Title A statistical approach to a verb vector task classifier

Name Zipeng Jiang

A STATISTICAL APPROACH TO A VERB VECTOR TASK CLASSIFIER

By

ZiPeng Jiang

A thesis submitted to the University of Bedfordshire, in fulfilment of the requirements for the degree of Master of Science by research

November, 2010

ABSTRACT

This thesis proposes an integrated statistics-based verb clustering process for Human-Robot Interactions.

How to enable a service robot to understand its user's intention is a hot topic of research today. Based on its understanding, the robot can coordinate and adjust its behaviours to provide desired assistance and services to the user as a capable partner. Active Robot Learning (ARL) is an approach to the development of the understanding of human intention. The task action bank is part of the ARL which can store task categories. In this approach, a robot actively performs test actions in order to obtain its user's intention from the user's response to the action.

This thesis presents an approach to verbs clustering based on the basic action required of the robot, using a statistical method. A parser is established to process a corpus and analyse the probability of the verb feature vector, for example when the user says "bring me a cup of coffee", this means the same as "give me a cup of coffee". This parser could identify similar verbs between "bring" and "give" with the statistical method. Experimental results show the collocation between semantically related verbs, which can be further utilised to establish a test action bank for Active Robot Learning (ARL).

Keywords: service robot, active robot learning, task classification, corpus, pre-processing, verb feature vector, verb clustering.

DECLARATION

I declare that this thesis is my own unaided work. It is being submitted for the degree of Master of Science by Research at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other university.

Name of candidate: ZIPENG JIANG

Signature: ____________________________

Date: ____________________________

ACKNOWLEDGMENT

I wish to thank to my supervisor: Prof. Yong Yue of the Department of Computer Science and Technology, University of Bedfordshire, for his guidance and supervision of the research work and this dissertation.

Thanks are given to Dr. Peter Norrington, Xiao Guo and Tao Cao for their help with experiments, valuable suggestions and proof reading.

My thanks also go to my family, for their support and encouragement during the year of study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 Motivation

The service robot is a branch of the third generation robots which includes home or personal service robots, entertainment robots, education robots, medical robots, healthcare and rehabilitation robots and rescue robots. Service robots are expected to provide services to their users at home and within workshops. From simple assembling tasks to helping aging people living in their own homes and assisting doctors with precise surgical procedures in hospitals. Service robots have played an important role in the development of “intelligent” robots.

Over the past twenty years, the advancement of service robot technology has been quite amazing. Depending on the particular form of their work they need to autonomously co-work with humans in a sensible and adaptable manner. That means that they must be able to recognise their user’s intentions and preferences. However, how a robot can understand human intention is still an exceptionally difficult challenge. Hence, robot learning plays an important part in knowledge acquisition, motivation establishment and preference identification. There are two approaches to solving this issue, the imitation learning and reinforcement learning. Imitation learning uses social cues such as pointing and gazing to indicate what the user intends to do next (Dillman 2004, Breazeal et al 2005, Calinon and Billard 2006). Imitation learning means humans can teach robot by demonstrating gestures. For example the robot can recognise a gesture while the user is pointing to an object. This gesture serves as social cues of the user interest on the object. Then the robot is able to imitate the gesture when the user intention is the same. However, this approach only allows the robot to learn the user intention passively and the robot cannot pick up the intention when the command is subtly different. Reinforcement learning was proposed by Tapus and Mataric (2007) for medical care robots. An award function is employed in the reinforcement learning. The robot will be rewarded when the optimal intention is reached. The aim of this approach is to develop a robotic system capable of adapting its behaviours according to the user's personality, preference and profile in order to provide an engaging and motivating customised protocol.

A new approach is proposed by Li et al (2008) named Active Robot Learning (ARL). This method does not rely on social cues and explicitly defined award functions. The robot can choose what to learn by itself. In ARL, test actions are obtained by the analysis of user intention from their responses. Test actions should represent mappings from the user intentions to his responses with respect to the actions. These actions need to be classified and organised in a hierarchical structure. Because the same task can be conveyed by different commands, robots can perform lots of similar tasks; the number of test actions for all tasks can be huge while the test actions for the similar tasks can be the same.

For this reason, robots have been gradually endowed cognitive capabilities (S. Lang et al 2003) such as social learning, intention recognition, and emotion etc. Natural Language Understanding (NLU), one cognitive capability, plays a crucial role in Human-Robot Interactions (HRI). Research in NLU has been carried out to solve problems such as speech segmentation, speech synthesis, text segmentation, automatic text generation, Word Sense Disambiguation (WSD), syntactic ambiguity, and imperfect and irregular input handling. The achievements in Natural Language Processing (NLP) research have been applied to Natural Language Interfaces (NLI) on restricted domains such as robots for elderly and disabled people, vacuum cleaning service robots and mobile service robots (Roy et al 2000, Iba et al 1999, Mandel et al 2005). These studies have contributed to the development of robot abilities to use natural language. However, these studies also limit use to a single verb to represent an action. However, users possibly use different verbs to represent an action so that the robot may not be able to take the appropriate action if a single verb is used by the robot to represent an action. Thus, the robot needs to recognise the group of verbs that represent an action. The group of verbs that represent an action can be viewed as a class of synonyms in natural language. Therefore, an automatic mechanism of clustering verbs is required by robots.

1.2 Aim and objectives

This study aims at the development of a Test Action Bank (TAB) for ARL, including establishing a pre-processing parser based on corpus and verb clustering model which is used to classify user commands into task categories.

The objectives are:

- ❑ To review literature on previous work in this area.
- ❑ To establish a robot corpus for analysing the user command.
- ❑ To develop a parser to process the corpus into a single collocation of verbs and nouns.

- ❑ To identify and define the features of typical tasks; part of the taught tasks will be classified into categories and organised in a hierarchical structure to support task classification.
- ❑ To define feature verbs in order to represent a scenario of tasks.
- ❑ To calculate the similarity of each verb using a statistical approach.
- ❑ To develop an integrated verb cluster using relevant nouns.
- ❑ To give a visualised example for the verb classifier.

1.3 Methodology and assumptions of this study

A literature review is conducted to understand previous work. Although experiments can give dissect result, the experiment takes much more time and it has different performance on different environments. The literature review support the development of the robot, Human Robot Interaction and Hidden Markov Model algorithm of Part-of-Speech (PoS) tagger which is used for assigning a word into an appropriate word category such as noun, verb, adjective etc., according to the context of the word. Inspired by the review an integrated verb cluster is proposed and this study is therefore established on solid ground.

A comparative analysis is used to find strengths and weaknesses in various studies. In this study, other algorithms such as Unigram are compared with the Hidden Markov Model (HMM), and HMM are proved a better solution to PoS tagging.

Multilevel modelling is applied to demonstrate previous work on syntax parsing and word clustering. It is also used to model the verb clustering procedure in this study.

Observational studies are applied to collect statistics in order to generate membership functions of words.

Case studies are applied to verb clustering. Several scenarios have been created in order to enable service robots to precisely understand a specific task in a restricted domain.

A machine-readable model is a representation of data and information that can be read naturally by computer. It is often encoded as marked up text. Moreover its computational model is mathematically based. Therefore quantitative analysis is employed to create a mathematical model in order to make the computer understand such as the experiment on Pointwise Mutual Information (PMI) and Weighted Jaccord (WJ).

This study focuses on the development of TAB and how to choose suitable test actions according to user commands and different tasks by using a statistical approach. It assumes the service robot has the ability to receive and convert user commands into a text format, and hence tasks described by the text commands can be classified and test actions can be chosen accordingly.

The service robot in this study is assumed for servicing at home. Therefore four typical taught tasks are discussed, as these are sufficient for explaining how the TAB works. Other untaught tasks will not be considered due to safety and other social issues.

For the sake of developing HRI, all the commands are for the robot. These commands are supposed to be simple utterances; sentences which have a complex structure or subordinate clauses are not in the scope of this research.

In the command corpus, some verbal phrases are used to present actions as well. However, these verbal phrases do not have fixed format and meaning; they will not be analysed as simple verbs.

1.4 Structure of this thesis

This thesis is divided into six chapters. This chapter, Chapter 1 gives an introduction to the research and thesis. Chapter 2 describes the main research areas in service robot; and HRI and NLP which give knowledge and an outline of the whole architecture. Chapter 3 establishes a parser for pre-processing on the corpus with some assumptions. In Chapter 4, an integrated verb clustering model is proposed. The integrated model consists of three components which include: the Pointwise Mutual Information (PMI) for measuring semantic-relatedness, the Weighted Jaccard similarity metric and a K-medoid algorithm. Chapter 5 gives an experiment to verify the whole cluster with an analysis of the results. Chapter 6 draws conclusions and discusses further work.

CHAPTER 2. LITERATURE REVIEW

Interaction between humans and robots plays an essential role for controlling robots. In order to enable robots to cooperate with humans effectively, robots need to understand user intentions. However, most of the time, the commands which are used to describe human intentions are not specific. This chapter introduces related work on service robots, principles of Human-Robot Interaction, and some approaches which are used for handling the issue of clustering.

2.1 Service robotics

2.1.1 Historical notes of robots

The word “robot” was coined by Czech playwright Karel Capek’s (1890-1938) brother and its meaning is forced labour or serf. A robot is defined as “a mechanism which moves and reacts to its environment” (Blackmore, B.S. et al 2004). The first robot was invented in 1965, and named “Unimate”. It was installed at a General Motors, plant to work with heated die casting machines. It worked reliably and saved money by replacing people.

Over past the twenty years robots have become much more important in many areas. The two commonest areas of application are military robot and civil robot. A military robot is a humanoid robot with a function of the automatic machine. Just like “BigDog” a military robot, it will be able to serve as a robotic pack mule to accompany soldiers in terrain too rough for conventional vehicles. Such robots are used to complete difficult tasks in battles. These tasks have high risks and need a precise solution, such as unmanned reconnaissance aircraft and bomb disposal.

Civil robots are divided into five types: industrial, entertainment, humanoid, agricultural and service robots. An industrial robot is reprogrammable, multi-functional, and has multi-degrees of freedom. It can be controlled automatically and is able to transport materials. In addition, it is a part of manipulation tool to complete various operations. An entertainment robot can be like human beings, animals, fantasy beings, or science fiction characters and so on. It can walk or complete actions, and has language skills. A humanoid robot not only looks like a human being, but also has human-like functionality, or even has the ability of think as an intelligent robot. An agricultural robot is used for agricultural production, a new development on multi-functional and efficient agricultural machinery. A service robot can be used for a wide range of applications, mainly engaging in maintenance, repair, transportation, cleaning, security, rescue, care, etc.

Robotics helps make products of high quality and low cost in the manufacturing industries. Although it may cause loss of unskilled jobs, it creates new jobs for skilled people in software and sensor development. These machines will have to be maintained and people will have to be trained to operate and maintain them. People could lose unskilled monotonous jobs for which could be replaced by robots and be trained for new skilled creative jobs which robots are not competent. Consequently, the overall loss may not be that serious.

2.1.2 Service robots

According to the IFR (International Federation of Robotics), a service robot is a robot which operates semi- or fully autonomously to perform services useful to the wellbeing of humans and equipment, excluding manufacturing operations (http://www.ifr.org/). Service robots can be divided into home or personal service robots, education robots, medical robots, rescue robots and healthcare and rehabilitation robots. They provide people with a life of ease, and complete difficult tasks for humans.

The advancement of service robot technology is quite amazing. A few years ago there were only a few car factories using robots for assembly or process work such as welding and spray painting. Resently, Gabor et al (2009) designed one service robot platform in the vertical direction. This platform, which consists of mechanical, navigational and control subsystems, ensures higher payload capability than aerial robots and is less environment invasive than industrial gantry robots. Mo Haijun and Huang Ping (2009) point out that grasping and manipulation are the key functions of service robots to help people with their household tasks. It can provide service robot knowledge about the object in a household. Kozima et al (2001) proposed a model of social activities aiming at making robots acquire communicative behaviour through interactions with the social environment, especially with human caregivers. He found that people with autism have difficulties in social interactions, verbal communications, and maintaining a diversity of behaviour. In July 2010, European scientists developed an intelligent robot cleaner to collect the rubbish automatically as the user's wishes. This robot was controlled by a triple intelligent controller, with obstacle avoidance; a data processing system to determine the street line; and a manual control centre to prevent accidents.

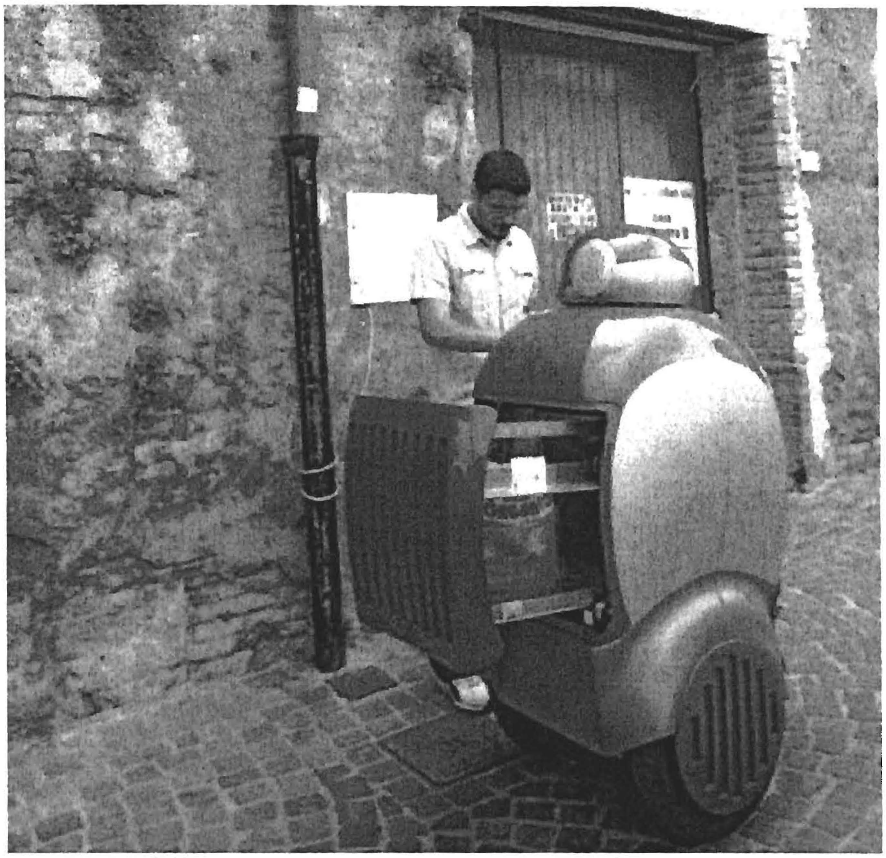


Figure 2.1 Automatic cleaning robot

As shown in Figure 2.1, service robots play an important role in everyday life. They can also cook or fetch meals for the elderly, clean their rooms and toilets, and even handle tasks such as bathing, dressing or supporting walk, sitting down or standing up. This study is based on the cooperation between robots and elderly or children. So the problem has to be addressed of how to make robots understand user intentions when commands are fuzzy or not specific.

2.2 Robot learning

2.2.1 What is robot learning?

Robot learning is a subset of machine learning and robotics. Usually, “robot learning” refers to learning to perform tasks such as obstacle avoidance, control and various other motion-related tasks. Briefly speaking, robot learning is the core of service robots, because service robots which are supposed to assist humans in their daily life must be adaptable and flexible. As a result, robots should know “what to do” and “how to do” from the process of learning.

2.2.2 Classification of robot learning

Robot learning can be divided into learning by imitation and learning by conversation. Imitation uses social cues like pointing and gazing to indicate what the user intended to do next (Dillmann 2004, Breazeal et al 2005). The user first taught a robot by demonstrating gestures, for example, pointing to and gazing at an object, to the robot. These gestures serve as social cues of his interest in the object. Then the robot imitates the gestures for the user’s approval. This imitation process enables the robot to recognise the user intention when it captures the same gestures.

In imitation learning, a Hidden Markov Model (HMM) with full covariance matrix is used to extract the characteristics of different gestures which are used later to recognise gestures from the user. The characteristic of a gesture is expressed by transition across the state of the HMM. Using such a model requires the estimation of a large set of parameters. An Expectation-Maximisation (EM) algorithm is used to estimate the HMM parameters. The estimation starts from initial estimates and converges to a local maximum of a likelihood function. It first performs a rough clustering. Next, EM is carried out to estimate a Gaussian Mixture Model (GMM). Finally, the transitions across the states are encoded in a HMM created with the GMM state distribution.

Another method of learning by conversation is to let robots understand directly the user intention. Hassch et al (2004) developed the Bielefeld Robot Companion (BIRON) which accompanies a human. It consists of cameras, microphones, laser range finder, speech recognition system and other components. This robot is able to understand its user intention through oral instructions and observation of the user's gaze.

The recognition of distant speech with two microphones is achieved by reconstructing a single channel representation of the speech originating from a known location on the basis of the different channels recorded by the microphones (Leese 2002).

The speech understanding components handle spontaneous speech phenomena in conversations between a user and the robot. For instance, large pauses and incomplete utterances can occur in such task oriented and embodied communications. However, missing information in an utterance can often be acquired from the scene. Such as the utterance “take it with you” and pointing at one book implies to the meaning “take this book with you”.

2.2.3 Active robot learning

Active Robot Learning (ARL) was proposed by Li et al (2008). The overall structure of ARL system is shown in Figure 2.2.

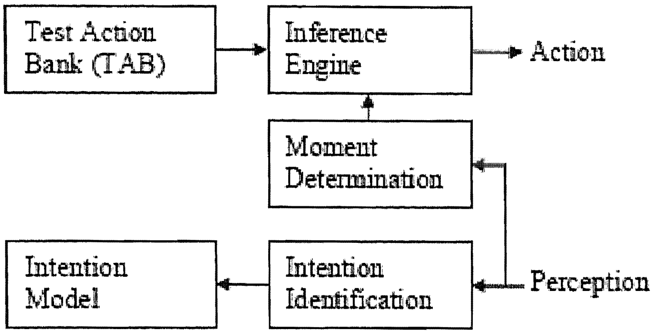


Figure 2.2 Structure of ARL system

The system consists of an action bank which stores actions that can be taken to test its users, an inference engine which reasons about what actions can be taken for a specific purpose, a moment determination mechanism to decide the moment of test, an intention identification mechanism to interpret user responses and to identify intention and preference, and an intention model which represents intentions. ARL differs from active machine learning (AML) because ARL requires a robot to carry out experiments to generate data (evidence), whilst AML only searches for and evaluates available data.

Test actions are those which can be taken to test users. They are associated with conditions and stored in the action bank. Each test action stored in the action bank has a name and content which can tell robot how to do. The conditions express reasons for performing the actions and are represented as propositions. For example, if a robot hands over a glass of water to its user, it would need to check whether the user intends and is ready to take over the glass. One of the test actions for testing the user in this case is to slightly loosen the glass and the condition associated is to confirm the user intention of taking over the glass. The actions and the associated conditions can be designed by robot designers before the robots are deployed.

This study focuses on the Test Action Bank (TAB) which stores test actions responsible for teaching tasks and can be organized in a hierarchical structure to support task classification.

2.3 Human-Robot Interaction (HRI)

2.3.1 What is Human-Robot Interaction?

Human-Robot Interaction (HRI) is the study of interaction between humans and robots. As a user needs to make robots know and understand the intention from his command, the robot must have the ability to communicate with humans to some extent. The interaction oriented robots are designed to communicate with humans and will be able to participate in human society. Especially for social service robots, the robot can be used in hospitals for health care, rehabilitation, and therapy or in family to help feed the elderly or children, and so forth. Fong (2003) pointed out that the core to the success of social service robots would be close and effective interactions between humans and robots. Thus, although it is important to continue enhancing autonomous capabilities, we must not neglect improving the human–robot relationship.

2.3.2 Historical notes on HRI

HRI involves human-computer interaction, artificial intelligence, robotics, natural language processing and social sciences. It is well known that humans are very good at mutual control of their interactions. Calinon and Billard (2006) used an imitation game with motion sensors to teach a humanoid robot to recognise communication gestures. Oliver et al (2005) proposed a method which keeps users in the loop and allows the systematic reduction of uncertainty inherent in implicit cooperation. They gave the architecture of robot control as shown in Figure 2.3.

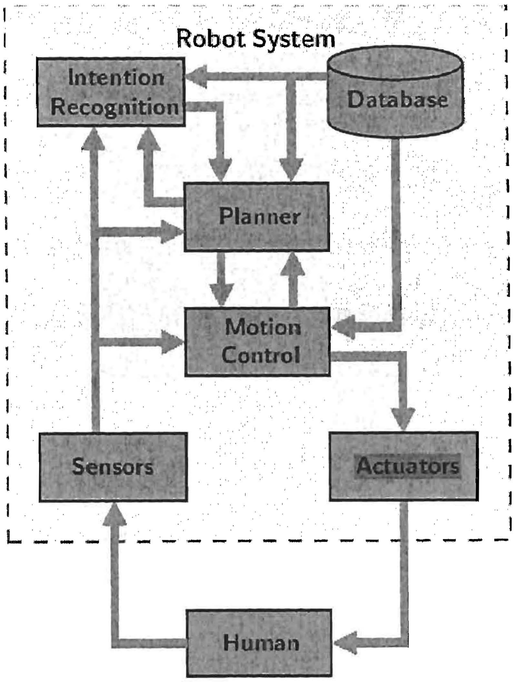


Figure 2.3 Robot control architecture

This model includes an Intention Recognition module, which can figure out the human intention when the user cannot do this perfectly. It allows the recognition and the planning of corresponding robot actions.

2.4 Natural Language Processing

2.4.1 Historical notes on Natural Language Processing (NLP)

Natural Language Processing (NLP) is the study of computer science and linguistics concerned with the interactions between computers and human language. Natural Language Understanding (NLU) is a subtopic of NLP in artificial intelligence and it plays a crucial role in Human-Robot Interactions (HRI). Research in NLU has been carried out to solve problems such as speech segmentation, speech synthesis, text segmentation, automatic text generation, word sense disambiguation (WSD), syntactic ambiguity, and imperfect and irregular input handling. With the emerging of artificial intelligence, researchers have realised that robots have to acquire the ability to understand the command and know how to react to the command when the meaning is not explicit. Therefore, this study contributes to the development of robot abilities to use natural language.

2.4.2 What are the specifics of NLP?

2.4.2.1 Verb feature vector

The verb is the crucial point of a sentence, which describes an action, an event, or a state (Holmes et al 1989). It is an essential part in the communication between humans and robots. Verbs are used to describe actions (She threw the stone), activities (She walked along the river) and states (I have $50). A regular English verb has the following morphological forms:

- The root or base form: walk
- The third singular present tense: walks
- The gerund and present participle: walking
- The past tense form and past/passive participle: walked

Usually, a simple feature vector is composed of numeric or nominal values. The collocation feature is one of the most popular used features. A general collocation refers to a quantifiable position-specific relationship between two lexical items. Collocation features encode information about the lexical inhabitants of specific positions which are located to the left and right of the target word. Typical items in this category include the word, root of the word, and part-of-speech for the word (PoS, e.g. noun, verb, adverb, and adjective) (Jurafsky and Martin 2000). This type of feature is effective at encoding local lexical and grammatical information that can often accurately isolate a given sense. In this study, the words themselves (or their root) serve as features. The value of the feature is the number of times the word occurs in a region surrounding the target word. This region is often defined as a fixed size window with the target word at the centre.

In terms of human experience in the use of language, some verbs have different meaning; however, they can have the same meaning when they collocate with some relevant nouns (Guo 2009). Sun and Korhonen (2010) used a set of supervised classifiers to evaluate English verb features and yielded a reasonable result.

2.4.2.2 Part of Speech tagging

Part of Speech (PoS) tagging is a process of marking up words in a text (corpus) using algorithms which describe discrete terms. A simplified form of PoS tagging is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.

Lafferty et al (1991) developed a formal grammatical system called a link grammar, which is a syntactic parser of English. It can be used to annotate words in one sentence. When a word is connected, this link is associated with one of the connectors of the formula of that word and no other links may satisfy the same connector.

Abney (1997) summarized some methods of tagging including HMM Taggers, which is based on a Hidden Markov Model. He said that the strongest advantage is the accuracy of this parser and it can be trained on an unannotated corpus. The error rates reported in the literature range from about 1% to 5%. The most important thing is this parser could give us a tagged corpus for analysis.

2.4.2.3 Using statistics in lexical analysis

Point Mutual Information (PMI), which is given in Equation 2.1, was first applied to measure the semantic association between two words by Church et al (1989) and x and y belonging to discrete random variables quantifies the discrepancy between the probability of their coincidence. Turney (2001) applied PMI to measure the semantic similarity between two words in order to explore synonyms to a word. However, there are some debates that PMI is inappropriate to exploit the differences between synonyms since date is sparse. Some previous studies also discussed the sparse data problem and its reduction in measuring selectional preference by using PMI.

$$I(x;y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \qquad \text{Equation 2.1}$$

Turney introduced a simple unsupervised learning algorithm for recognising synonyms based on the mutual information. He evaluated the performance of this method using 80 synonym test questions from the Test of English as a Foreign Language (TOEFL) and 50 synonym test questions from a collection of tests for students of English as a Second Language (ESL). The combined information obtains a score of 73.75% on the 80 TOEFL questions (59/80) and 74% on the 50 ESL questions (37/50). By comparison, the average score on the 80 TOEFL questions, for a large sample of applicants to US colleges from non-English speaking countries, was 64.5% (51.6/80). He notes, "...we have been told that the average score is adequate for admission to many universities."

Read (2004) presented a project which attempted to classify the emotions (affect) representing in a sentence in written language. He used Mutual Information based on a small corpus of 759 sentences from the domain of fiction. But these tests showed an accuracy of 32.78% which was below a baseline informed by prior knowledge of the distribution of classifications. He noted that the algorithm could perhaps inform a larger-scale process which includes consideration of other measures related to sentiment and affect.

2.4.2.4 Corpus

A corpus, simply defined, is a large body of text. Increasingly the term “corpus” is used to refer to the machine readable variety. Machine-readable corpora have a number of advantages over other forms of storage and can be adapted in many ways. Firstly, and the most importantly, machine-readable corpora may be searched and manipulated without other formats. Secondly, it can be swiftly and easily enriched with additional information (McEnery and Wilson 1993). McEnery also defined a corpus as a large body of text existing in machine-readable form stated as written texts or recorded speech.

Mitchel et al (1993) constructed a large corpus: The Penn Treebank, a corpus consisting of 4.5 million words of American English. They used Part-of-Speech to annotate the corpus and this corpus has a wide range of Treebank users.

2.4.2.5 Word clustering

Li and Abe (1998) proposed an algorithm based on the Minimum Description Length (MDL) in order to improve the efficiency of the method which was obtained by Brill and Resnik (1994). This algorithm is a variety of Mutual Information (MI) and called “2D-Clustering”. The characteristic of this algorithm is that it can iteratively select a suboptimal MDL model from those hard clustering models which can be acquired from the current model by merging a noun (or verb) class pair. They used this algorithm to make progress on a state-of-the-art disambiguation method; disambiguation accuracy had been increased by 2.8% (from 82.4% to 85.2%).

Bellegarda et al (1996) described a word clustering based on the latent semantic analysis paradigm (Wiemer-Hastings 1999). He created two paradigms which can complement each other with named global value (the weight of one word in total corpus) and local value (the weight of one word in each sample text). The weight means the importance of the word in the corpus. K-means and Bottom-up have been employed in the first rough clustering and final clustering.

Matsuo et al (2006) proposed a clustering approach which uses a web search engine. The experimental result shows the algorithm Chi-square has a better performance than MI when calculating the probability between two verbs. The reason is that MI is not the primary method to measure similarity between two verbs. MI is usually used for calculating the probability between Target words and Context words such as verbs and nouns.

Cao et al (2009) proposed how to define a verb feature vector using relevant nouns and presented the development of a task classifier based on the verb feature vector. Cosine similarity has been employed in his paper as follows:

$$similarity = \cos ine(\theta) = \frac{A \bullet B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 * \sum_{i=1}^{n} y_i^2}} \qquad \text{Equation 2.2}$$

Two vectors of attributes are given, A={x_1, x_2... x_n} and B= {y_1, y_2... y_n}, θ, which is represented by a dot product. The result of cosine (θ) is in the range of [-1, 1]. A value of -1 means that the meanings A and B are exactly opposite, 0 means the meanings are independent, and 1 means the meanings are exactly the same. Values in between indicate intermediate similarities or dissimilarities.

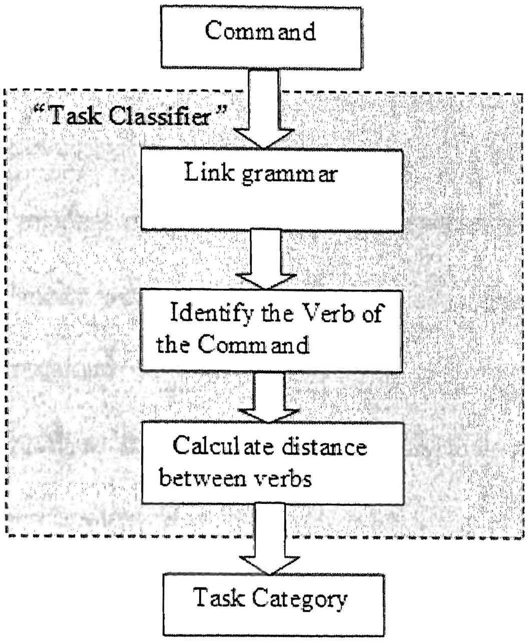

Command
"Task Classifier"
Link grammar
Identify the Verb of the Command
Calculate distance between verbs
Task Category

Figure 2.4 Schematic diagram of the task classifier

Cao also depicted a schematic diagram of the TAB using cosine similarity. When the classifier receives a command, it will identify a useful verb and use its feature vector to represent this command, then compare this vector with the typical task categories' feature vectors and calculate the distance between them. A classification will be made according to the available categories. Using this method, a command with a meaningful verb can be successfully classified. However, the disadvantage of this method is that the test sentences are given by the tester stochastically. Therefore, the result is affected by the practice of using sentences, not by the corpus. Furthermore this method handles large amounts of word clustering with difficulty.

2.5 Summary

This chapter presents an overview of HRI, especially for service robots. Section 2.1 introduced the concept and general situation of service robots. Sections 2.2 and 2.3 stated the history and development respectively in robot learning and Human-Robot Interaction. During the process of interaction, people realise it is necessary to find out a method in order to make service robots understand humans. Natural Language Processing (NLP), introduced in Section 2.4, has great potential to resolve this problem. It is used to analyse the user command by corpus and classifies them into different categories. The conclusion is that service robots can understand humans with these reasonable categories. In this study, tasks are classified through different verbs.

CHAPTER 3. PRE-PROCESSING ON CORPUS

The verb is the crucial point of a sentence, which describes an action, an event, or a state. Therefore the analysis of verbs will help robots to understand human command. The clustering which will be applied in the following chapter is based on a corpus. Pre-processing is a program that processes its input data to produce output which is used as input to another program. In this chapter, the reason for pre-processing on corpus is discussed. An integrated model of a parser is built based on a few assumptions to process the original corpus. The pre-processing on the corpus is phased over several steps. In order to classify verbs, the corpus will be tagged, separated and cleaned. Finally, the corpus will be sorted out as format. The architecture of the pre-processing model will be given in this chapter as well.

3.1 Data acquisition

The experimental result depends on the selection of the dataset, it will be better if the source of the dataset is appropriate. In this part, the condition of the source dataset will be analysed by using psychology. In addition the requirement of the dataset and the implementation of programming will be introduced.

3.1.1 Cognitive ability

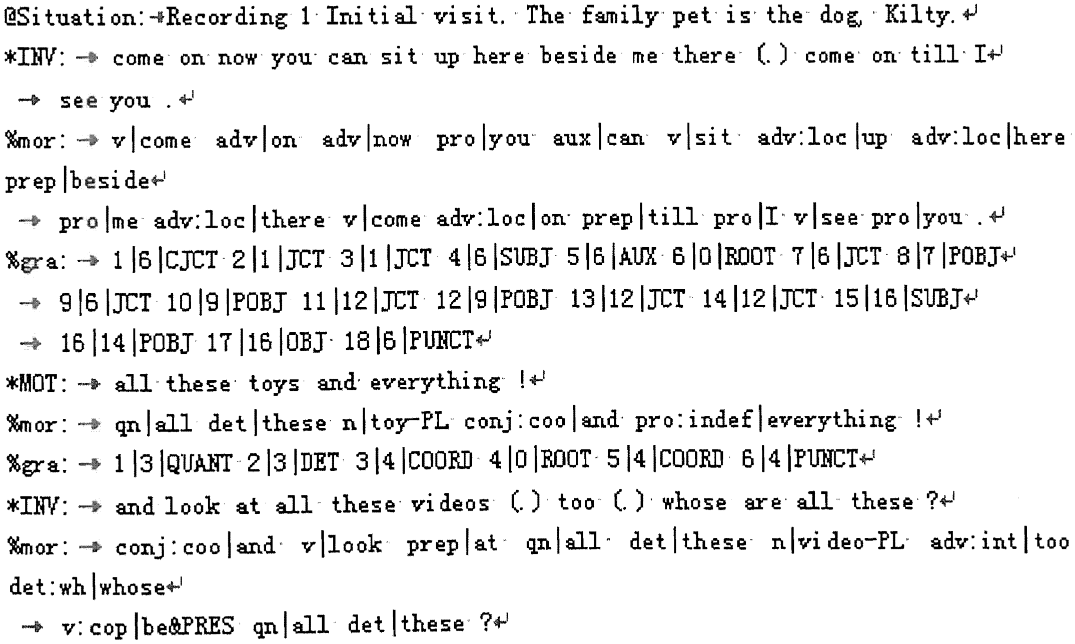
The cognitive ability of robots has been discussed for several years. Cognition means knowing, therefore, cognitive process refers to all those ways in which knowledge of the world is attained, retained and used, including attention, memory, perception, language, thinking, problem solving, reasoning and concept formation. Where robots are concerned, cognitive ability is presented as social learning, intention recognition and memory. Robot learning can be categorized as learning by intimation and learning by conversation. Hence, in order to figure out the meaning of human intentions, the command becomes the object of analysis. Therefore learning by conversation cannot be developed without human commands. On the other hand, command is fundamental to conversation.

As command plays an important role in robot learning, most user intentions are conveyed by command. In one user command the verb is used to represent an action or a type of action. However, users may use different verbs to represent an action so that robots may not be able to take the appropriate action if they use only one verb to represent an action. Thus robots need to recognise the group of verbs that represent an action. The group of verbs that represents an action can be viewed as a class of synonyms in a natural language. As a result, clustering on the verbs is a crucial part of robot learning, especially in intention recognition.

3.1.2 Corpus acquisition

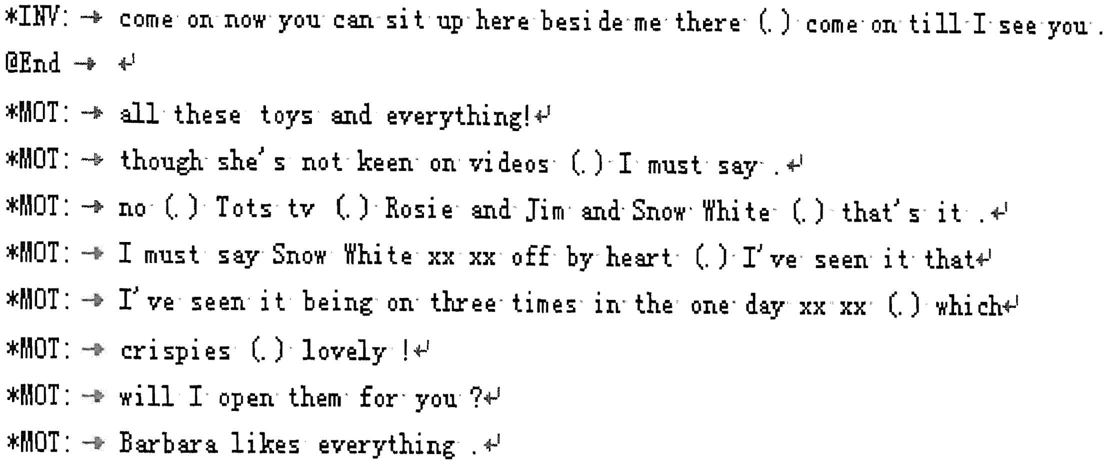
Roughly speaking, this pre-processing parser is mainly applied to processing the corpus. Supposing that the final target is a verb cluster then the corpus selected needs to have plenty of verbs and nouns. Moreover, service robots can hardly make a complex response under the human commands and complex action corresponds to complex verbs. To be brief, during the service robot working it usually uses simple verbs such as “bring”, “take”, “put” et al to talk with users. Therefore the corpus does not need to have verbs with complex meaning such as “abdicate”, “decorate”, “circumvolve” and simple action verbs will be expected in this corpus such as “move”, “take”, “give”. Thus, only one part of the Manchester Text Corpora (MTC) is employed in the following research.

The Manchester Text Corpora is a child language development corpus. It includes Standard English from different parts of the UK, such as Belfast, Birmingham, and Bolton in Lancashire. These small size corpuses contain a lot of daily conversations between children with their parents. From these conversations, plenty of common verbs can be extracted. Moreover, each sentence is just like the user command which can use to interact with robots. All of these corpuses are combined into a large corpus which is the original corpus in my research. However, these corpuses have already been tagged in an unknown way; these tags cannot be used in this study. Above all, these tags have to be deleted. Here are the two samples:



```
@Situation:	Recording 1 Initial visit. The family pet is the dog, Kilty.
*INV:	come on now you can sit up here beside me there (.) come on till I
	see you .
%mor:	v|come adv|on adv|now pro|you aux|can v|sit adv:loc|up adv:loc|here
prep|beside
	pro|me adv:loc|there v|come adv:loc|on prep|till pro|I v|see pro|you .
%gra:	1|6|CJCT 2|1|JCT 3|1|JCT 4|6|SUBJ 5|6|AUX 6|0|ROOT 7|6|JCT 8|7|POBJ
	9|6|JCT 10|9|POBJ 11|12|JCT 12|9|POBJ 13|12|JCT 14|12|JCT 15|16|SUBJ
	16|14|POBJ 17|16|OBJ 18|6|PUNCT
*MOT:	all these toys and everything !
%mor:	qn|all det|these n|toy-PL conj:coo|and pro:indef|everything !
%gra:	1|3|QUANT 2|3|DET 3|4|COORD 4|0|ROOT 5|4|COORD 6|4|PUNCT
*INV:	and look at all these videos (.) too (.) whose are all these ?
%mor:	conj:coo|and v|look prep|at qn|all det|these n|video-PL adv:int|too
det:wh|whose
	v:cop|be&PRES qn|all det|these ?
```

Figure 3.1 Original corpus with tag in MTC



```
*INV: → come on now you can sit up here beside me there (.) come on till I see you .
@End → ↵
*MOT: → all these toys and everything!↵
*MOT: → though she's not keen on videos (.) I must say .↵
*MOT: → no (.) Tots tv (.) Rosie and Jim and Snow White (.) that's it .↵
*MOT: → I must say Snow White xx xx off by heart (.) I've seen it that↵
*MOT: → I've seen it being on three times in the one day xx xx (.) which↵
*MOT: → crispies (.) lovely !↵
*MOT: → will I open them for you ?↵
*MOT: → Barbara likes everything .↵
```

Figure 3.2 Original corpus without tag in MTC

3.2 Implementation

“Perl” is employed to develop the script. Perl is a highly capable, feature-rich programming language with over 22 years of development. It was originally developed by Larry Wall in 1987 as a general-purpose UNIX scripting language to make processing easier. Perl borrows features from other programming languages including C, shell scripting and AWK. This language provides powerful text processing facilities without the arbitrary data length limits of many contemporary UNIX tools and its major features are easy to use. This language parser fits the processing of the simulation for the proposed research (www.perl.org).

Comprehensive Perl Archive Network (CPAN) is an archive of over 20000 modules of software written in Perl. These documents contain dozens of common functions. Normally, large Perl programs often make use of lots of modules and CPAN can save programmers weeks of time. Packages which are employed in the following script come from CPAN.

3.3 Process of pre-processing

3.3.1 Architecture of the pre-processing

As mentioned previously, the function of verbs is to present an action and the other things too. Classifying verbs is a crucial approach to figuring out the intention of users through the command. In this study, therefore, verbs, nouns and collocations between them need to be picked up from the corpus. These elements are used to calculate the similarity of verbs. However, the original text corpus contains much punctuation, words, and the abbreviation of words; these might affect the precision rate in the following experiment. Thus, pre-processing is necessary to be executed.

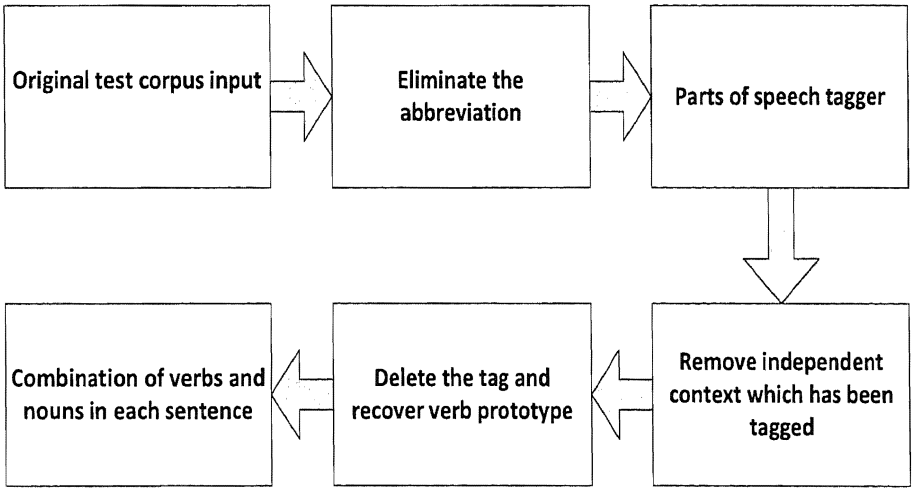

Original test corpus input
Eliminate the abbreviation
Parts of speech tagger
Remove independent context which has been tagged
Delete the tag and recover verb prototype
Combination of verbs and nouns in each sentence

Figure 3.3 Architecture of the pre-processing

Trying to obtain a formal corpus, the pre-processing in the corpus can be divided into six steps shown in the Figure 3.3. Firstly, the input data is the original corpus which contains many utterances in oral English. Secondly, clean abbreviations and contractions in order to tag the corpus, then use Part of Speech to tag the corpus. The next step is to remove the independent context which has been tagged. The final two steps are recovering the corpus and finding out the collocation of verbs and nouns.

3.3.2 Text cleaning

3.3.2.1 Corpus input

A corpus is defined as a large and structured set of texts and it is used to undertake statistical analysis, check occurrences of words or validate linguistic rules. The input data to the pre-processing could be any stochastic text. However, the task classifier is to analyse the correlation of verbs with nouns. The frequency of the common verbs and the size of the corpus should be considered in order to get the result effectively and accurately. An original corpus is waiting for the pre-processing and after it passes through the parser a new formal corpus will be acquired. This formal corpus will have separate statistical elements which can be employed in the verb cluster.

3.3.2.2 Eliminate the abbreviations and contractions

Abbreviation and contraction are common features of written English. This writing style depends on human habits and makes oral English rapid. However, abbreviations and contractions which are convenient in speech can be the obstacle in the process of tagging. Hence abbreviations and contractions need to be eliminated during the process.

An abbreviation is a shortened form of a word or phrase. It consists of a letter or group of letters taken from the word or phrase and it could present the complete meaning of the original word or phrase. For example, the word “Professor” is represented by the abbreviation “Prof”. In addition a contraction is an informal writing type as well. It occurs frequently in speech and writing, in which a syllable is substituted by an apostrophe or other mode of elision, such as, “can not” contracted to “can’t” or “I will” contracted to “I’ll”. Abbreviation and contraction share some similar features in semantics and phonetics. However, they cannot be confused with each other. In this study, abbreviations and contractions need to be restores to the prototype as follows:

Table 3.1 Prototype of abbreviation and contraction

Short form	Category	Result
Prof.	Abbreviation	Professor
Dr.	Abbreviation	Doctor
I’ll	Contractions	I will
You’d	Contractions	You had
I’m	Contractions	I am
Let’s	Contractions	Let us
You’re	Contractions	You are
He’s	Contractions	He is /has

3.3.3 PoS tagging

3.3.3.1 Parts of Speech

Parts of Speech (PoS) are shallow syntactic categories of words, for example: noun, verb, etc. (Robins, 1989). Parts of Speech represent information about how words are used in a sentence, for example which types of word are modifiers, which types of word perform mainly functional roles, and which types of words are the central content bearers in a sentence. There are different categorisations of arts PoS, the most well-known of which is the Penn Treebank (Marcus, Santorini & Marcinkiewicz, 1993), which has 36 main PoS tags. These words have been divided into 9 sets shown in Table 3.2. PoS features can be useful in the analysis of the corpus because they often reflect the characteristics of writing. For instance, they have been used extensively to classify documents by author or genre (Santini, 2007). The motivation for using PoS as classification feature here is that they can indicate the categories of words to some extent. By analysing these kinds of tags on words, each set of words can be classified. In this study the PoS parser has two functions. On one hand it will be deleted when its tag is not noun or verb; on the other hand the number of occurrences of words will be obtained by counting the tags.

Table 3.2 Tags of PoS

PoS Labels	Feature Set
nn,nnp,nns,nnps	# of nouns
vb,vbd,vbg,vbn,vbp,vbz,mb	# of verbs
in	# of prepositions
rb,rbr,rbs	# of adverbs
det	# of determiner
prp,prps	# of pronouns
jj,jjr,jjs	# of adjectives
cc	# of conjunction
pp,pps	# of punctuation

3.3.3.2 PoS tagging by Hidden Markov Model

Hidden Markov Model taggers work well when there is a large tagged training set and could even tag a text from a specialized domain or text in a foreign language to which training corpora do not exist at all.

First of all a Markov model is a stochastic model that assumes the Markov property. It is a recursive process. According to the definition of conditional probability:

$$\mathrm{P}\ (\mathrm{A}|\mathrm{B})\ =\ \mathrm{P}\ (\mathrm{AB})/\ \mathrm{P}\ (\mathrm{B})\ \mathrm{P}\ (\mathrm{A}|\mathrm{B})\ =\ \mathrm{P}\ (\mathrm{AB})/\ \mathrm{P}\ (\mathrm{B}) \qquad \text{Equation 3.1}$$

So:

$$\mathrm{P}\ (\mathrm{AB})\ =\ \mathrm{P}\ (\mathrm{A}|\mathrm{B})\ *\ \mathrm{P}\ (\mathrm{B}) \qquad \text{Equation 3.2}$$

Hence:

$$\mathrm{P}(w_1,w_2,\ldots w_n)$$

$$= \mathrm{P}(w_1,w_2,\ldots w_{n-1})\ *\ \mathrm{P}(w_n\,|\,w_1w_2\ldots w_{n-1})\mathrm{P}(w_1,w_2,\ldots w_n)$$

$$= \mathrm{P}(w_1,w_2,\ldots w_{n-1})\ *\ \mathrm{P}(w_n\,|\,w_1w_2\ldots w_{n-1}) \qquad \text{Equation 3.3}$$

A sequence of events probability of occurrence can be unfolded by the following multiplication:

$$\mathrm{P}(w_1,w_2\ldots w_n)\ =\ \mathrm{P}(w_1)\ \mathrm{P}(w_2\,|\,w_1)\ \mathrm{P}(w_3\,|\,w_1w_2)\ \ldots\ \mathrm{P}(w_n\,|\,w_1w_2\ldots w_{n-1})$$

Equation 3.4

Suppose the number of appearances of any one event w_i is concerned only with the previous probability w_{i-1}. Then the multiplication formula becomes the following formula:

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)\ldots P(w_i \mid w_{i-1})\ldots \text{Equation 3.5}$$

This model is simple, but very useful indeed. For example, when it comes to tagging sentences by using a Markov chain and the constitution of each word in this sentence is just related to the last word, it is a Markov chain. This issue becomes to find w_1, $w_2, \ldots w_n$ which can obtain Max(P(w_1, $w_2, \ldots w_n$)).

Here is the Markov model in Figure 3.4. For example, there are three weather conditions: sunny, cloudy and rainy. We are not sure of the weather condition the next moment, however, we can generate a pattern to draw the weather of tomorrow. We can simply assume that current weather is only concerned with the weather before. This is called the Markov assumption. Although this is a rough estimate and will lose some information, this approach is suitable for analysis.

The current state of the Markov process is only related to the former n state which is called the n order Markov. The simplest model is when n=1, first order mode.

For a state of first order M, there are M*M state transitions. Each state has a certain probability and all the transition probabilities can be presented in one matrix. The assumption is: this transfer matrix is constant.

$$
\begin{array}{cc}
 & \text{Today} \\
\text{Yesterday} & \begin{array}{r}
 \\ \textit{sunny} \\ \textit{cloudy} \\ \textit{rainy}
\end{array}
\begin{array}{c}
\begin{array}{ccc} \textit{sunny} & \textit{cloudy} & \textit{rainy} \end{array} \\
\left[\begin{array}{ccc}
0.500 & 0.375 & 0.125 \\
0.250 & 0.125 & 0.625 \\
0.250 & 0.375 & 0.375
\end{array}\right]
\end{array}
\end{array}
$$

Figure 3.4 Matrix of transfer probability

This matrix reads: If yesterday is fine, today the probability of sunny, cloudy, and rainy is 0.5, 0.375, and 0.125. The sum of the probability of each row and line is 1.

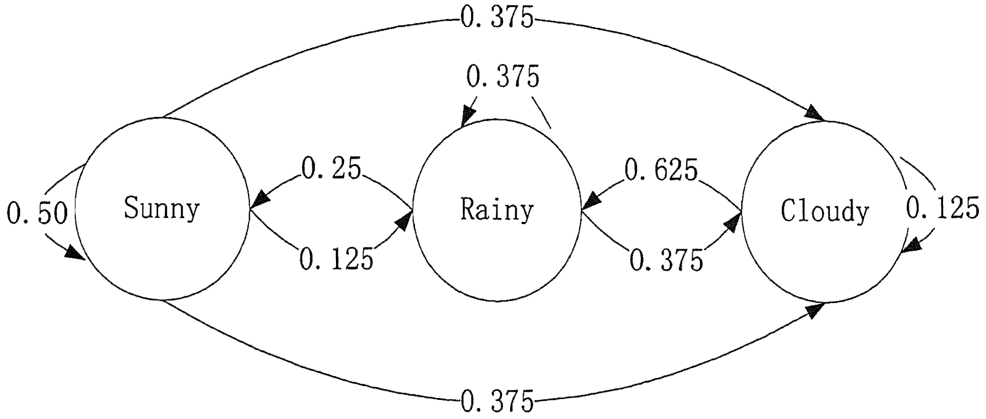


Figure 3.5 Markov Model

A Hidden Markov Chain is more complicated and the basic question is: There are two sequences, one sequence is reason and another is result. The result of the sequence is already known and the reason needs to be figured out. That means in the PoS, the sequence of words is known and the part of speech of each word needs to be calculated.

Using the mathematical formula:

$P(h(t_1),h(t_2),h(t_3),...\,|\,o(t_1),o(t_2),o(t_3)....)$, o represents observed (The result), h represents hidden states (it cannot be observed), t means the times of observing. In the formula above o is already determined, therefore, P(o(t_n)) is a constant. Hence when the maximum probability $P(h(t_1),h(t_2),h(t_3),...\,|\,o(t_1),o(t_2),o(t_3)...)$ is requested then this constant can be ignored.

Two hypotheses can be formed:

- h(t_1),h(t_2),h(t_3), … is a Markov chain, meaning h(i) is only decided by h(i-1) .
- The observation o_i is just concerned with h(i) (also called the independent output hypothesis), hence,

$$P(o(t_1),o(t_2),o(t_3)...\,|\,h(t_1),h(t_2),h(t_3)...)$$

$$= P(o(t_1)\,|\,h(t_1)) * P(o(t_2)\,|\,h(t_2)) * P(o(t_3)\,|\,h(t_3))...$$ Equation 3.6

This problem is becoming much simpler:

$$\text{HHM} = P(h(t_1),h(t_2),h(t_3)...\,|\,o(t_1),o(t_2),o(t_3)...)$$

$$P(h(t_1)) * P(h(t_2)\,|\,h(t_1))... * P(o(tl)\,|\,h(t_1)) * P(o(t_2)\,|\,h(t_2)) * P(o(t_3)\,|\,h(t_3))...$$

Equation 3.7

Then, calculate the Max HMM.

Take a simple example:

Say person cannot directly observe the weather, but just has some algae and knows the probability relations between weather and algae. He can also predict the weather tomorrow. At this time, he has two groups: observation of the algae (state) and implied condition (weather). Therefore, this man is hoping there is an algorithm which can help him predict the weather when there is no direct observation. The HMM can solve these kinds of problems.

Transfer diagram is shown as follows:

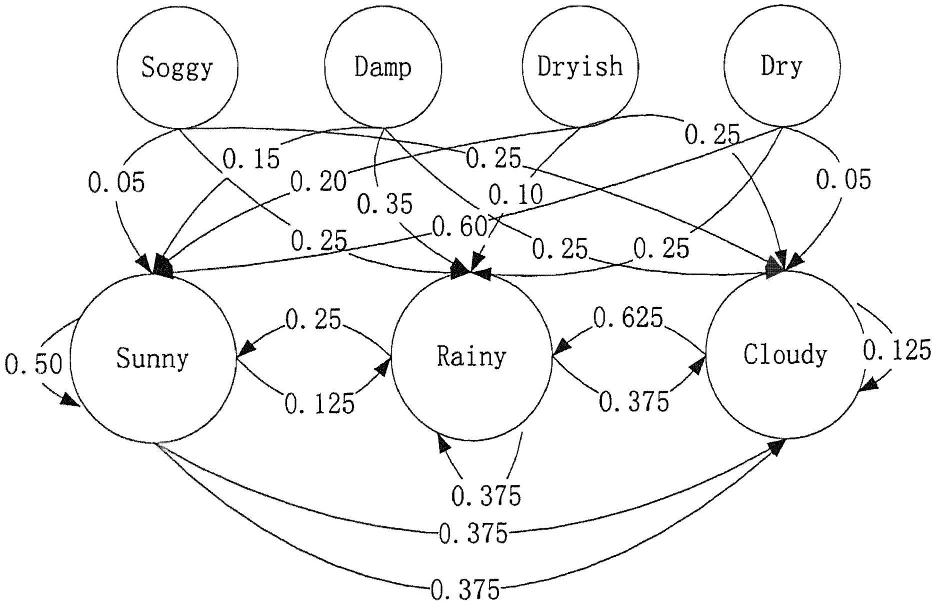


Figure 3.6 Hidden Markov Model

The area between hidden state and observed state presents: In a Markov process, a particular hidden state corresponding to the probability of observation can be presented as a matrix:

$$
\begin{array}{ccc}
 & & \textit{Seaweed} \\
 & & \begin{array}{cccc} \textit{dry} & \textit{dryish} & \textit{damp} & \textit{soggy} \end{array} \\
\textit{Weather} & \begin{array}{r} \textit{sunny} \\ \textit{cloudy} \\ \textit{rainy} \end{array} & \begin{bmatrix} 0.60 & 0.20 & 0.15 & 0.05 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.05 & 0.10 & 0.35 & 0.50 \end{bmatrix}
\end{array}
$$

Figure 3.7 Matrix of hidden state

The question has become how to calculate the Max of HMM = P(h(t_1)),h(t_2),h(t_3)| Dry Dryish Damp). The model of HMM has two kinds of state: observed state and hidden state and three groups of probability: initial probability; transfer probability and emission probability.

Generally speaking HMM can solve these three problems:

- □ Matching the most likely system to a sequence of observations -evaluations, solved using the forward algorithm;
- □ determining the hidden sequence most likely to have generated a sequence of observations decoding, solved using the Viterbi algorithm;
- □ determining the model parameters most likely to have generated a sequence of observations learning, solved using the forward-backward algorithm.

When it comes to a PoS tagger based on HMM, the sentence is already known as the observation state and the PoS tag is the hidden state.

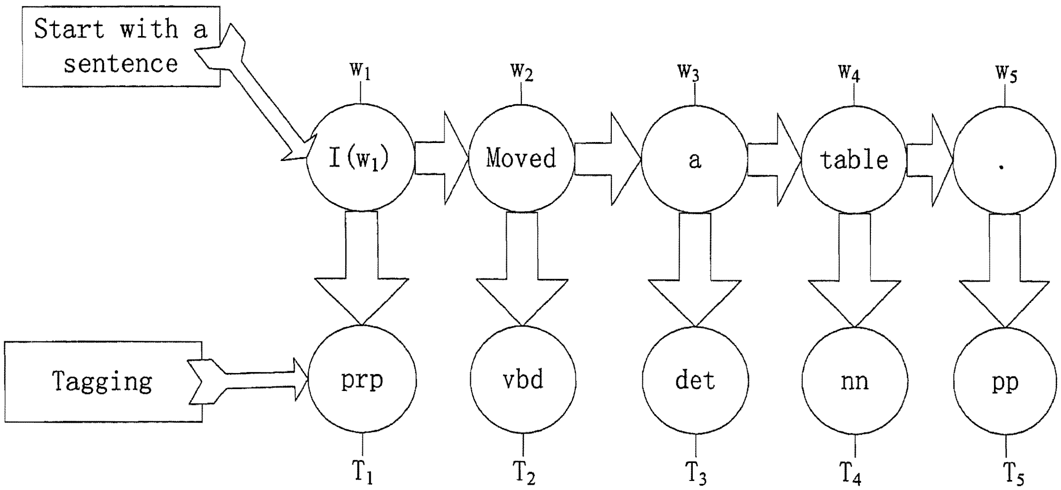


Figure 3.8 HMM on tagging

At the training stage, parameters of the HMM are estimated by Maximum Likelihood Estimation (MLE). MLE for HMM can be implemented by using some classic algorithms such as the Viterbi algorithm and the Forward-Backward algorithm. Matrices of transition and emission probabilities of HMM are therefore obtained. At the testing stage, the HMM tagger first assigns each word in the input sentence a transition probability and an emission probability. The joint probability of transition and emission can then be calculated. A PoS tag is then selected and assigned to a word if the PoS tag is with maximum joint probability of transition and emission given a word and the PoS tag of the preceding word of the current word.

Transition Probability: P ($W_{n+1}|W_n$), represents the probability that the word W_{n+1} occurs following word W_n.

$$\text{Transition Matrix: } \begin{pmatrix} P_{W_1W_1} & P_{W_1W_2} & P_{W_1W_3} \\ P_{W_2W_1} & P_{W_2W_2} & P_{W_2W_3} \\ P_{W_3W_1} & P_{W_3W_2} & P_{W_3W_3} \end{pmatrix}$$

Figure 3.9 Transition matrix

Emission Probability: P ($T_n|W_n$), represents the probability that the PoS tag T_n occurs given the word W_n.

$$\text{Emission Matrix:} \begin{pmatrix} P_{T_1W_1} & P_{T_1W_2} & P_{T_1W_3} \\ P_{T_2W_1} & P_{T_2W_2} & P_{T_2W_3} \\ P_{T_3W_1} & P_{T_3W_2} & P_{T_3W_3} \end{pmatrix}$$

Figure 3.10 Emission matrix

In the Confusion Matrix, each entry is the number of words. The row of the Confusion Matrix represents the class that a word is classified into. The column of the Confusion Matrix represents the class that a word actually belongs to.

$$\text{Confusion Matrix:} \begin{array}{cc} & \begin{matrix} nn & prp & vbd \end{matrix} \\ \begin{matrix} nn \\ prp \\ vbd \end{matrix} & \begin{pmatrix} \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \end{pmatrix} \end{array}$$

Figure 3.11 Confusion matrix

The process of tagging needs to be divided into two steps: supervised learning and unsupervised learning. Hence the original corpus can be separated from two parts: training part and test part. During the supervised learning corpus has been tagged. The target of this process is obtaining the standard of PoS and calculating the confusion matrix. On the other hand the unsupervised learning corpus just consists of word without tag which is called plain text. This parser can't work well without the supervised learning. The following is input and output for training and testing corpus.

Supervised HMM tagger

- Train:
  - input: Tagged corpus
  - output: Trained HMM tagger
- Test:
  - input: Plain corpus
  - output: Tagged corpus

If the corpus has just two sentences:

~	A	lion	ran	to	the	rock
~	det	nn	vbd	in	det	nn
~			nn			vbd
~	The	cat	slept	on	the	mat
~	det	nn	vbd	in	det	nn
~		vbd				vbd

This example shows the probability of “det”, “nn”, “vbd” is greater than “det”, “det”, “vbd” hence Cat should be tagged as “nn”. Moreover the probability of “vbd”, “in”, “det” is greater than “vbd”, “nn”, “det” then “to” and “on” should be tagged as “in”.

Comparing with other algorithms HMM has better performance in both accuracy and speed. The result shows above that the HMM is high accurate with some accepted speed.

Table 3.3 Comparison of algorithms on tagging

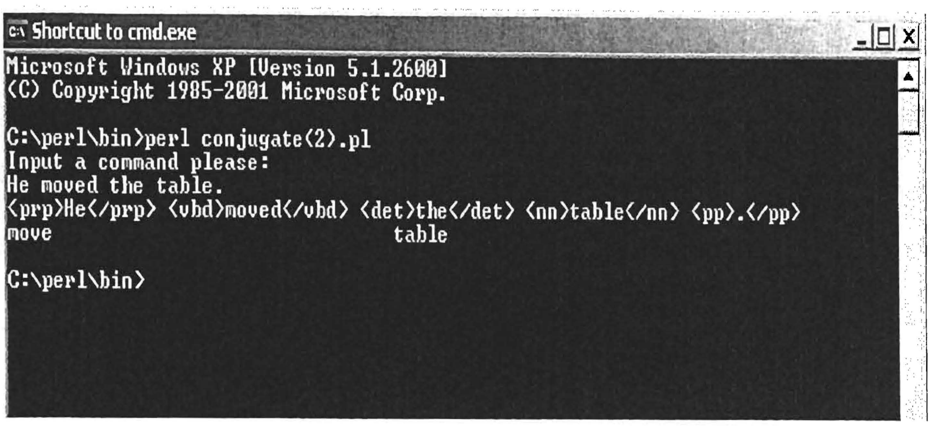
Algorithms	Accuracy	Speed (second per pass)
Unigram	85.4%	0.0003
Unigram with Regexp	88.0%	0.0005
Bigram	89.4%	0.0007
Trigram	88.8%	0.009
Brill	89.9%	0.0029
Hidden Markov Model	89.3%	0.0013

3.3.4 Corpus filtering and recovering

The formal corpus which contains with word and its tagger has been obtained through PoS tagger. In order to collect verb collocations, independent context and tag should be deleted for the end result. For example a formal corpus just has one sentence: "He moved the table." When it passes through the tagger, the result is "<*prp*>He</*prp*> <*vbd*>moved</*vbd*> <*det*>the</*det*> <*nn*>table</*nn*>." Every word in the sentence has two tags beside it. In this study, aiming at keep the verbs and nouns only, "<*prp*>He</*prp*>"," <*det*>the</*det*>" will be no help and need to be eliminated by programming. Moreover, the tag on each word will be deleted by the same method. Hence, this utterance becomes "moved table".

It's easy to observe that the tense of each verb is not the same in every utterance. These tenses need to be recovered to the base form prototype. In the assumption above, utterances are considered as the simple sentence. That means sentences which have complex semantic structure are not in this scope. Therefore the most commonly used three types are "the third singular present tense", "the gerund and present participle", "the past tense form and past/passive participle". In the script a replace approach is employed to deal with this issue. A hash table is created in the Perl script and each verb deformation has a prototype in the sheet. Then, the "moved table" becomes "move table".

The final step is contextual information extraction. Just in case one utterance has more than one verb or noun. At that moment, whatever how many verbs and nouns appear in one utterance, they should be counted as the different collocation. For example "move table and chair" should be considered as collocations "move table" and "move chair".



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\perl\bin>perl conjugate(2).pl
Input a command please:
He moved the table.
<prp>He</prp> <vbd>moved</vbd> <det>the</det> <nn>table</nn> <pp>.</pp>
move                              table

C:\perl\bin>
```

Figure 3.12 Result of pre-processing

So after these processes the result is showed above. This parser is not only used for dealing with corpus but also with the command which comes from the user. Just as expected the formal corpus is consist with many collocations such as "move table". The formal corpus will be showed in the experimental chapter.

3.4 Summary

In this chapter, an integrated approach was proposed to pre-processing the corpus. Section 3.1 gave the source of the data (corpus). Section 3.2 introduced the language which was used to simulation. Section 3.3 described the entire pre-processing including text cleaning, PoS tagging and the filter. The result of the pre-processing will be presented in the chapter 5.

CHAPTER 4. VERB-CLUSTERING

The previous chapter has presented a model to preprocess a corpus. This chapter discusses the verb clustering used for Task Action Bank (TAB). An integrated verb clustering is built in order to choose the appropriate action to response the observed verb in a user instruction. The algorithm for verb-clustering is also introduced in this chapter. Finally, a graph will be employed to display the visible clustering.

Firstly, a service robot fulfils tasks according to user commands. Many commands are declarative sentences containing verbs. Some of these commands have similar senses. Four typical tasks are generalized as the most useful tasks by human user. The results may explain that the statistical approach been used in this study is much more optimal than others. This method based on a corpus will be rational. At least, the results will give a useful guideline for using statistic in verb classification.

Secondly, verbs which play a significant part in these commands have been extracted in the last process. The framework of verb clustering consists of four steps. The first step is to use Pointwise Mutual Information to estimate the semantic relatedness between a verb and its contextual information. The second step is to organise verbs and their context as a bipartite graph. Edges between verbs and their context are weighted by PMI scores. Weighted Jaccard similarity (WJ) is then applied to compute the similarity between two verbs. At the fourth step, K-medoid method is applied to cluster verbs in terms of the results obtained from the first stage.

4.1 Typical task classification

Since the concept of a service robot was proposed, it has provided additional assistances and make a significant difference to people's daily life. Particularly, this sort of robot is competent for tasks such as being a servant at home in order to look after elderly and disabled people.

In this study, service robots need to understand users' intentions when the commands are not so specific. As mentioned above, these implicit commands should be generalized into several tasks. For example, service robots can only receive simple sentences aiming at finishing with commands while looking after disabled people. Suppose a disabled user wants the robot to bring a book for him. Therefore, he can say "bring a book for me". However, it likely to have the same meaning when the command is "pass me the book". All these kinds of orders should be in the same task action bank such as "pass an object". This study is focussed on four classes of daily tasks fulfilled by service robots. These classes of task include "pass an object", "feed the user", "move an object" and "find an object".

4.1.1 Task: pass an object

Transferring of objects between robots and humans is a fundamental way to coordinate activity and cooperatively perform useful work. Suppose an elderly is alone. This user wants to get a book to read. However, this book may be too far from where he is sitting. Fortunately, the service robot is standing beside the user at the moment. Therefore the service robot receives an order from the user: "pass me that book". The service robot will search the task bank in order to find out what the response should be. After successfully finding the requested book, the robot can give this book straight to the user or put in front of him. As the humans' habit the word "pass" can be replaced by "give" or "bring". Therefore service robots should apply the same task "pass an object". "Pass" is defined as a feature verb for this action.

The object in the command can be a book, a cup or other small object. If objects are too large, such as car and house, the commands cannot be accomplished. The inappropriate response or correct suggestion should be given by the service robot.

4.1.2 Task: feed the user

Eating food is absolutely necessary to human life. Some people such as some sick people, elderly people, or post-operative patients cannot eat by themselves. Instead, they need someone to help them to eat.

This task requires robots to act promptly and precisely inferring the user intentions. To accomplish these sorts of task, the service robot needs to recognise when and how much a user will eat, in addition to what the user wants to eat. These similar commands should be generalized in this task such as "can I have a drink?" and "I want to eat a burger." "Eat" is defined as a feature verb for this action.

4.1.3 Task: move an object

Service robots can not only help people without or with reduced self-care ability, but can also be useful when objects are heavy or not easily moved. The targets of these tasks may be beds or tables. Firstly, a service robot needs to find out where the object is. Secondly, it will follow the command to pick it up. A new command to the service robot will then be given by the user in order to specify the next action, while the service robot holding the object. For instance, the object needs to be moved up, down, left, right, forward, backward, stay for a moment, or move to a special location. Then the service robot considers whether the task can be done, and moves the object if it can. Otherwise, an error response will be given by the service robot. Finally, task is accomplished. Also, the uncertainty of the demand is an important factor that could influence the accuracy of the action. Users may probably change their intention while the service robot is holding an object. "Move a box for me please" and "move the fridge to the right place" should be included in this task bank. "Move" is defined as a feature verb for this action.

4.1.4 Task: find an object

Finding an object together with its user is another typical task for a service robot. Though human beings have the most creative ability in the world, they can hardly have better performance than robots on memory. Especially interfacing to huge databases with a lot of similar elements such as library and archives, robots have a decisive edge. Take a simple example, there are many books in one library. Every book has been located in a fixed position and has a label on it. Human beings are unable to remember the location for each book, and in fact there is no needs do so. Suppose a service robot with a position map in its storage can make this task possible. Once its user wants to get a book, file or document in library, the only thing he or she need to do is to send a command to the robot: “Find ‘Gone with the Wind’ please”. The service robot will quickly search its database for where the book is and go to get it. Moreover if this book is not available or even not recorded, an error response will be given. As a result, these sorts of command can be used independently or combined with Task: move an object. Different commands such as: “find out this book?”, “find out if this book is available?” can be classified into the same task bank. “Find” is defined as a feature verb for this action.

4.2 Representations of verbs

When service robots communicate with humans verbs represent the crucial action of intentions. These keywords have been used to identify and to classify tasks. As mentioned above, “pass an object” and “move an object” are two typical tasks. The commands for these tasks can be: “please give me a cup of tea”, “bring a glass of water to me”, “help me take this table”, or “please help me carry this TV”, etc. In order to pick up a verb from a user command, Natural Language Processing (NLP) techniques can be used to process the commands, which are considered as text input. The initial input consists of a target word along with a portion of the text in which it is embedded, which is called its context. Here, the target word is the verb and the sentence is its context.

In terms of human being's experience in the use of language, some verbs have different meanings; however, they can have the same meaning when they collocate with some relevant nouns (Guo 2009). Therefore the meaning of the target word can be figured out through the analysis of the structure of a sentence. For instance, when "pass" and "give" are collocated with "drink", in these collocations, two verbs have the same meaning, but usually they do not. This means these two words have similar collocation features, and the collocation feature of verbs will be used to cluster the verbs. Two methods will be applied in the following sections to calculate the distance between verbs and nouns.

4.3 Verbs semantic similarity

There is a duality of word and document clustering observed by Dhillon (2001). Duality of word and document clustering presents that word cluster and document cluster can be reasoned from each other. Inspired by duality of word and document clustering and Resnik's study on selectional preference (Resnik, 1999), duality of verb and noun clustering is assumed in this study. Duality of verb and noun clustering states that verb clustering is able to be established by the induction of nouns clustering, while nouns is able to be established by the induction of verbs clustering. In this study, clusters of noun clustering are assumed to correspond to be sets of hyponyms in WordNet. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Therefore, clusters of verbs can be induced by clusters of nouns.

4.3.1 Cosine similarity

Cao et al (2009) proposed how to define verb feature vectors using relevant nouns. For example, when people using “move”, people can say “move the table”, “move the bed” or “move the box”, but not “pass the table” or “pass the bed”. Correspondingly, “move me a cup of tea” or “move me a glass of water” does not make any sense. Instead, people would say “pass me a cup of tea” or “give me a glass of water”. Therefore, people can choose some nouns to compose a vector to determine the target verbs’ feature vectors, and use these feature vectors to cluster these verbs and their related commands into task categories.

Collection of nouns and verbs is chosen as:

{cup, beer, tool, bed, fridge, box, soup, water, apple}

{pass, give, bring, move, take, carry, feed, need, drink, support}

This vector is used to define the chosen verbs’ feature vectors. If a verb can collocate with the relevant noun, the corresponding noun will be indicated by “1”; if not, it will indicated by “0”. Using such a method, the corresponding feature vectors for ‘move’ and ‘pass’ are {0,0,1,1,1,1,0,0,0} and {1,1,1,0,0,1,1,1,1}. Each verb vector is presented in Table 4.1.

Table 4.1 The verbs' feature vectors

noun verb	cup	beer	tool	bed	fridge	box	soup	water	apple
pass	1	1	1	0	0	1	1	1	1
give	1	1	1	0	0	1	1	1	1
bring	1	1	1	0	0	1	1	1	1
move	0	0	1	1	1	1	0	0	0
take	0	0	1	1	1	1	0	0	0
carry	0	0	1	1	1	1	0	0	0
feed	0	1	0	0	0	0	1	1	1
need	0	1	0	0	0	0	1	1	1
drink	0	1	0	0	0	0	1	1	1
support	0	0	0	0	0	0	0	0	0

After defining a verb's feature vector, the similarity of verbs could be calculated by using equation 4.1:

Similarity of (move, pass) =

$$\frac{0*1+0*1+1*1+0*1+0*1+1*1+0*1+0*1+0*1}{\sqrt{(1^2+1^2+1^2+0^2+0^2+1^2+1^2+1^2+1^2)*(0^2+0^2+1^2+1^2+1^2+1^2+0^2+0^2+0^2)}}$$

$$=\frac{2}{\sqrt{28}}=0.38 \qquad \text{Equation 4.1}$$

The cosine algorithm can calculate the similarity of verbs by using few steps. However, the fatal drawback is the similarity between verbs and nouns is defined by human habit. The following algorithm is based on the corpus and the result will be much more convincing.

4.3.2 Pointwise Mutual Information (PMI)

Pointwise Mutual Information is a concept in information theory and it is a measure of correlation between the two sets of events (mutual dependence). The definition of average mutual information is:

$$I(X;Y)=\sum_{y\in Y}\sum_{x\in X}p(x,y)\log(\frac{p(x,y)}{p_1(x)p_2(y)}) \qquad \text{Equation 4.2}$$

Mutual information I $(x_i;y_j)$ is the statistical average in the joint probability space P (X;Y). The average mutual information I (X;Y) overcomes the randomness in the mutual information I $(x_i;y_j)$, and become a certain amount, as shown in Equation 4.2.

The mutual information model is commonly used in Natural Language Processing (NLP). Using mutual information of feature extraction is based on the following assumptions: a particular element has high frequency; however, another element has relatively low frequency and the mutual information between these two elements is large. Mutual information is commonly used as the measure of characteristics of words and categories. If the feature of the words belongs to the same classification then they have the maximum of mutual information. Since this method does not require any assumptions of category, feature words and the nature of the relationship, it is suitable for the characteristics of text classification.

In this study, the simplest method for find collocations in a corpus is counting. If two words occur together often, then that is evidence that they have a special relationship, though that is not simply explained as the function that results from their combination. As the result of pre-processing, the frequency of each useful word can be counted. Therefore, probabilities of verbs, nouns, and their co-occurrence can be calculated. In Equation 4.2, variable x represents verb and y represents noun. The probability of co-occurrence of verbs and nouns (the joint probability P (verb, noun)) is compared with the probabilities of observing verbs and nouns independently (the chance probability P (verb) and P (noun)). If there is a genuine association between the verb and noun, then the joint probabilities P (verb, noun) will be much larger than chance P (verb)*P (noun). If there is no interesting relationship between the verb and noun, then P (verb, noun) will be almost the same as P (verb)*P (noun). If the verb and noun are in complementary distribution, then P (verb, noun) will be much less than P (verb)*P (noun). Probability P (verb) and P (noun) are estimated by counting the number of occurrences of the verb and noun in a corpus, and normalising by N (the number of the collocations), the size of the corpus. Joint probability, P (verb, noun), is estimated by counting the number of times that verb is followed by noun, and normalising by N.

For a simple example, the number of times which a word repeatedly appears in the corpus has been counted as in Table 4.2 (Church, K et al 1991):

Table 4.2 Frequency and PMI of verbs and nouns

I(x;y)	P(x,y)	P(x)	P(y)	x	y
10.23	8	7809	36	Move	Table
6.41	11	7809	76	Move	Chair
6.71	22	7809	115	Move	Box

Table 4.2 shows the mutual information and frequency values for three pairs of words. The frequency values were computed over one corpus, where N = 44.3 million words. In this table P(x,y) is joint probability and P(x), P(y) presents the probability of each word. The table shows that I (move, table) has a mutual information value of 10.23, since $\log_2((8 \times N)/(7809 \times 36)) = 10.23$. The results of mutual information value are ranked from high to low. Therefore the noun-verb pair which has very high mutual information values is supposed to be strongly associated. From the table, the correlation of "move" and "table" delivers more relevant results than other pairs.

4.3.3 Bipartite graph

A bipartite graph is a special case of a k-partite graph with k=2. On a bipartite graph, vertices are decomposed into two disjoint sets. Vertices within the same set are not allowed to be adjacent. Therefore, an edge on a bipartite graph can only be used to connect two vertices which belong to two different disjoint sets. For example, the graph colouring problem specifies that there are no two adjacent vertices that share the same colour. Hence the problem can be induced by the construction of bipartite graph. A bipartite graph G = (U, V, E) can be constructed in terms of the statement of the graph colouring problem, where U represents a set of nodes coloured blue, V represents a set of nodes coloured green and E is the set of edges. One of the endpoints of each edge is coloured blue, and another is coloured with green. No edge exists to connect two vertices if they are in the same set. In contrast, such a colouring is impossible in the case of a non-bipartite graph, such as a triangle: after one node is coloured blue and another green, the third vertex of the triangle is connected to vertices of both colours, preventing it from being assigned either colour.

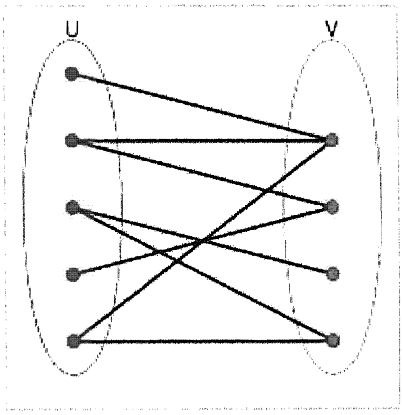


Figure 4.1 Example of bipartite graph

On this graph, vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V. U and V are independent sets and a bipartite graph is a graph that does not contain any odd-length cycles.

Table 4.3 PMI for bipartite graph

noun	Verb	PMI
box	pass	4.56
box	move	6.71
table	move	10.23
chair	move	6.41
pen	pass	9.37
book	pass	8.55
book	move	3.88

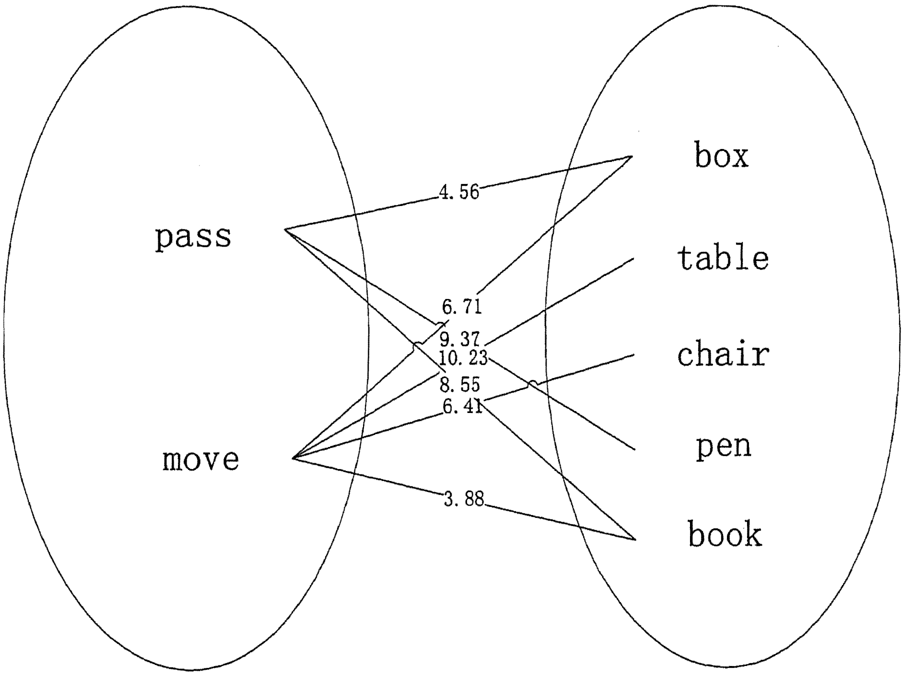


Figure 4.2 Example of PMI on bipartite graph

The bipartite graph in Figure 4.2 is used to establish mappings between verbs and nouns. According to Table 4.3, weights of graph edges are computed with the similarity measure PMI in this study. Bipartite graphs are appropriate for matching problems. The advantage of the combination is that it allows mappings to have many concept correspondences and is easy to understand.

4.3.4 Weighted Jaccard similarity measurement

In order to cluster verbs, the semantic similarity between verbs also needs to be explored. The semantic similarity between two verbs can be estimated through Weighted Jaccard similarity measurement (Equation 4.3). The Weighted Jaccard coefficient is known as a statistic used for comparing the similarity and diversity of sample sets. In general the result is the intersection divide union.(*w* is defined as the verbs and n is defined as the nouns)

$$WJ(\omega_1,\omega_1)=\frac{\sum_{N(\omega_1)\cap N(\omega_2)}\min(PMI(\omega_1,n_i),PMI(\omega_2,n_i))}{\sum_{N(\omega_1)\cup N(\omega_2)}\max(PMI(\omega_1,n_i),PMI(\omega_2,n_i))} \qquad \text{Equation 4.3}$$

The Weighted Jaccard measure considers a global and a local weight for each attribute. The global weight g_w depends on how many different words are associated with a given attribute. The local weight l_w is based on the frequency of the attribute with a given word. They are computed by the following formulas:

$$gw(n_i)=1-\sum_i\frac{|p_{ij}\log(p_{ij})|}{SUM} \qquad \text{Equation 4.4}$$

$$lw(w_i,n_i)=\log(freq\ of\ n_i\ with\ w_i) \qquad \text{Equation 4.5}$$

The Sum is the total number of relations extracted from the corpus and p_{ij} is defined as follows:

$$p_{ij} = \frac{freq\ of\ n_i\ with\ w_i}{total\ of\ n_i\ for\ w_i} \qquad \text{Equation 4.6}$$

The semantically similar words have been extracted from the corpus. The probability of each noun, verb and their collocations has been calculated using PMI. By computing this approach, the similarity of verbs can be extracted. The result shows when the global weight is very high, it contributes to make these words semantically close.

For example, consider the previous bipartite graph of nouns and verbs in Figure 4.2. The Weighted Jaccard similarity of "pass" and "move" can be computed with the following procedure. First, two sets of nouns with respect to "pass" and "move" are explored, namely N("pass") and N("move"). The intersection of N("pass") and N("move") and the union of N("pass") and N("move") are then obtained, respectively Equation 4.8, Equation 4.9. Edges of the bipartite graph are also weighted by PMI. Subsequently, Weighted Jaccard similarity is applied to calculate the similarity of "pass" and "move" (i.e. Equation 4.10). The similarity of "pass" and "move" is therefore estimated as 0.271 in this example. Henceforth the similarity of "pass" and "move" can be used in the verbs clustering algorithm (see Section 4.6).

$$N("pass") \cap N("move") = \{box, book\} \qquad \text{Equation 4.7}$$

$$N("pass") \cup N("move") = \{box, table, chair, pen, book\} \qquad \text{Equation 4.8}$$

$$\begin{aligned} WJ("pass","move") &= \frac{3.88+6.71}{4.56+10.23+6.41+9.37+8.55} \\ &= 0.271 \end{aligned} \qquad \text{Equation 4.9}$$

4.3.5 K-medoids cluster

K-means is an unsupervised statistical learning method for clustering. Given a set of observations (x_1, x_2..., x_n) where each observation is an n-dimensional real vector. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Suppose, k clusters S={S_1, $S_{2,...,}S_k$} so as to minimize the within-cluster sum of squares:

$$\arg_S \min \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2 \qquad \text{Equation 4.10}$$

where μ_i is the mean of points in S_i.

The semantic similarity between two verbs can be estimated through Weighted Jaccard similarity measurement. The K-medoids algorithm is a clustering algorithm related to the K-means algorithm and the medoidshift algorithm. Both the K-means and K-medoids algorithms are partitional (breaking the dataset into groups) and both attempt to minimize squared error. The standard K-medoids clustering algorithm

The K-medoids algorithm is a common clustering algorithm in Partitioning Methods. Around the centre of the division of PAM (Partitioning Around Medoid) is proposed as the first one of K-medoids algorithms. Compared with K-means, K-medoids has a better performance for dealing with noises and outliers since K-medoids is seldom affected by extreme data, unlike K-means. However, it requires a high level of implementation. The procedure for K-medoids is as follows:

1. Initialize: randomly select k verbs of the n verbs as the medoids
2. Associate each verb to the most similar medoid verb
3. For each medoid m
4. For each non-medoid verb o
5. Swap m and o and compute the total similarity weights of the configuration
6. Select the configuration with the maximum similarity
7. Repeat steps 2 to 5 until there is no change in the medoid

For example: If we want to clustering the follow data set in Table 4.4 of ten objects(X_j) into two clusters.

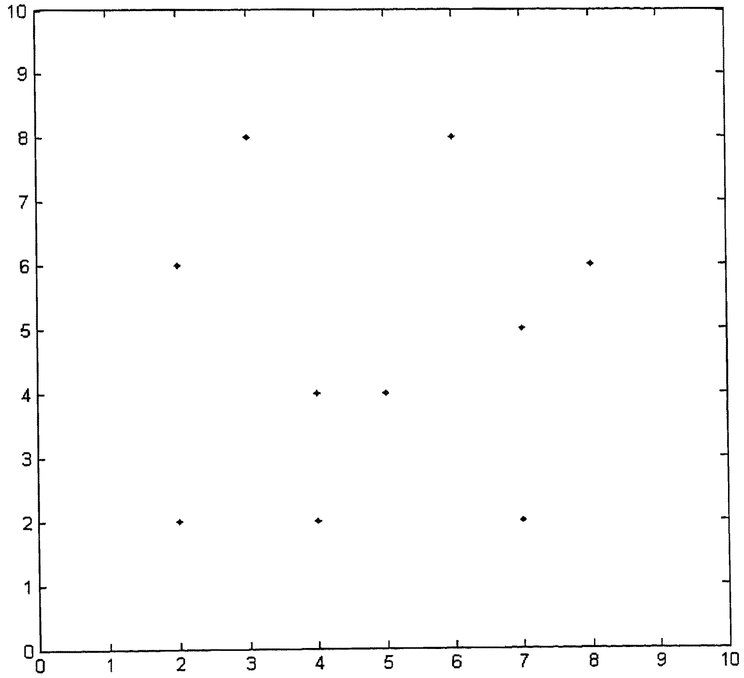


Figure 4.3 Distribution of the data

Table 4.4 Original data of ten objects

Object	X-axis	Y-axis
X_1	2	2
X_2	2	6
X_3	3	8
X_4	4	4
X_5	4	2
X_6	5	4
X_7	6	8
X_8	7	5
X_9	7	2
X_{10}	8	6

Table 4.5 Distance of X_j with C_1

C_1		Data objects (X_j)		Distance
4	2	2	2	2
4	2	2	6	6
4	2	3	8	7
4	2	4	4	2
4	2	6	8	8
4	2	7	5	6
4	2	7	2	3
4	2	8	6	8

Step 1:

Initialize K centre and assume c_1 = (4, 2) and c_2 = (5, 4). So here c_1 and c_2 are selected as medoids. Calculate the distance so as to associate each data object to its nearest medoid.

Table 4.6 Distance of X_j with C_2

C_2		Data objects (X_j)		Distance
5	4	2	2	5
5	4	2	6	5
5	4	3	8	6
5	4	4	4	1
5	4	6	8	5
5	4	7	5	3
5	4	7	2	4
5	4	8	6	5

So the clusters then become: Cluster_1 = {(4,2)(2,2)(7,2)}

Cluster_2 = {(5,4)(2,6)(3,8)(4,4)(6,8)(7,5)(8,6)}

The distance between any two points is found using this formula:

$$\text{Distance}(x,c) = \sum_{i=1}^{d} |x - c| \qquad \text{Equation 4.11}$$

Where x is any data object, c is the medoid, and d is the dimension of the object which in this case is 2.

Total cost is the summation of the distance of data object from its medoid in its cluster so here:

Total cost = {cost ((4,2)(2,2)) + cost ((4,2)(7,2)) + cost ((5,4)(2,6)) + cost ((5,4)(3,8)) + cost ((5,4)(4,4)) + cost ((5,4)(6,8)) + cost ((5,4)(7,5)) + cost((5,4)(8,6))} = (2+3)+(6+5+1+5+3+5) = 24

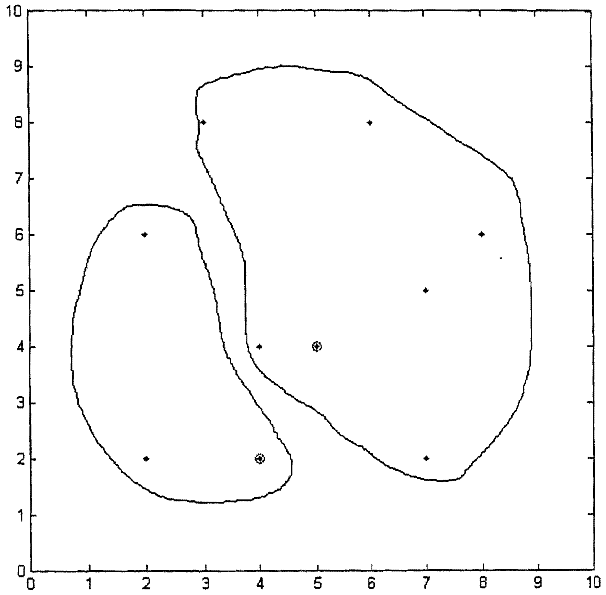


Figure 4.4 Clusters after step 1

Step 2:

Select the nonmedoid O randomly and assume O = (7, 2). Therefore, the medoids are c_1 = (4, 2) and O = (7, 2). Calculate the total cost by using Table 4.7.

Table 4.7 Distance of X_j with O

O		Data objects (X_j)		Distance
7	2	2	2	5
7	2	2	6	9
7	2	3	8	10
7	2	4	4	5
7	2	5	4	4
7	2	6	8	7
7	2	7	5	3
7	2	8	6	8

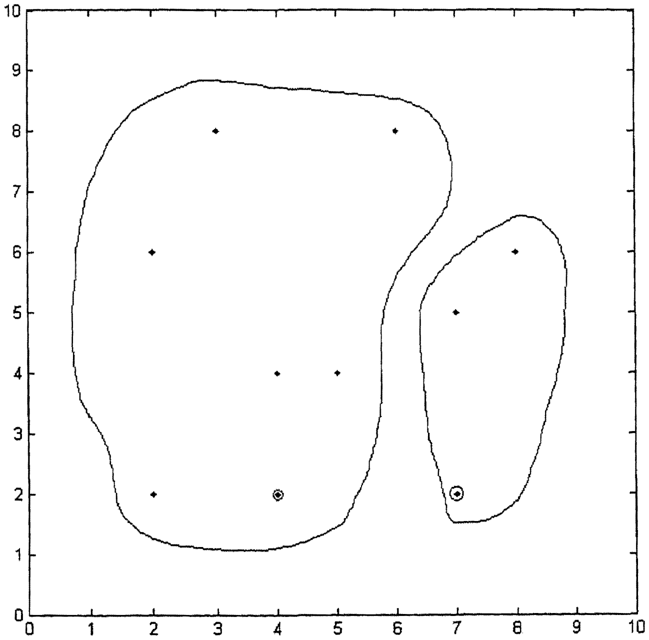


Figure 4.5 Clusters after step 2

Total cost = 2+6+7+2+3+6+3+5 = 34

So the cost of swapping medoid from c_2 to O is:

S = current total cost – past total cost

$= 34 - 24$

$= 10 > 0$

As the result shows, moving to O is a bad choice and the previous choice is better. The previous choice is selected and fixed as the medoid of the cluster. Iterations of the algorithm are terminated if the medoid of each cluster has been selected and fixed; otherwise, iterations will be continued until the best choice of the medoid of each cluster has been obtained. Distances of nonmedoid data points and selected medoids are then computed with Equation 4.11 once medoids of all clusters have been selected. Each nonmedoid data point is subsequently assigned to the cluster if the minimum distance of the nonmedoid data point and the medoid of the cluster have been achieved.

4.4 Summary

This chapter introduced typical task classification and a serial algorithms applied in the verb cluster. Section 4.1 described 4 kinds of task for service robot and the following section the algorithms including PMI, WJ, bipartite graph and the K-mediods cluster. One example which data is from (Church, K et al 1991) was applied in the cluster. The real data from natural corpus will be used and analysis given in the next chapter.

CHAPTER 5. EXPERIMENTS AND ANALYSIS

The previous chapters introduced an integrated verb clustering model based on quite a several assumptions. Some of these assumptions seem somewhat unrealistic and could lead users to doubt the practicability of the model. Therefore, a real experimental model is presented in this chapter. Results of the classification may not be optimal, but they are good approximations. At least, they provide useful guidelines for clustering verbs for Active Robot Learning.

This integrated model consists of two main components: pre-processing and verb clustering. The pre-processing is used on the text-corpus in order to make collection of the statistics of characteristics easier. It used to identify the verb which will be analysed in the final classifier as well. The pre-processing is also known as the contextual information extractor. In nature, the contextual information is a simplified representation of a command, that is, a collocation of a verb and a noun will be used to represent the command contained in an utterance. Verb clustering is used for task classification. Depending on the prototype of each task, the process of verb clustering attempts to discover the hidden knowledge about the groups of verbs that are contained in the corpus. Several mathematical algorithms are employed in the following parser including PMI, WJ and K-medoid. After this procedure similar verbs can be grouped into different clusters. Consequently, similar verbs for each prototype can be easily observed. The architecture of the model is shown in Figure 5.1:

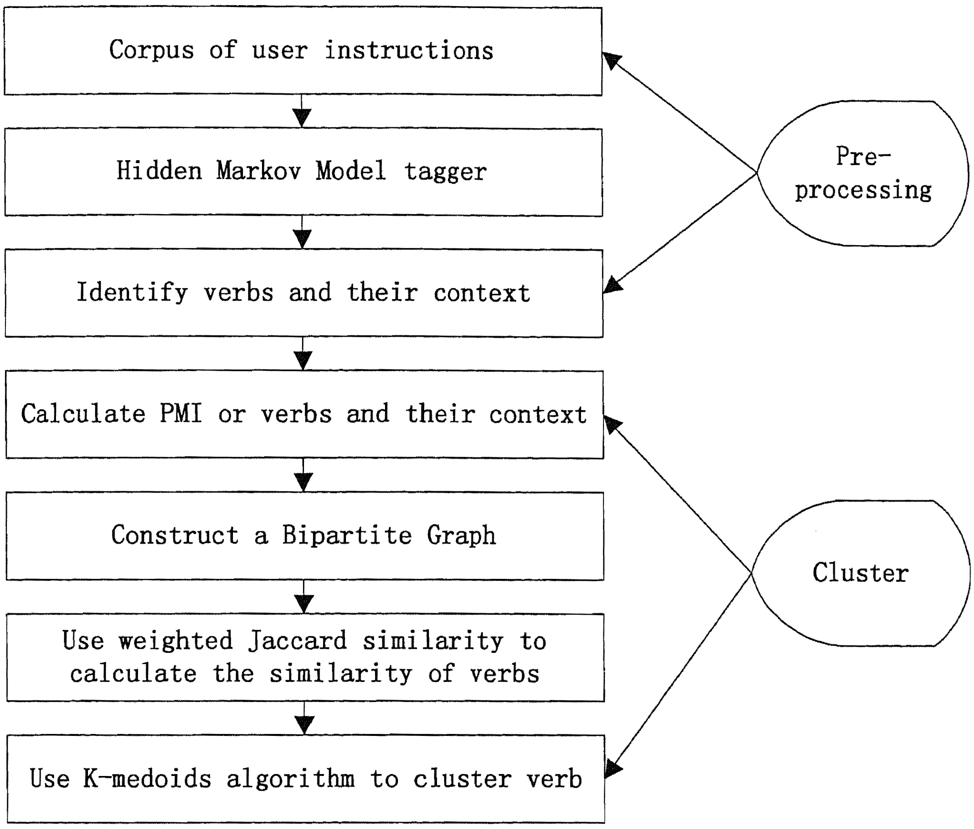

Corpus of user instructions
Hidden Markov Model tagger
Identify verbs and their context
Calculate PMI or verbs and their context
Construct a Bipartite Graph
Use weighted Jaccard similarity to calculate the similarity of verbs
Use K-medoids algorithm to cluster verb
Pre-processing
Cluster

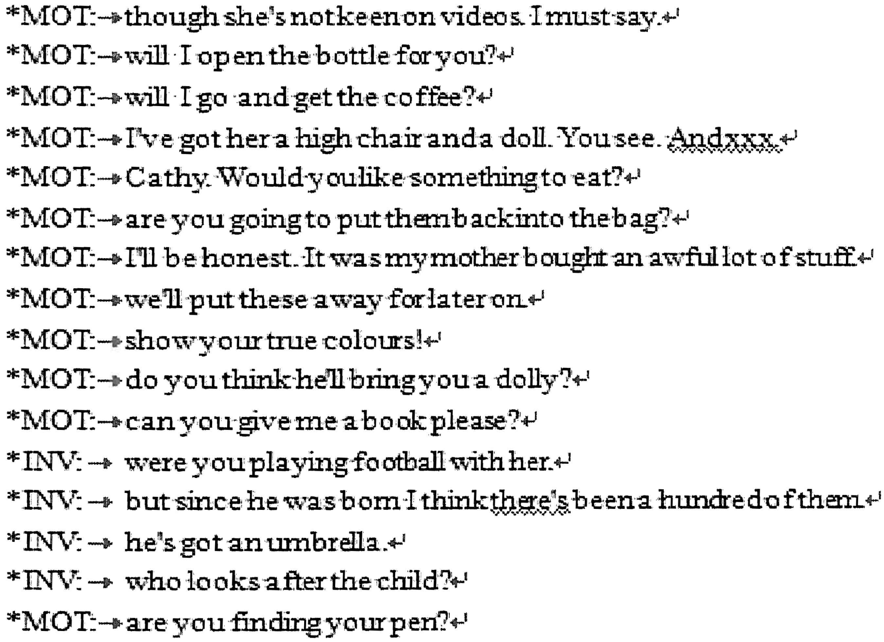
Figure 5.1 Architecture of the integrated verb clustering model

5.1 Contextual information extractor

Human intention is hidden in the human command. The text–corpus can be seen as a collection of commands. In order to figure out the intention from the command, the text-corpus which is very complex must be analysed. The intention is mostly indicated by verbs and nouns. The other words in the corpus will be removed since these words do not aid robot understanding to the intention contained in user commands. This extractor is applied to identify the prototype of verb and noun.

5.1.1 Corpus:

The test corpus is selected from the partial Manchester corpus introduced in Chapter 3, combined with normal conversations from Internet. This corpus can be swiftly and easily enriched with additional information. This corpus consists of a number of machine-readable commands and can be reused in the future. The following sample of the corpus is the part of the complete corpus used in the experiments.

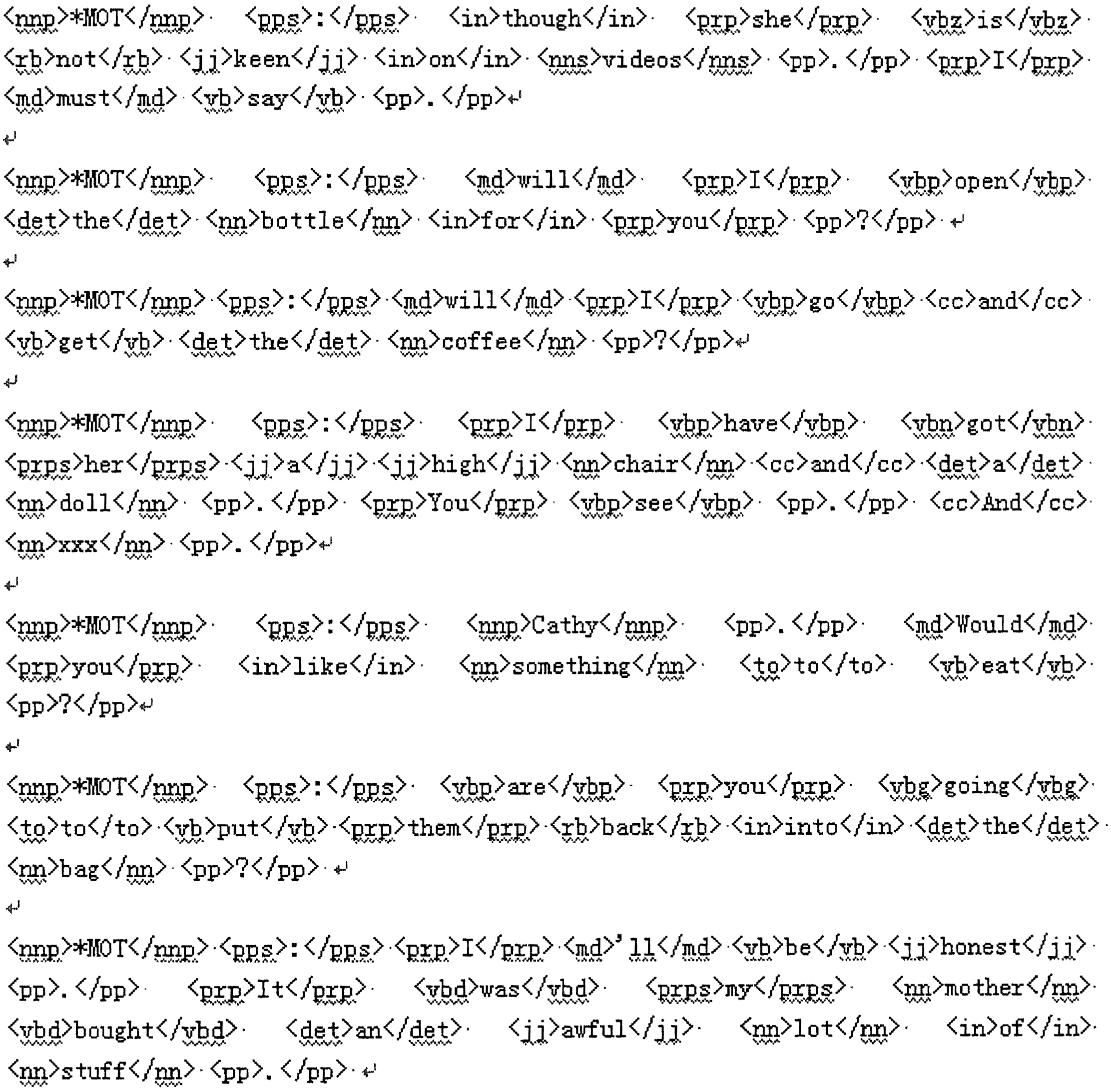


```
*MOT:→though she's not keen on videos. I must say.↵
*MOT:→will I open the bottle for you?↵
*MOT:→will I go and get the coffee?↵
*MOT:→I've got her a high chair and a doll. You see. Andxxx.↵
*MOT:→Cathy. Would you like something to eat?↵
*MOT:→are you going to put them back into the bag?↵
*MOT:→I'll be honest. It was my mother bought an awful lot of stuff.↵
*MOT:→we'll put these away for later on.↵
*MOT:→show your true colours!↵
*MOT:→do you think he'll bring you a dolly?↵
*MOT:→can you give me a book please?↵
*INV: →  were you playing football with her.↵
*INV: →  but since he was born I think there's been a hundred of them.↵
*INV: →  he's got an umbrella.↵
*INV: →  who looks after the child?↵
*MOT:→are you finding your pen?↵
```

Figure 5.2 Input corpus

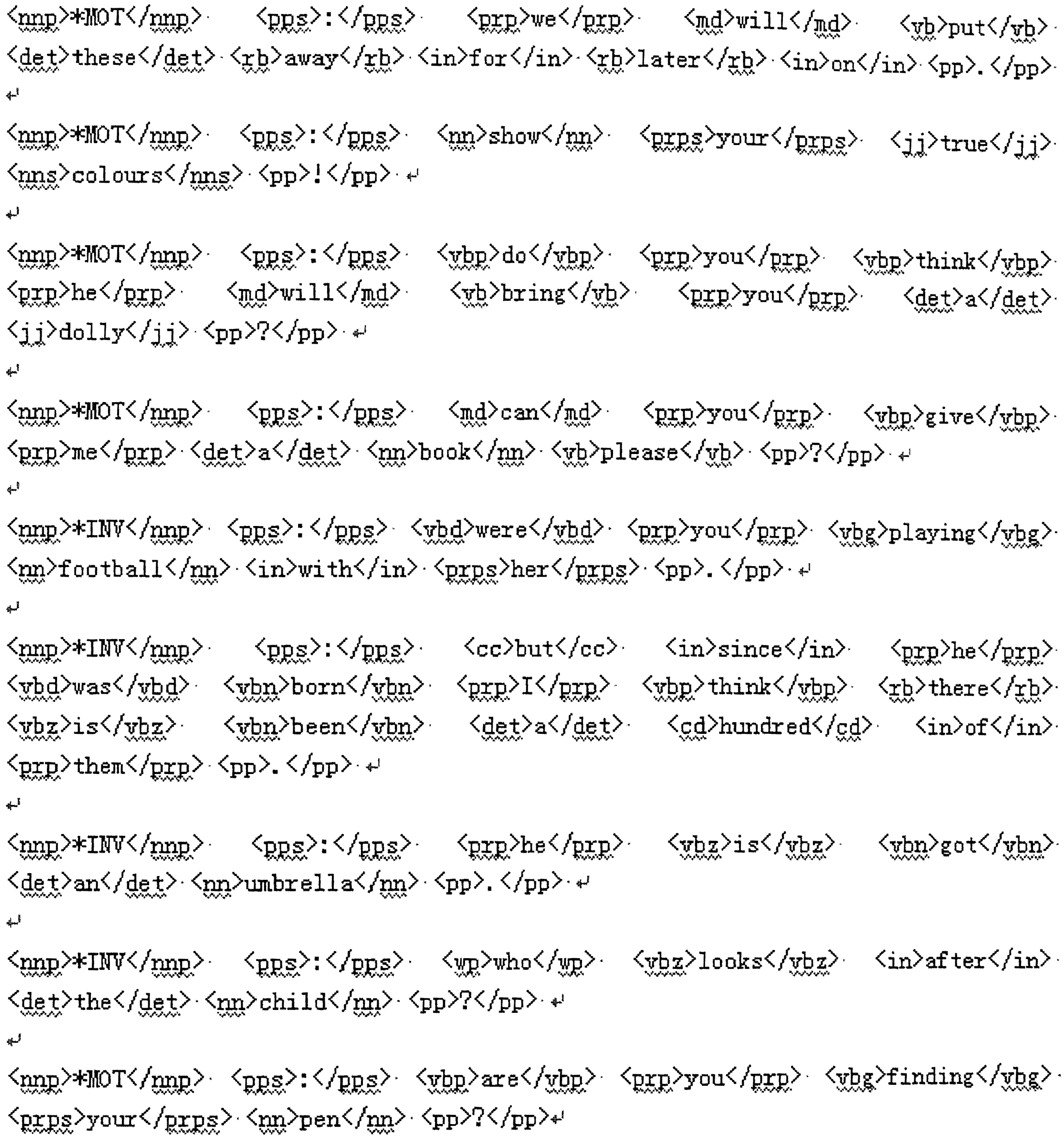
5.1.2 PoS tag:

After inputting the corpus, Link Grammar is employed to tag the corpus. It is an application from CPAN (Comprehensive Perl Archive Network) named "Lingua-EN-Tagger"; a HMM (Hidden Markov Model) is used in this programming. HMM taggers work well when we have a large tagged training set and could even tag a text from a specialized domain or text in a foreign language.



```
<nnp>*MOT</nnp> <pps>:</pps> <in>though</in> <prp>she</prp> <vbz>is</vbz> <rb>not</rb> <jj>keen</jj> <in>on</in> <nns>videos</nns> <pp>.</pp> <prp>I</prp> <md>must</md> <vb>say</vb> <pp>.</pp>

<nnp>*MOT</nnp> <pps>:</pps> <md>will</md> <prp>I</prp> <vbp>open</vbp> <det>the</det> <nn>bottle</nn> <in>for</in> <prp>you</prp> <pp>?</pp>

<nnp>*MOT</nnp> <pps>:</pps> <md>will</md> <prp>I</prp> <vbp>go</vbp> <cc>and</cc> <vb>get</vb> <det>the</det> <nn>coffee</nn> <pp>?</pp>

<nnp>*MOT</nnp> <pps>:</pps> <prp>I</prp> <vbp>have</vbp> <vbn>got</vbn> <prps>her</prps> <jj>a</jj> <jj>high</jj> <nn>chair</nn> <cc>and</cc> <det>a</det> <nn>doll</nn> <pp>.</pp> <prp>You</prp> <vbp>see</vbp> <pp>.</pp> <cc>And</cc> <nn>xxx</nn> <pp>.</pp>

<nnp>*MOT</nnp> <pps>:</pps> <nnp>Cathy</nnp> <pp>.</pp> <md>Would</md> <prp>you</prp> <in>like</in> <nn>something</nn> <to>to</to> <vb>eat</vb> <pp>?</pp>

<nnp>*MOT</nnp> <pps>:</pps> <vbp>are</vbp> <prp>you</prp> <vbg>going</vbg> <to>to</to> <vb>put</vb> <prp>them</prp> <rb>back</rb> <in>into</in> <det>the</det> <nn>bag</nn> <pp>?</pp>

<nnp>*MOT</nnp> <pps>:</pps> <prp>I</prp> <md>'ll</md> <vb>be</vb> <jj>honest</jj> <pp>.</pp> <prp>It</prp> <vbd>was</vbd> <prps>my</prps> <nn>mother</nn> <vbd>bought</vbd> <det>an</det> <jj>awful</jj> <nn>lot</nn> <in>of</in> <nn>stuff</nn> <pp>.</pp>
```

Figure 5.3 Part1 of tagged corpus (exact copy of original)



```
<nnp>*MOT</nnp> <pps>:</pps> <prp>we</prp> <md>will</md> <vb>put</vb> <det>these</det> <rb>away</rb> <in>for</in> <rb>later</rb> <in>on</in> <pp>.</pp> ↵
↵
<nnp>*MOT</nnp> <pps>:</pps> <nn>show</nn> <prps>your</prps> <jj>true</jj> <nns>colours</nns> <pp>!</pp> ↵
↵
<nnp>*MOT</nnp> <pps>:</pps> <vbp>do</vbp> <prp>you</prp> <vbp>think</vbp> <prp>he</prp> <md>will</md> <vb>bring</vb> <prp>you</prp> <det>a</det> <jj>dolly</jj> <pp>?</pp> ↵
↵
<nnp>*MOT</nnp> <pps>:</pps> <md>can</md> <prp>you</prp> <vbp>give</vbp> <prp>me</prp> <det>a</det> <nn>book</nn> <vb>please</vb> <pp>?</pp> ↵
↵
<nnp>*INV</nnp> <pps>:</pps> <vbd>were</vbd> <prp>you</prp> <vbg>playing</vbg> <nn>football</nn> <in>with</in> <prps>her</prps> <pp>.</pp> ↵
↵
<nnp>*INV</nnp> <pps>:</pps> <cc>but</cc> <in>since</in> <prp>he</prp> <vbd>was</vbd> <vbn>born</vbn> <prp>I</prp> <vbp>think</vbp> <rb>there</rb> <vbz>is</vbz> <vbn>been</vbn> <det>a</det> <cd>hundred</cd> <in>of</in> <prp>them</prp> <pp>.</pp> ↵
↵
<nnp>*INV</nnp> <pps>:</pps> <prp>he</prp> <vbz>is</vbz> <vbn>got</vbn> <det>an</det> <nn>umbrella</nn> <pp>.</pp> ↵
↵
<nnp>*INV</nnp> <pps>:</pps> <wp>who</wp> <vbz>looks</vbz> <in>after</in> <det>the</det> <nn>child</nn> <pp>?</pp> ↵
↵
<nnp>*MOT</nnp> <pps>:</pps> <vbp>are</vbp> <prp>you</prp> <vbg>finding</vbg> <prps>your</prps> <nn>pen</nn> <pp>?</pp>↵
```

Figure 5.4 Part2 of tagged corpus (exact copy of original)

Part1 and Part2 are samples of corpus which have been tagged. Every word has an opening and a closing tag beside it. The corpus is separated into utterances. Depending on the architecture of the sentence, collocations can be extracted as follows.

5.1.3 Extractor:

A script has been developed to indentify verbs and nouns. Each sentence has a simple structure, subordinate clauses are not present in the corpus. Hence we can extract one crucial verb and some pairs of nouns in one sentence. When one verb links with two or more nouns such as “Take the book and pen” both the nouns can be used to construct the collocation.

Table 5.1 Result of pre-processing

open	bottle
get	coffee
get	chair, doll
eat	something
put	bag
put	
show	color
bring	dolly
give	book
play	football
born	
get	umbrella
find	pen

Many collocations can be extracted from the corpus. However, there are two cases in which collocations are difficult to extract. Some utterances have either verbs or nouns, and the referents of the target word are in other utterances in the corpus. In the following part, the approach to choosing a verb depending on the nouns is introduced.

5.2 Verb clustering

5.2.1 Pointwise Mutual Information

After counting the number of appearance, all collocations will be calculated by Equation 5.1:

$$I(X;Y) = \lg(\frac{p(verb, noun) \bullet N}{p(verb)p(noun)}) \qquad \text{Equation 5.1}$$

Table 5.2 Part1 of PMI between verbs and nouns

verb	noun	PMI(10base)	verb	noun	PMI(10base)
give	box	1.361193442	pass	point	2.841658656
give	spoon	1.518316862	pass	minute	2.841658656
give	pen	1.553078969	pass	spoon	2.540628660
give	knife	1.428140232	pass	fork	2.364537401
give	paper	0.969502383	pass	news	1.996560616
give	hand	2.030200223	pass	plate	2.364537401
give	coat	1.729170228	pass	knife	2.239598664

Table 5.3 Part2 of PMI between verbs and nouns

verb	noun	PMI(10base)	verb	noun	PMI(10base)
move	table	2.239598664	find	sausage	1.709696136
move	chair	1.586386151	find	table	1.207751297
move	desk	1.762477410	take	shirt	2.061414416
move	box	1.394500624	eat	corn	1.540628660
move	toy	1.035478682	eat	egg	1.637538673
move	house	2.239598664	eat	chip	1.938568669
move	bear	1.285356155	change	dress	2.029229732

Because the number of collocations is very large, they cannot be shown in here in full.

5.2.2 Bipartite graph

The bipartite graph is applied to present the semantic relatedness between verbs and nouns.

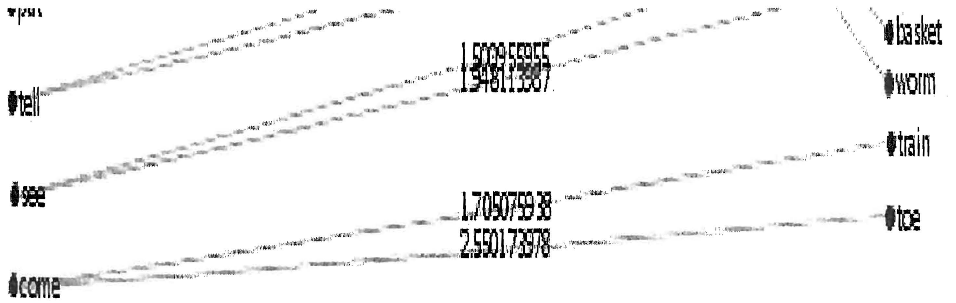


Figure 5.5 Partial enlargement of bipartite graph

Figure 5.5 is a partial enlargement of bipartite graph of PMI.

5.2.3 Weighted Jaccard similarity

The similarities of verbs are shown in Table 5.4. Due to limited space of the thesis, here, only part of whole table is presented. For example "0.0858" is the similarity between "pass" and "give". If two verbs do not share any nouns in the collocations, the similarity of these words is 0.

Table 5.4 Result of verbs similarity

	give	pass	hand	come	get	sit
give	1	-	-	-	-	-
pass	0.0858	1	0	0	0	0
hand	0.0388	0	1	0	0	0
come	0	0	0	1	0.078	0
get	0.0338	0	0	0.0780	1	0.0392
sit	0.0394	0	0	0	0.0392	1
put	0.0454	0	0	0	0	0.0682
change	0	0	0	0	0	0
play	0	0.1083	0	0	0	0
see	0	0	0	0	0	0
move	0.0410	0	0	0	0.0776	0
like	0	0	0	0	0.0381	0
make	0	0	0	0	0.0733	0
watch	0	0	0	0	0	0
catch	0	0	0	0	0	0
buy	0	0	0	0	0	0
try	0	0	0	0	0	0
find	0	0	0	0	0	0
turn	0	0	0	0	0	0.1087
bring	0	0	0	0	0.0485	0.0900
tell	0	0	0	0	0	0
take	0	0	0	0	0	0
look	0	0	0	0	0	0
want	0.0452	0	0	0	0	0
cut	0	0	0	0	0	0
allow	0	0.1852	0	0	0	0
eat	0	0	0	0	0	0

5.2.4 Wordnet

Home service is the key word in this study. Hence, the relevant nouns are common words in everyday's life. In other words, the objects that a service robot needs to recognise are the simple object in daily surroundings. In this experiment, Wordnet is applied to define the relevant nouns. Wordnet is a lexical database and consists of many sets of hyponyms. The following nouns are picked up from the set of house: dress, shoe, box, pen, knife, paper, coat, cup, book, phone, apple, baby, bottle, spoon, fork, plate, paper, strawberry etc. All of the collocations have been extracted from corpus and this step just decides which verb should be counted.

5.2.5 K-medoid

When the similarity of the verbs has been calculated, K-medoid is applied to calculate the final result of clustering. The iteration using verb similarity is different from the normal K-medoid. These similarities present the distance between two verbs. That means the distance between two points is known not the coordinate position.

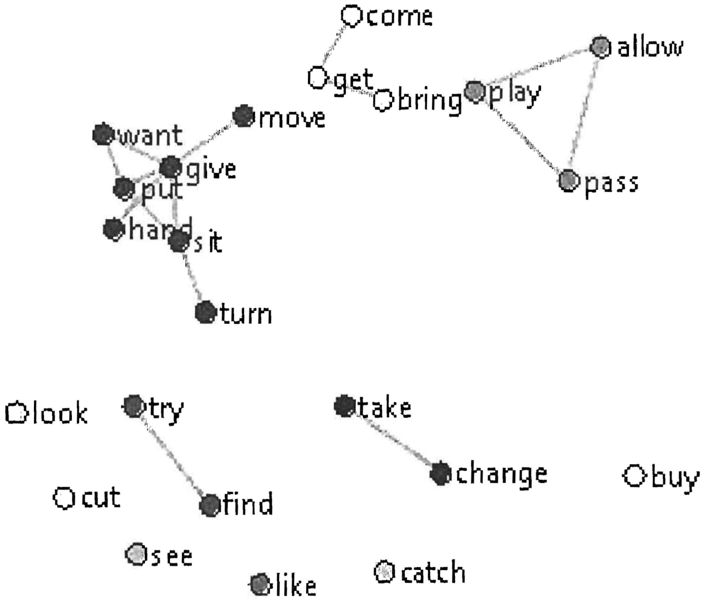


Figure 5.6 Initializing on 500 utterances

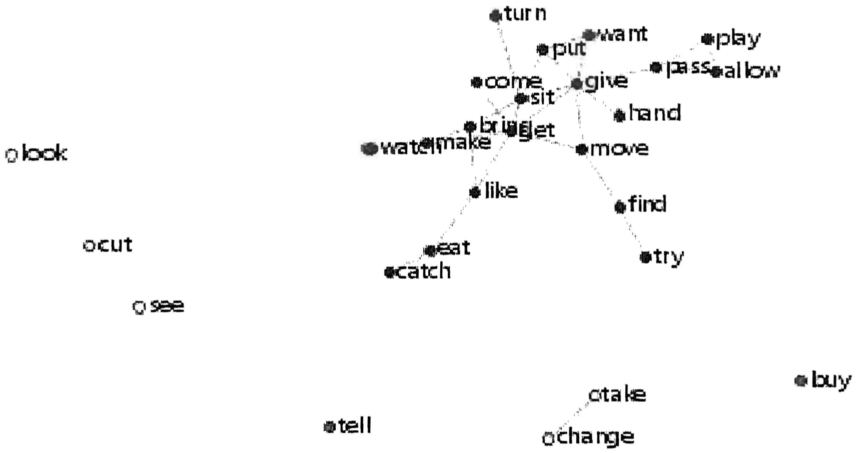


Figure 5.7 Initializing on 150792 utterances

Comparing the two graphs above, Figure 5.7 shows more relationships are found as the size of corpus becomes larger. At the first time, “give” and “pass” are related, but they are not clustered into the same cluster. Figure 5.8 cluster them into one cluster and it is the most comprehensive condition. Increasing the corpus can also minimize the error rate calculated by PMI.

The initializing of clustering is shown in Figure 5.9.

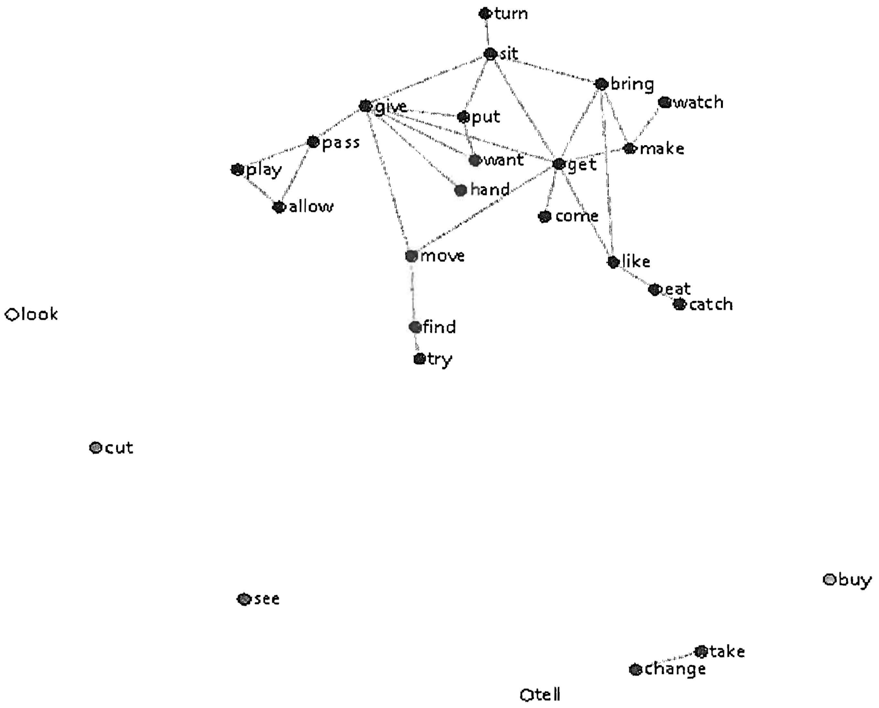


Figure 5.8 Initializing of clustering

In order to compare the performance of this cluster, different numbers of iteration are used to test the cluster. The following are the result of clustering 100, 200, and 2500.

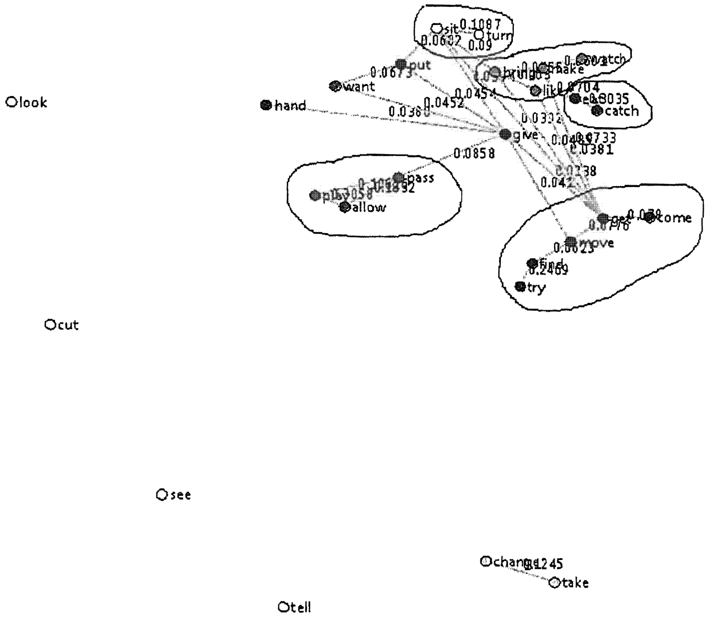


Figure 5.9 Result of 100times of iterations

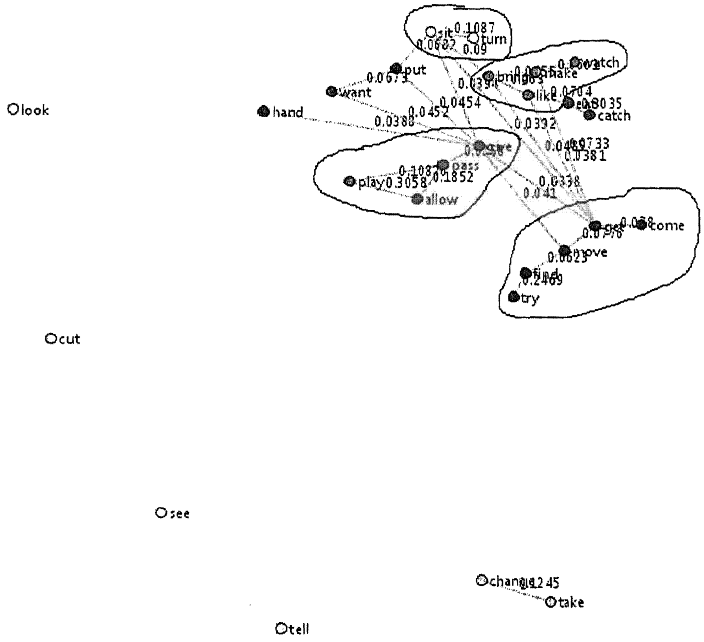


Figure 5.10 Result of 500 iterations

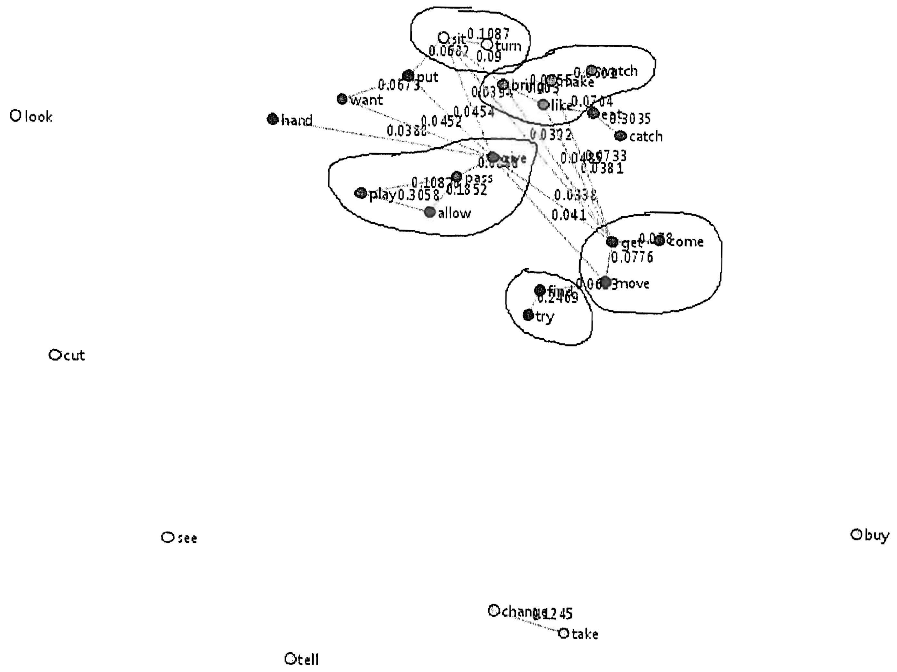


Figure 5.11 Result of 2500 iterations

After 2500 iterations, the result of clustering becomes increasingly stable. Obviously, as mentioned above, the "pass an object" task category has four words: "give, pass, play, allow". Although "give" has similar meaning to "pass" in everyday life, "play" and "allow" cannot make any sense in this cluster. This sort of situation happens in every cluster e.g. "move", "eat", and "find". There are three possible reasons for the mis-clustering. First, PMI lacks the capability to deal with data sparseness and the corpus used in this study may not be large enough. Therefore, some incorrect results arise in the experiments. Second, errors occurred in the PoS tagging and the contextual information extraction may also deliver unexpected results since different PoS tags and different contextual information will endow a word with different meanings. Third, the pre-processing of the corpus either may not completely remove the redundant information from the original corpus, such as the tag "*MOT", or may not correctly transform the tenses of a word to its prototype. As a result, several verbs have been classified by the relevant nouns; however, the final result is not precise for the above reasons.

5.3 Summary

This chapter presented the experiment. The data from Manchester corpus was pre-processed by the method proposed in chapter 3. The pre-processing result was clustered into different sets showed in section 5.2. However, This experimental result was just acceptable, and could hardly be employed in the TAB for the service robot. The following chapter will draw the conclusion and future work.

CHAPTER 6. CONCLUSIONS AND FURTHER WORK

6.1 Conclusions

In recent years, there have been growing trends in both service robots and Natural Language Processing. This research has considered a statistical approach for verb clustering base for service robots. At the beginning of this study, a survey of the state-of–the-art approached the area of interaction between service robots and humans. The hierarchical structure of verb clustering has been introduced, which is one component of Task Action Bank (TAB) in Active Robot Learning (ARL). Then, the result of experiments has verified the performance of the cluster. It can be concluded that:

- ❑ Established a robot corpus for analysing the command from users, based on Manchester corpus. This corpus can be enriched easily and swiftly and it can be a template for a machine-readable corpus.
- ❑ Developed a pre-processing parser in order to process the original corpus into collocations. The collocations can be identified successfully.
- ❑ Generalised four typical task categories, from which several specific taught tasks and their corresponding test actions, are derived. These four typical task categories are: “pass an object”, “move an object”, “feed the user” and “find an object”.
- ❑ Defined verb features by relevant nouns and established verb feature vectors, which can be used to calculate distances between verbs. These defined verbs are used to map commands to task categories.

□ Two sets of experiments were carried out on pre-processing and verb clustering. Experimental results show that the size of the corpus has a great influence on the accuracy and number of iterations of K-medoid, which also induced clearer clusters.

6.2 Further work

Although the result of verb clustering gives the classification of tasks based on feature verbs has statistical support, several extensions are still worthwhile for consideration.

□ One part of the integrated model is applied to process a corpus. The experimental results show Pointwise Mutual Information (PMI) could not work well when the dataset is sparse. Therefore, the machine-readable corpus which is employed in this experiment urgently needs to be enriched. More utterances which come from everyday life can make this corpus richer.

□ Pre-processing can process the corpus by itself. During the experiment, the time cost of pre-processing was very expensive. The time cost will be increased when a larger corpus is used. Therefore, the codes in the script of the pre-processing parser need to be optimized in the future.

□ Limited range of task categories is also a crucial part of the restriction. The whole model is established for a home service robot; however, many tasks are not included in these four categories. Defining more feature verbs can generalise more tasks into different categories and this can expand the verb clustering to other domains.

REFERENCES

Abney, S., "Part-of-speech tagging and partial parsing", in Corpus-Based Methods in Language and Speech Processing, in Young S. and Bloothooft, G. (Eds.), *Kluwer Academic Publishers*, Dordrecht, 1997.

Bellegarda, J. R., Butzberger, J. W., Chow, Y. L., Coccaro, N. B. and Naik, D., "A novel word clustering algorithm based on latent semantic analysis", In *Proceedings of the International Conference on Speech, Signal Processing*, Atlanta, GA, May 1996, pp. I172–I175.

Blackmore, B.S., Fountas, S., Vougioukas, S., Tang, L., Sorensen,C.G., Jorgensen, R., 2004. A method to define agricultural robot behaviours. *Automation Technology for Off-Road Equipment. ASAE Publications*, St. Joseph, MI, USA, pp. 155 - 161.

Breazeal, S Brooks, A., Chilongo, D., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J. and Lockerd, A., "Working Collaboratively with Humanoid Robots", In *Proceedings of the IEEE/RAS Fourth International Conference on Humanoid Robots* (Humanoids 2004), Los Angeles, CA. 253-272, 2004.

Brill, E. Resnik, P., "A rule-based approach to prepositional phrase attachment disambiguation" *Proceeding COLING '94 Proceedings of the 15th conference on Computational linguistics - Volume 2* 1994

Calinon, S. and Billard, A., “Teaching a humanoid robot to recognise and reproduce social cues”, In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communications*, (RO-MAN 2006), 2006.

Cao, T., Li, D., Jiang, D., Maple, C., Yue. Y., “Task classifier based on verb's feature vector”, In *Proceedings of the 15th International Conference on Automation & Computing*, University of Bedfordshire, Luton, UK, 19 September 2009.

Church, K., Gale, W., Hanks, P. and Hindle, D., “Using Statistics in Lexical Analysis”, *Bell Laboratories and Oxford University Press*, February 1989.

Daniel, D., Sleator, K., and Temperley, D., “Parsing English with a Link Grammar”, *Technical Report CMU-CS-91_196, Carnegie Mellon University*, October 1991.

Dhillon, I., “Co-clustering documents and words using bipartite spectral graph partitioning” *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* 2001

Dillmann, R., “Teaching and learning of robot tasks via observation of human performance”, *Robotics and Autonomous Systems*, Vol. 47, No. 2-3, 2004, pp. 109-116.

Firth, N., “MeetDustCart: The intelligent ‘bin on wheels’ robot that will come to pick up your rubbish when you call”, *http://www.dailymail.co.uk/sciencetech/article-1291142/DustCart-The-bin-wheels-robot-picks-rubbish-call.html* ,2010.

Fong, T., Nourbakhsh, I. and Dautenhahn, K., “A survey of socially interactive robots”, *Robotics and Autonomous Systems*, vol. 42, pp. 143-166, 2003.

Gabor Stepan, Andras Toth., “ACROBOTER: a ceiling based crawling, hoisting and swinging service robot platform”, *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium 2009*

Guo X., Li D., Clapworthy G. and Pritchett N., “A conditional mutual information based selectional association and word sense disambiguation”, In *Proceedings of IEEE Natural Language Processing and Knowledge Engineering 2009*, pp. 249-255, 2009.

Hassch, A., Hohenner, S., Huwel, S., Kleinehagenbrock, M., Lang, S., Toptsis, I., Fink, G. A., Fritsch, J., Wrede, B. and Sagerer, G., “BIRON-The Bielefeld robot companion”, *Conference Paper of Workshop on Advances in Service Robotics,* May, 2004, Stuttgart, Germany, pp. 27-32.

Holmes, V. M., Stowe, L., and Cupples, L., “Lexical expectations in parsing complement-verb sentences.” *Journal of Memory and Language*, 1989.

Iba, S., Paredis, C. J. J. and Khosla, P. K. , “Intention Aware Interactive Multi-Modal Robot Programming”, *International Conference on Intelligent Robots and Systems*, (IROS) 2003, Las Vegas, NV, 2003.

Jurafsky, D. and Martin, J., "Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics", *Prentice-Hall, Upper Saddle River*, NJ, 2000, pp.285-318.

Kanda, T., Ishiguro, H., Imai, M., Ono, T. and Mase, K., "A constructive approach for developing interactive humanoid robots", In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1265–1270, 2002.

Kawamura, K., Peters II, R. A., Bagchi, S., Iskarous, M. and Bishay, M., "Intelligent robotic systems in service of the disabled", *IEEE Transactions on Rehabilitation Engineering*, Vol. 1, No. 3, pp. 14-21, March 1995 and \Erratum, Vol. 1, No.11, November 1995.

Kelley, R., Tavakkoli, A., King, C., Nicolescu, M., Nicolescu, M. and Bebis, G., "Understanding Human Intentions via Hidden Markov Models in Autonomous Mobile Robots", In *Proceedings of Third ACM/IEEE Int'l Conf. Human Robot Interaction (HRI '08)*, pp. 367-374, 2008.

Kozima, H. and Yano, H., "A robot that learns to communicate with human caregivers", In *Proceedings of the International Workshop on Epigenetic Robotics*, 2001.

Lafferty, John D.; Sleator, Daniel; and Temperly, Davy "Grammatical trigrams: a probabilistic link grammar." *AAAI Fall Symposium on Probabilistic Approaches to Natural Language.* 1992.

Lang, S., Kleinehagenbrock, M., Hohenner, S., Fritsch, J., Fink, G. A. and Sagerer, G., "Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot", In *Proceedings of International Conference on Multimodal Interfaces*, pp. 28-35. ACM, 2003.

Leese, S. J., Microphone arrays, In Davis, G. M., (editor), "Noise reduction in speech applications", *CRC Press*, 2002, Boca Raton, London, New York and Washington D. C., pp. 179-197.

Li D., Liu B., Maple C., Jiang D. and Yue Y., "Active robot learning for building up high-order beliefs", In *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, Jinan, China, August, 2008.

Li H. and Abe N., "Word clustering and disambiguation based on co-occurrence data", In *Proceedings of Computational Linguistics – Association for Computational Linguistics* 1998.

Mandel, C., Huebner, K. and Vierhuff, T., "Towards an Autonomous Wheelchair: Cognitive Aspects in Service Robotics", *In Towards Autonomous Robotic Systems (TAROS 2005), Proceedings,*2005.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A., "Building a Large Annotated Corpus of English: The Penn Treebank", *Computational Linguistics*, 1993, Vol.19 pp.313-330.

Matsuo, Y., Sakaki, T., Uchiyama, K. and Ishizuka. M., "Graph-based word clustering using web search engine", In *Proceedings of Empirical Methods in Natural Language Processing*, 2006.

McEnery, T. and Wilson. A., "Corpora and Translation: Uses and Future Prospects", *Technical Report from the Unit for Computer Research on the English Language (UCREL)*, University of Lancaster. March 1993.

Merlo, P. and Stevenson, S., "Automatic verb classification based on statistical distributions of argument structure", *Computational Linguistics*, Vol.27 No.3, pp.373-408, September 2001.

Mitchell P. Marcus, Mary Ann Marcinkiewicz., "Building a large annotated corpus of English: the Penn Treebank", *Journal Computational Linguistics - Special issue on using large corpora: II Volume 19 Issue 2*, June 1993

Mo Haijun, Huang Ping., "Planning of Grasping with Multifingered Hands Based on the Maximal External Wrench", *Journal of Mechanical Engineering* 2009-03

Oliver C. Schrempf, Uwe D. Hanebeck, Andreas J. Schmid., Heinz Wörn, A Novel Approach to Proactive Human-Robot Cooperation, Proceedings of the 2005 IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2005), pp. 555-560, Nashville, Tennessee, August, 2005.

Read, J., "Recongnising Affect in Text using Pointwise-Mutual Information", *Msc Thesis, University of Sussex*, September 2004.

Roy, N., Baltus, G., Fox, D., Gemperle, F., Goetz, J., Hirsch, T., Magaritis, D., Montemerlo, M., Pineau, J., Schulte, J., and Thrun. S., "Towards personal service robots for the elderly", In *Proceedings of the Workshop on Interactive Robotics and Entertainment (WIRE)*, Pittsburgh, PA, 2000. Carnegie Mellon University.

Santini, M., "Automatic identification of genre in Web pages", *PhD thesis, University of Bristol*, 2007.

Schrempf, O.C., Hanebeck, U.D., Schmid, A.J., and Wörn. H., "A Novel Approach to Proactive Human-Robot Cooperation", In *Proceedings of the 2005 IEEE International Workshop on Robot and Human Interactive Communication (ROMAN, 2005)*, pp. 555-560, Nashville, Tennessee, August 2005.

Stepan, G., et al., "ACROBOTER: a ceiling based crawling, hoisting and swinging service robot platform", In *Proceedings of BCS HCI 2009 Workshop*, September 1, 2009, Cambridge, UK.

Sun, L, Korhonen, A., et al "Verb Class Discovery from Rich Syntactic Data", *A. Gelbukh (Ed.): CICLing 2008, LNCS 4919, pp. 16–27, 2008.*

Turney, P.D., "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL", *Lecture Notes in Computer Science*, Springer-Verlag, 2001.

Wiemer-Hastings P., “How Latent is Latent Semantic Analysis?” *In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI’99)*, Stockholm, 1999.

Xue, Z., Kasper, A., Zoellner, J. M. and Dillmann, R., “An automatic grasp planning system for service robots”, In *Proceedings of 14th International Conference on Advanced Robotics (ICAR)*, 22-26 Jun. 2009.

http://www.ifr.org/

www.perl.org