances to Responses: Generally, user utterances are matched to input templates, *i.e.* designers try to make educated guesses about what users might say to a

system. Semantic similarity offers a unique alternative or complementary approach. It is possible to directly match user input to responses in some circumstances.

It is possible to map utterances directly to responses based on the semantic similarity score. If this approach is adopted, a large overhead of creating trigger scripts could be avoided, thus saving long script development time and the need for a large number of test users. This approach would be more suitable for large scalable systems where longer script development periods cannot be tolerated. However, this approach would be prone to generate false positives and negative responses, if the response script were not developed efficiently. The response script must contain distinct responses to avoid generating false positives and negative responses. To achieve this, the Script Effectiveness Index could be used as a measure. Also the process for reducing false positives/negatives which has been discussed earlier should be followed, which would lead to development of an efficient response script by the experts.

REFERENCES

1. Brent Rossen, Scott Lind, and Benjamin Lok: *Human-Centered Distributed Conversational Modelling: Efficient Modelling of Robust Virtual Human Conversations*, Intelligent Virtual Agents, 2009.
2. Dickerson, R.: *Evaluating a Script-Based Approach for Simulating Patient-Doctor Interaction*, International Conference on Human-Computer Interface Advances for Modeling and Simulation, pp. 79–84, 2005.
3. George A. Miller: *WordNet: A Lexical Database for English*, http://www.wordnet.princeton.edu, 1995.
4. Kenny, P: *Building interactive virtual humans for training environments*, ITSEC 2007.
5. Brill, Eric: *A Simple Rule-Based Part of Speech Tagger*. In Proceedings of the Third ACL Applied NLP, Trento, Italy, 1992.
6. Banerjee, S., Pedersen, T.: *An adapted Lesk algorithm for word sense disambiguation using Word-Net*, CICLing Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, 2002.
7. *WordNet* by Princeton http://www.wordnet.princeton.edu .
8. Bates, M. Subject Access in Online Catalogue: *A Design Model, J. American Society for Information Science* 11: page 357-376, 1986.
9. Meadow, C.T., Boyce, B.R. and Kraft, D.H.: *Text Information Retrieval Systems*. 2nd. Ed. Academic Press, 2000.
10. Massaro, D.W., Cohen, M.M., Beskow, J., Daniel, S., and Cole, R.A.: *Developing and*

Evaluating Conversational agents, Santa Cruz: University of California, 1998.

11. Weizenbaum, J.: *ELIZA* – "*A Computer Program for the Study of Natural Language Communication between Man and Machine*", Communications of the Association for Computing Machinery, Vol. 9, pp. 36-45, 1996.
12. Colby, K.: *Artificial Paranoia: A Computer Simulation of Paranoid Process,* New York: Pergamon Press, 1975.
13. Wallace, R.S.*: ALICE: Artificial Intelligence Foundation Inc. [Online]*. Available: http://www.alicebot.org, 2008.
14. Michie, D., Sammut, C.: *InfochatTM Scripter's Manual. Manchester: Convagent Ltd*, 2001.
15. Roda, C., Angehrn, A. and Nabeth, T.: *Conversational agents for Advanced Learning:* Applications and Research. Fontainebleau, France: INSEAD – Centre for Advanced Learning Technologies, 2001.
16. Cole, R.: *New Tools for Interactive Speech and Language Training: Using Animated Conversational agents in the Classroom of Profoundly deaf Children*. Boulder: University of Colorado, 1999.
17. Sanders, G.A and Scholtz, J.: "*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.
18. Turing, A.: "*Computing machinery and intelligence*" Mind, vol 54, page: 236, 1950.
19. O'Shea K., Bandar Z., and Crockett K., *A Novel Approach for Constructing Conversational Agents using Sentence Similarity Measures.* World Congress on

Engineering, International Conference on Data Mining and Knowledge Engineering, London, pp. 321-326, 2008.

20. Kerly, Alice., Richard Ellis, Susan Bull, CALMsystem: *A Conversational Agent for Learner Modeling, Knowledge-Based Systems*, Volume 21, Issue 3, AI 2007.
21. Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley: "*Speech Recognition with Weighted Finite-State Transducers"*, Handbook on Speech Processing and Speech Communication, Part E: Speech recognition, 2008.
22. Levenshtein VI: *"Binary codes capable of correcting deletions, insertions, and reversals*". Soviet Physics Doklady 10: 707–10, 1966.
23. Van Rijsbergen, C.J, Robertson, S.E, and Porter, M.F.: *New models in probabilistic information retrieval*. London: British Library, 1980.
24. Crowe, M., Simpson, T. WordNet-based semantic similarity measurement. *http://www.codeproject.com/KB/string/semanticsimilaritywordnet.aspx?display=Mobile* [Last Accessed 7/25/2011]