

**ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA**

**Application of Case-based Reasoning for Amharic Legal Precedent Retrieval: A
Case Study with the Ethiopian Labor Law**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR
THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION SCIENCE**

**BY
ETHIOPIA TADESSE AKANA
July, 2002**

**ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA**

**APPLICATION OF CASE-BASED REASONING FOR AMHARIC LEGAL
PRECEDENT RETRIEVAL: A CASE STUDY WITH THE ETHIOPIAN LABOR
LAW**

**BY
ETHIOPIA TADESSE AKANA**

Signature of the Board of Examiners for Approval

DECLARATION

This thesis is my original work and has not been submitted as a partial requirement for a degree in any university

Ethiopia Tadesse Akana
July, 2002

The thesis has been submitted for examination with our approval as university advisors.

Ato Tesfaye Biru

W/t Saba Amsalu

DEDICATION

This thesis is dedicated to all I love

ACKNOWLEDGEMENT

I would like to thank many people for their kindness and assistance during my thesis writing. I sincerely thank my thesis advisors Ato Tesfaye Biru and W/t Saba Amsalu for their guidance and encouragement. Special thanks go to Ato Tesfaye whose brilliance inspires me in all that I do. I owe a lot for his patience, encouragement and heart-warming personality.

I truly appreciate the support of the people at the Federal Supreme Court of Ethiopia. Special thanks are also due to my friends and classmates who have one way or another contributed to the success of my thesis. I am also indebted to all staffs of SISA for their help and encouragement.

Most importantly, I thank my father and mother, who have given me great support and encouraged me throughout this difficult but exciting journey.

A heart felt thanks is also due to my sweetheart Menbere. I wouldn't have finished this thesis without his support and love.

Above all, I thank God who surrounded me with so many wonderful people.

Ethiopia Tadesse.

Table of Contents

CHAPTER ONE

INTRODUCTION	1
1.1 Background	1
1.2 Statement of the Problem and Justification of the Study	2
1.3 Objectives of the Study	8
1.3.1 General Objective	8
1.3.2 Specific Objectives	8
1.4 Methods.....	9
1.4.1 Review of Related Literature	9
1.4.2 Data Collection	9
1.4.3 Experimentation Method	10
1.4.4 Testing Techniques	10
1.5 Scope and Limitation of the Study.....	11
1.6 Application of Results and Beneficiaries.....	11
1.7 Organization of the Thesis	12

CHAPTER TWO

LITERATURE REVIEW	13
2.1 Introduction.....	13
2.2 Examples of CBR Applications.....	13
2.3 Case-Based Reasoning (CBR).....	16
2.4 Organization and Representation of Cases	19
2.4.1 Case Library.....	20
2.4.2 Case Structure	21
2.5 The Case-based Reasoning (CBR) Cycle	22
2.5.1 Case Retrieval and Similarity Assessment	23
2.5.2 Case Adaptation (Reuse).....	25
2.5.3 Case Revision.....	25
2.5.4 Case Retainment (Learning)	26
2.6 Case-based Reasoning in the Domain of Law	26

CHAPTER THREE

AMHARIC LEGAL PRECEDENTS	29
3.1 Introduction.....	29
3.2 Legal Precedents	29
3.2.1 Rule Comparison	31
3.2.2 Fact Comparison	32
3.3 The Federal Supreme Court of Ethiopia.....	33
3.4 The Amharic Writing System	35
3.4.1 The Amharic Characters	35
3.4.2 Punctuation Marks	36

3.4.3	Problems with the Amharic Writing System	36
3.4.3.1	Problems Regarding Consonants with Different form.....	36
3.4.3.2	Problems Related to Certain Interchangeably Used Orders	38
3.4.3.3	Problem regarding compound words	38
3.4.3.4	Problem regarding abbreviations	39
3.5	Amharic Software	39

CHAPTER FOUR

DESIGN OF THE PRECEDENT RETRIEVAL SYSTEM.....		42
4.1	Introduction.....	42
4.2	Selection of CBR Tool.....	43
4.3	The Legal Precedent Retrieval System	43
4.4	Setting up the Experiment.....	44
4.4.1	Domain Definition	44
4.4.2	Collection of Decided Cases	46
4.4.3	Feature Selection.....	46
	Feature.....	47
4.4.4	Concept Definition.....	47
4.4.5	Attribute Type Definition	49
4.4.5.1	Type Similarity Specification	50
4.4.6	Rule Definition.....	51
4.5	The Case-Base	53
4.5.1	The Case Representation.....	55
4.6	Case Retrieval	55
4.6.1	Case Similarity, Matching and Ranking	56
4.7	User Interface Development	57
4.8	Empirical Evaluation of the Prototype.....	60
4.8.1	Selection of Domain Experts	61
4.8.2	The Test Case.....	62
4.8.3	Evaluation with Recall and Precision	63
4.8.3.1	Testing with Attribute Similarity and Weight	64
4.8.3.2	Testing with Varying Similarity Rank Threshold.....	66
4.9	Analysis of Results	67

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS		68
5.1	Conclusions.....	68
5.2	Recommendations.....	71
REFERENCES		73
APPENDICES		77
	Appendix I	77
	Appendix II	78

List Of Tables

Table 3.1	Amharic Alphabets with Different Forms	37
Table 4.1	Selected Features for the Precedent Description	47
Table 4.2	Attribute types and the respective range	49
Table 4.3	The Performance Evaluation of the System by Domain Experts	62
Table 4.4	Case/Query Relevance Judgment	64
Table 4.5	Recall/ Precision Table	65

List of Figures

Figure 2.1	The CBR Cycle (adapted from Aamodt and Plaza, 1994).....	22
Figure 4.1	Concept Definition.....	48
Figure 4.2	The Defined Data Type and The Ranges for the Attribute 'gudayu'	50
Figure 4.3	Example of a similarity table for the case type 'ygudayu aynte'	51
Figure 4.4	A Sample Completion Rule.....	53
Figure 4.6	Nearest Neighbor Matching Algorithm.....	56
Figure 4.7	User interface window for online query (attribute: value).....	58
Figure 4.8	User interface window for online query (question: answer) wizard	59
Figure 4.9	Query Result Window.....	59
Figure 4.10	User interface Window for Case Entry into the Case Base.....	60
Figure 4.11	Precision/ Recall Graph.....	66

LIST OF ABBREVIATIONS

AI - Artificial Intelligence

ASCII -American Standard Code for Information Interchange

CBR -Case-based Reasoning

CMS -Case Management System

CRS -Court Record System

CQL -Case Query Language

FSCE -Federal Supreme Court of Ethiopia

GREBE -Generator of Recursive Exemplar-Based Explanation

HTTP -Hyper Text Transfer Protocol

NN - Nearest Neighbor

URL – Uniform Resource Locator

ODBC- Open Database Connection

ABSTRACT

This Research is concerned with the development of a Case-Based Reasoning (CBR) based precedent retrieval system in the domain of Ethiopian Labor Law. The requirement for the system was to build a knowledge base in which complete decided cases could be entered and then recalled when similar cases arose again.

Standard case representation to the original knowledge source (legal cases) has been used to store legal cases. Legal cases have a predefined case structure with a number of features. The features are extracted to reflect the important aspects of a legal case. Given a new case, the feature values are used to do the search for a similar case from the case-base.

Content based matching mechanism is used in the retrieval process. Content based matching matches the equivalent parts of the target and the source cases and calculates the degree of similarity according to the number of features matched, and feature weights.

To increase the retrieval effectiveness, a mechanism for feature importance value (weight) assignment was required. The approach adopted takes into account domain experts' opinions to assign weights to the features.

A Case-Based Reasoning prototype has been implemented by using the CBR-Works toolkit. To facilitate the insertion of additional cases and searching, an online interface has also been included.

CHAPTER ONE

INTRODUCTION

1.1 Background

As stated by Yang et al. (1994), legal and regulatory reasoning is one of artificial intelligence¹ application domains where substantial amounts of research has been done over the years (e.g., FLEXICON (Fast Legal Expert Information Consultant) By Gelbart and Smith, 1990; GREBE (Generator of Recursive Exemplar-Based Explanation) by Branting and Porter, 1993). The law contains legal rules, which state what should be done in certain circumstances in the context of a legal system. Most of the legal rules are expressed in the form of statutes² and precedents³.

Since it is normal to view statutory regulations as legal rules, it is also natural to expect these to be most easily expressed formally in terms of decision rules (Ashley, 1990). However, as rules are ill defined and incomplete, it is becoming increasingly common to make the regulations established through precedent. In this type of system, the legal rules do not in themselves allow any decision to be made. Much greater importance is attached to the means by which precedent is accessed and interpreted (Osborn et al., 1999). This research is an attempt to develop a precedent retrieval system in the domain of Ethiopian Labor Law.

¹ Artificial intelligence is the mimicking of human thought and cognitive processes to solve complex problems automatically.

² Statute is a law that declares, commands, or prohibits something, written by the legislature body.

³ Precedent is simply a prior decision that can be used as a standard in a later similar case.

Labor law has the generic objective of safeguarding employees and employers in the work environment. Such objectives express government policies and the principles which bring labor regulations into existence. The policies or principles are translated into more detailed issues in such subject areas as: remuneration, suspension, termination, reduction of workers, severance pay and compensation, etc. These issues or requirements are very general and the expressions are highly abstract. The argument surrounding each appeal decision is often complex, involving the balancing of a variety of assertions by the parties concerned. Usually, the experts make a decision after consulting all information relevant to the given case. In addition to statute and case histories relevant to a given case, experts refer to all information available to provide justification for their decision.

The problem is therefore; to support the consultation of these loosely expressed legal requirements without being forced to make explicit all the information needed to arrive at a decision. Building a knowledge base in which complete decided cases could be entered and then recalled or reused when similar problems arose again could be one way of addressing the problem.

1.2 Statement of the Problem and Justification of the Study

The purpose of legal research is to help a client solve a legal dispute, prevent such a dispute from arising, or prevent the dispute from getting worse. All three are

accomplished by analyzing the facts of the client's case and by applying any existing mandatory primary authority⁴ to these facts (Statsky, 1998).

In general, a lawyer searching for solutions to a problem must dig through a wealth of case law and doctrinal writings to find relevant guidance on the problem case at hand. The overriding goal for the lawyer in this context is to get potentially relevant legal resources (decided cases and doctrinal writings) that can help him/her better understand the legal issues he/she is dealing with and how he/she might go about tackling the legal problems at hand.

Once the lawyer has completed the laborious background research work, he/she will use the results to begin the legal reasoning process i.e. that of forming an argument for his/her client and in doing so will inevitably need to revisit these research tasks. Searching through this document corpus manually in a law library can be a very time consuming process (De Mulder and Combrink-Kuiters, 1996). The efficiency of the lawyer in this context could be greatly improved if an information retrieval system could boast, at the very least, the following functionality: (Curran and Higgins, 2000,)

- An interface designed with the guidance of a legal expert, which 'walks' the lawyer through the various possible issues in the case, helping the lawyer build up a profile of his case at hand.

⁴ Primary authority is any law that the court can rely on in reaching its decision including other court opinions, administrative regulations, statutes, constitutional provisions etc.

- Once a basic profile is built up the system should indicate (through a process of basic pattern matching) those important cases in the field which best match the profile of the current problem i.e., cases, which are most likely to discuss the kind of issues that the lawyers' case involves.
- If the system does indicate which cases best 'match' the input case, it should indicate how this match occurs and also indicate how the retrieved cases are distinguishable from the current case.

As to the role of computer based systems in the domain of law from the judges perspective Schjolberg (2002), Stated:

We are used to problem solving in our discretionary judgments. We do not need support systems making the final decision, but systems with updated and comprehensive information on previous similar court decisions applied on the facts in the individual current case brought before us. The judge should only be assisted in the decision-making, leaving the judge to choose the proposed solution or reject it.

More radical approaches to improving retrieval seek to utilize techniques related to artificial intelligence and legal expert systems⁵. They seek to represent legal documents in terms of the legal concepts for which they may be significant (Stefik 1995).

This research used the case-based approach to develop a legal precedent retrieval system. At this point it is appropriate to address the issue of why case-based reasoning is preferred rather than rule based legal decision support system.

⁵ Expert system encapsulates the specialist knowledge gained from human expert (such as a lawyer) and applies that knowledge to make decisions automatically.

Legal rules are the unit of knowledge to be represented in the knowledge base. Legal rules are requirements, which describe a required design solution in certain circumstances. Given this description of legal rules, conventional if-then rules seem a natural choice for representing legal rules in the form of “**if** circumstances **then** solution”.

According to Yang et al. (1994), legal rules used in different systems are formal and prescriptive. However, as the regulations are becoming less prescriptive and more emphasis is put onto the interpretive use of the regulations. This presents several difficulties in employing a rule based decision support system approach: (Ibid)

1. Legal rules are described at different levels of detail
2. There exists a terminological gap between legal rules and cases
3. The meaning of terms is often in debate
4. New legal rules are constantly emerging
5. Legal rules are the generalization of the rulings made in case histories

In this regard, Racine (1996) noted that to solve a problem, a reasoner recalls previous situations similar to the current one and use it to interpret the current situation. In many practical application domains, this technique is more effective in solving problems than rule-based expert system approaches, since it can overcome the knowledge acquisition bottleneck by storing entire cases for later analysis, rather than asking the domain experts to extract their knowledge in the forms of rule like languages.

In CBR, as opposed to rule-base approaches, knowledge about the domain is acquired and maintained through similar cases and does not need a domain expert or knowledge about the problem domain as such. The generic concept of this research hence moves

away from a rule-based implementation, as for each legal case, specific rule sets would need to be encoded.

According to Curran and Higgins (2000), of more interest here, therefore are those systems which adopt a more realist view of the law and place emphasis on recognizing that an important component of legal reasoning is identifying from decided case-law precedents for decisions in a particular case. Case-based reasoning (CBR) basically involves reasoning from collected examples of previous problem solving experiences, these experiences in this field being actual legal decisions.

All case-based reasoning methods have in common the following process: (Harrison 1997):

- Retrieve the most similar case (or cases) comparing the case to the library of past cases;
- Reuse the retrieved case to try to solve the current problem;
- Revise and adapt the proposed solution if necessary;
- Retain the final solution as part of a new case.

The most basic problems in CBR as emphasized by De Mantaras and Plaza (1994), are the retrieval and selection of cases, since the remaining operation of adaptation and evaluation will succeed only if past cases are the relevant ones.

Retrieving a case starts with a (possibly partial) problem description and ends when a best matching case is found. This subtasks involves: (Harrison 1997)

- Identifying a set of relevant problem descriptors;
- Matching the case and returning a set of sufficiently similar cases (given a similarity threshold of some kind); and
- Selecting the best case from the set of cases returned

This research, therefore attempted to develop a precedent retrieval system in the domain of Ethiopian Labor Law by using the CBR methodology.

On average, the Federal Supreme Court of Ethiopia handles 2,000 legal case files every year. Since 1975, the court maintained the files manually for twenty-three years. However, as the number of legal documents grew over the years, manual file maintenance became more cumbersome and time taking for retrieval. As a result it becomes difficult to render adequate services to users. To solve this problem, in 1998 the Supreme Court implemented an automated system known as CRS (Court Record System). CRS is a home-developed software geared for handling Amharic legal cases. It is designed and implemented on a Microsoft Access based back end database system. The software has two parts- CRS-MAIN and CRS-REPORT. CRS-MAIN helps to maintain a database of legal cases, while CRS-REPORT as the name implies enables generation of different statistical reports. The system is currently used to give effective access to the paper form legal cases, which are further organized by color codes. The implementation of the system has helped the court to store case files (active and disposed), generate court

lists on a daily bases, basic management reports as well as producing several statistical reports.

However, the existing system has the following limitations:

- It lacks the capacity to search for specific Judgments or any other document classified by case type when law practitioners or researchers need them for reference.
- The upshot is that lawyers wishing to make effective use of the database must overcome a ‘conceptual gap’- translating the information need they have in their head in legal terms into a query, which should be put in technical database terms thus distorting the semantics of their request.

1.3 Objectives of the Study

1.3.1 General Objective

The general objective of the study is to explore the potential application of case-based reasoning to develop precedent retrieval system for Amharic legal cases.

1.3.2 Specific Objectives

With the aim of achieving the general objective of this study, the following specific objectives were drawn:

- To collect decided legal cases
- Represent legal cases, including selection of representative features.
- Organize the case base and retrieve cases that are potentially relevant to a new case (query)

- Assign similarity rank for the retrieved cases to decide which are the most relevant to the new case
- Incorporate general knowledge that can facilitate query formulation and addition of cases to the case base
- Testing the prototype system
- Draw conclusions and forward recommendations based on the research outcome.

1.4 Methods

The following methods were employed in order to achieve the above stated objectives.

1.4.1 Review of Related Literature

Literature review was made to get further understanding of case-based reasoning and its application and the structure and properties of Amharic legal cases. For this purpose, reference was made to books, journal articles, legal cases and materials on the Internet.

1.4.2 Data Collection

In order to get the required information for the research and comments at the different stages of the experimentation and evaluation, discussions and interviews were conducted with domain experts at the Federal Supreme Court of Ethiopia (FSCE). In addition, review of legal cases was made to gather information on the content and structure of legal cases. Since the analysis and extraction of important features was time consuming, a total of 39 labor cases were taken for this research.

1.4.3 Experimentation Method

In Case-Based Reasoning (CBR), the primary knowledge source is a memory of stored cases recording specific prior episodes. Accordingly, sample cases in the domain of labor law were taken from the Federal Supreme Court of Ethiopia (FSCE). There were two reasons for selecting FSCE as a source of data: the first being that, the court has a database of cases, which could facilitate the CBR-system development process. The second reason is, as the main objective of this research is to assist legal professionals in precedent retrieval, precedents from high courts have more acceptance than lower level courts.

Various tools that manage basic parts of the CBR process are available. For this research, CBR-Works is used. The focus of this research was retrieval of relevant decided legal case(s) similar to the case at hand. For this purpose, the method suggested by Harrison (1997) (page 7 of this report) was adopted.

1.4.4 Testing Techniques

The suggested CBR based legal precedent retrieval system has been experimented by developing a prototype. Then actual users tested the prototype. A reasonably representative testing set (legal queries) has been taken from the Supreme Court. Then the performance of the system was evaluated by taking the following parameters:

- The users feedback, regarding:
 - The relevance of the information retrieved (how suitable the precedent is to the case at hand)
 - The relevance of the fields (Attributes) in the case structure

- The ease of use of the system, in terms of the entry forms layout and level of detail

In addition to this, the standard measures of relevance in the information retrieval community- precision and recall have been used to evaluate the performance of the prototype system. For the implementation of this test, different experimental groups with varying definition of the expert knowledge based similarity of attributes and attribute weights have been used.

1.5 Scope and Limitation of the Study

The scope of this research is limited to the retrieval of the most similar cases comparing the case at hand to the library of past decided-cases, using the CBR methodology. It does not involve construction of precedent based legal arguments. The cases considered in this research are those verdicts given by the Federal Supreme Court of Ethiopia in the domain of labor law. Due to the limited time available, it is only the verdict of the cases considered for representation in the case base, rather than all the hearings and justifications.

[

1.6 Application of Results and Beneficiaries

As an academic exercise, this research will serve to fulfill the requirement of the program the researcher is enrolled in. In addition, the study is also important for the following purposes:

- The Federal Supreme Court of Ethiopia can apply the output of this research so as to overcome the limitations of the existing database system and render effective information services to its users.

- As the system helps in the retrieval of legal precedents, by studying previous similar to their own, legal agents can gather elements to guide their actions; judges can make consistent rulings, and lawyers can seek arguments that favor their case.
- The research output will help lawyers to render adequate and timely advisory services to their clients.
- This system accumulates case histories of interpretation of regulations used to establish precedents. These precedents can be used when revising statutory regulation and enrich the resulting new versions of regulations.
- As provision of efficient legal information services support proper conduct of justice, the general public will in a way be the beneficiary of this research result.
- This research can serve as a base for future researchers in the area.

1.7 Organization of the Thesis

This thesis is organized into the following chapters: Chapter Two gives an overview of related research work and background introduction about case-based reasoning. Chapter three focuses on the domain legal precedents and Amharic writing system. Chapter Four presents the approach used by this study in using case-based reasoning methodology to address Amharic legal precedent retrieval in the domain of Ethiopian labor law and discusses the online interface developed for the system. Chapter Five gives final conclusions and forward recommendations for future studies.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The previous chapter discussed the problem addressed by this study and the approach used. This chapter provides a review of related systems and the lessons learnt from such systems. In addition, the case-based technique and its functionalities are described in the subsections of this chapter.

2.2 Examples of CBR Applications

This section describes some related works on the application of CBR in the domain of law. One of the earliest AI systems to represent legal cases, the Legal Research System (LRS) was a retrieval system for cases in negotiable instruments law (Hafner, 1981). LRS used a frame like slot-filler structure to encode the main elements of cases: Plaintiff, Defendant, Cause of Action, Facts, Hypotheticals, Legal-Criteria, Legal-Effects, and Holdings. The fillers of the slots were symbolic descriptions of legal concepts: those concepts that appeared in the last three slots comprised the legal issues of the case. The justification (or *ratio decidendi*) of the decision given by the court was encoded as part of the Holdings slot. The LRS retrieved cases with matching facts, issues, and *ratios* based on queries rather than a description of a new case; it did not perform similarity ranking, nor did it construct legal arguments.

The first complete case-based legal reasoning model was embodied in HYPO (Rissland and Ashley 1987, Ashley 1990)⁶, a computer program for analyzing cases and constructing legal arguments in the domain of trade secrets law. HYPO used a multi-level description of cases: *facts* and *factual predicates* represented the underlying circumstances of a case, and a library of *dimensions* (selected by the system's designers, based on a survey of relevant legal authorities) was used to retrieve cases.

A dimension in the HYPO model is a domain specific factor, representing a fact pattern that influences the outcome of cases in the domain (Rissland et. al, 1984). Dimensions form an intermediate level of abstraction, between the facts of a case (“defendant hired an employee who previously worked for plaintiff”), and the top-level legal issues (“defendant misappropriated plaintiff’s trade secrets”) whose resolution determines which side wins. Dimensions in HYPO can be binary, multi-valued, or quantitative; for example, in the trade secrets domain a binary factor is **Generally-Known**, indicating that the alleged secret information is generally known within the industry, and a quantitative factor is **Secrets-Voluntarily-Disclosed**, representing the number of people to whom the plaintiff told the information.

Structurally HYPO consists of a knowledge base of thirty cases. HYPO’s legal domain knowledge includes, for each dimension, which side of a dispute would be favored by the factor’s having particular values (e.g., *a true* value for binary dimensions, or a larger value for quantitative dimension), and how the factor’s value is related to a case’s

⁶ HYPO evolved over the time period of its development (approximately 1984-1990); the description below and other references to the HYPO model in this study are based on the doctoral thesis of Kevin Ashley (1990).

underlying facts. For example, in the trade secrets domain, a *true* value for **Generally Known** favors the defendant; a *false* value favors the plaintiff.

When a new case is presented to HYPO, the dimensions applicable to the new case are computed. Then, HYPO constructs a “claim lattice” – a similarity network that has the new case as its root node, and prior cases that share a maximal set of dimensions with the new case as its immediate successors. These immediate successors are called the *most-on-point cases*. The claim lattice would be further populated with other on-point cases (those sharing at least one factor-value with the new case), ordered by the subset relation (i.e., if the set of shared dimensions of one prior case properly includes the set of shared dimensions for another prior case, then the first case is more “on point” and is ranked higher than the second case.)

GREBE (Generator of Recursive Exemplar-Based Explanation) is a system developed for classification of legal cases. It uses knowledge in form of generalizations and category exemplars to determine the classification of new cases (Branting and Porter, 1993). GREBE’s knowledge base contains rules and a small collection of exemplar cases concerned with Texas’ laws for injuries to workers traveling outside of the work place. GREBE uses a semantic network representation for cases and uses generalization and exemplar-based explanation. GREBE represents an advance over previous exemplar-based systems in that it accepts detailed semantic network representation for cases, retains and reuses the explanations of category exemplars and use exemplar based reasoning recursively to assess similarity.

For the experimental evaluation, GREBE's analysis of 18 worker's compensation cases was compared to the analysis of the same cases by law students. The purpose of the evaluation was primarily to assess the overall effectiveness of GREBE in addressing the task. But a secondary purpose was to demonstrate that analysis of worker's compensation cases is a challenging task, even for humans with legal training.

The memoranda produced by the students, together with GREBE's analysis of the same 18 hypothetical cases were then given to a domain expert, an attorney who is a recognized authority on Texas worker's compensation law. The domain expert was asked to grade all of the analyses.

The evaluation is tentative for two reasons. First, the accuracy of the evaluation is uncertain because it rests upon the judgment of a single domain expert. This evaluation approach is used for the current research by involving a number of domain experts. It is believed that the increase in number of domain experts will make the evaluation more objective. A second reason that the evaluation is tentative is that it fails to isolate the contributions of the various components of the GREBE system. The next section describes the CBR technique and its functionalities.

2.3 Case-Based Reasoning (CBR)

Case-based reasoning relates to a reasoning process based on recalling a related previous experience (a memory of stored cases recording specific prior episodes) rather than reasoning based on generalized rules (Leake, 1996). According to Aamodt and Plaza (1994), in CBR terminology, a case usually denotes a problem (situation). A previously

experienced situation, which has been captured and stored in a way that it can be reused in solving future problems, is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem (situation).

Case-based Reasoning (CBR) combines the knowledge based support philosophy with a simulation of human reasoning when past experience is used, i.e. mentally searching for similar situations happened in the past and reusing the experience gained in those situations. In CBR, the knowledge cases are structured and stored in a database, which the user queries when trying to solve a problem. In processing a query, the system evaluates the similarity of features between each case in the database and the query. The most similar case(s) are presented to the user as possible scenarios for the problem, i.e. the system doesn't make the decision it only supports the decision making process (Haque, et al, 1999).

In other words, in CBR new solutions are generated by retrieving the most relevant cases from memory and adapting them to fit new situations i.e., reasoning by analogy (based on remembering). In fact, much of the original inspiration for the CBR approach came from the role of reminding in human reasoning (Schank, 1982). Case-based reasoning therefore, refers to a style of designing a system so that thought and reaction in a situation are guided by a distinctive prior case (precedent, prototype, exemplar, or episode).

The CBR approach as stated by Leake (1996), is based on two tenets about the nature of the world.

- The first tenet is that the world is regular i.e., similar problems have similar solutions. Consequently, solutions for similar prior problems are a useful starting point for new problem solving.
- The second tenet is that the types of problems we encounter tend to recur. Consequently, future problems are likely to be similar to current problems.

When the two tenets hold, it is worthwhile to remember and reuse current reasoning.

Case-based reasoning systems are classified as Interpretative and problem solving (Jurisica, 1995).

- **Interpretative systems** compare new situation to recalled experience. The interpretation is used when the problem is not well understood and when there is a need to criticize a solution. Cases are used to provide justification for solutions, evaluation of solutions when no clear-cut methods are available, and interpretation of situations when definitions of the situation's boundaries are not well defined.
- **Problem-solving systems** derive solutions to a new problems by adapting an old solution for the current situation.

As stated by Williamson (1996), the main distinction between interpretative and problem-solving case-based reasoning systems is that while the former group of systems does not

use adaptation⁷ at all the latter group relies heavily on adaptation. This research is interpretative type.

According to Harrison (1997), the following are among some of the characteristics of a domain that indicate the suitability of a CBR approach:

- Records of previously solved problems exist
- Historical cases are viewed as an asset which ought to be preserved
- Remembering previous experiences is useful
- Specialists talk about their domain by giving examples
- Experience is at least as valuable as textbook knowledge

Accordingly, law makes an excellent domain for studying case-based reasoning (Ashley, 1990). Expert system⁸ designers use case-based reasoning to capture expertise in domains where rules are ill defined, incomplete, or inconsistent (Bench-Capon, 2001). As an indispensable supplement to reasoning deductively with legal rules, attorneys and judges reason analogically with precedent cases; rule predicates are simply not sufficiently well defined for them to infer correct decisions deductively. Legal experts make competing arguments instead, pitting conflicting interpretations of cases and facts against each other (Gelbart and Smith, 1991).

2.4 Organization and Representation of Cases

The reasoning by analogy of CBR is based in collecting a lot of relevant cases or experiences in a particular domain (A-TEAM, 2001). A case is a conceptualized piece of

⁷ Adaptation refers to reusing the information and knowledge in a previous case to solve new problem. Please refer section 2.3.2 of this thesis for further information.

⁸ Expert system a problem- a type of application program that makes decisions or solves problems in a particular field by using knowledge and analytical rules defined by experts in the field.

knowledge representing experience. It contains the past lesson that is the content of the case and the context in which the lesson can be used (Watson and Marir, 1994). The set of stored cases or experiences is usually named as the Case Library or the Case Base or the Case Memory (A-TEAM, 2001, Leake 1996, Aamodt and Plaza, 1994). A Case-based reasoner is heavily dependent on the structure and content of its collection of cases (Aamodt and Plaza, 1994).

2.4.1 Case Library

Main memory organizations, in CBR systems, can be summarized in two general approaches (A-TEAM, 2001): flat memories and hierarchical memories. Flat memories always retrieve the set of cases best matching the input case. Moreover, adding new cases to memory is cheap in time. But they have a major disadvantage: the retrieval time is very expensive since every case in the memory is matched against the current case (Watson, 1996).

On the other hand, there are the hierarchical memories. In such kind of memories, matching process and retrieval time are more efficient, due to the fact that only few cases are considered for similarity assessment purposes, after a prior discriminating search in the hierarchical structure. Anyway, they also have some disadvantages. Keeping the hierarchical structure in optimal conditions requires an overhead on the case library organization, and the retrieval process could miss some optimal cases searching a wrong area of the hierarchical memory (Harrison, 1997).

Case-representation is important because the representation will determine what data can be stored, how many cases can be stored, how similarity can be assessed, and crucially for large cases bases, how quickly cases can be retrieved (Aha, 1991).

The representation problem in CBR is primarily the problem of deciding what to store in a case, finding an appropriate structure for describing case contents, and deciding how the case memory should be organized for effective retrieval and reuse (Aamodt and Plaza, 1994). Cases can be represented in a variety of forms using the full range of AI representational formalisms including frames, objects, predicates, and semantic nets (Watson and Marir, 1994).

2.4.2 Case Structure

The cases stored in the case library are past experiences, which have been captured and learned in such a way that they can be reused to solve future causalities. A case does incorporate a set of features such as: (A-TEAM, 2001)

- An identifier of the case
- The description of the case
- The diagnostic of the case
- The solution of the case
- Other relevant information about the case

2.5 The Case-based Reasoning (CBR) Cycle

Case-based reasoning is in effect a cyclic and integrated process of solving a problem (Aamodt and Plaza, 1994). As depicted in figure 2.1, Aamodt and plaza, (1994) have described CBR typically as a cyclical process comprising the four REs:

1. REtrieve the most similar case(s)
2. REuse the information and knowledge in that case to solve the problem
3. REvise the proposed solution if necessary, and
4. REtain the new solution if it is likely to be useful for future problem solving

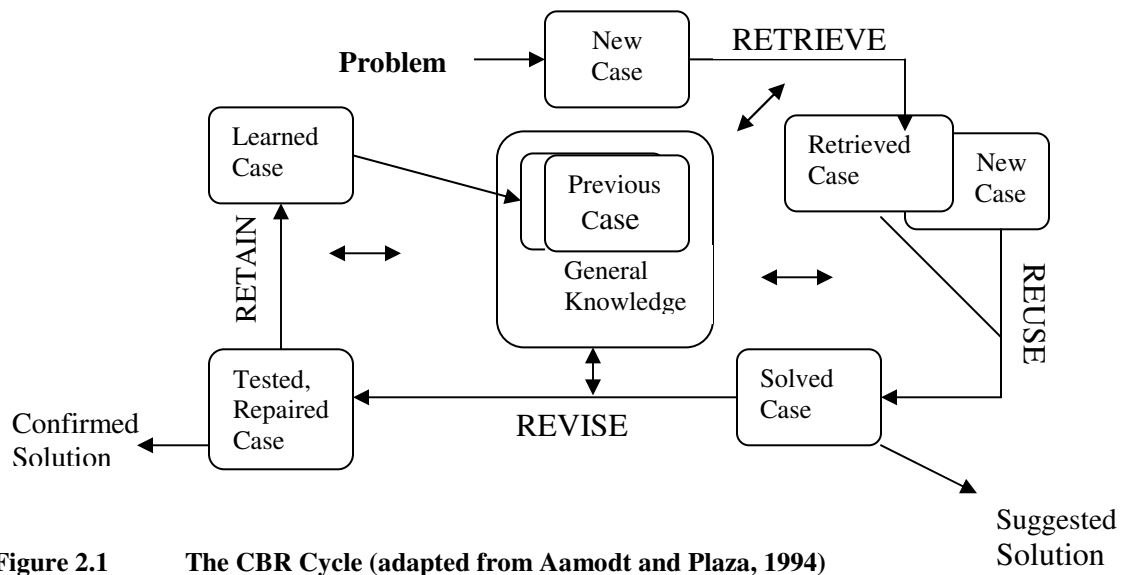


Figure 2.1 The CBR Cycle (adapted from Aamodt and Plaza, 1994)

As shown (top of figure 2.1), an initial description of a problem defines a new case. This new case is used to RETRIEVE a case from the collection of previous cases. The retrieved case is combined with the new case through REUSE into a solved case, i.e. a proposed solution to the initial problem. Through the REVISE process this solution is tested for success, e.g. through application to the real world environment or evaluation by an expert, and repaired if fails. During RETAIN, useful experience is retained for future reuse, and the case base is updated by a new learned case, or by modification of some

existing cases. General knowledge⁹, as indicated in the figure, usually plays a part in this cycle, by supporting the CBR processes. The following section further outline how each process in the cycle can be handled.

2.5.1 Case Retrieval and Similarity Assessment

The task of retrieving cases in the case library is slightly more difficult than typical retrieval in structured databases (A-TEAM, 2001). In database systems, the recalling algorithms use an exact matching method, whereas in case library retrieval, because of the very nature of the structure, a partial matching strategy should be used. A retrieval method should try to maximize the similarity between the actual case and the retrieved one(s). And the task implies most of the time the use of general domain knowledge.

The retrieving process of a case (or a set of cases) from the system's memory strongly depends on the case library organization. Major case library structures as described in section 2.4.1 are flat memories or hierarchical ones. Flat memories have an intrinsic problem of bad performance in time, so that the retrieval time is proportional to the size of the case library (A-TEAM, 2001). Hierarchical memories are very effective in time retrieval because only a few cases are considered for similarity assessment purposes, after a prior discriminating search in the hierarchical structure (Watson, 1996). Also some times such systems could not reach optimal cases because of exploring a wrong area of the hierarchy.

⁹ General knowledge refers to general domain dependent knowledge, as opposed to specific knowledge embodied by cases.

According to Bench-Capon (2001), the retrieval process in case libraries, usually consists of two main sub steps:

- **Searching the most similar cases to the new case:** the goal of this stage is recalling the most promising cases based on using some direct or derived features of the new case into the case library (given that the subsystem has a goal and therefore the relevance of the cases depends upon that goal).
- **Selecting the best case(s):** the best case(s) among those ones collected in the previous step are selected. Commonly, this selection is made by means of a case ranking process through a similarity or distance function. The best-retrieved case is the closest one (most similar) to the new case.

Selecting the best similar case(s) is usually performed in most Case-based reasoning systems by means of some evaluation heuristic functions or distances, which are possibly domain dependent. They are usually named as Nearest Neighbor (NN) algorithms (Watson, 1996). The evaluation function usually combines all the partial matching through a dimension or attribute of the cases into an aggregate or full-dimensional partial matching between the searched cases and the new case. Commonly, each attribute or dimension of a case has a determined importance value (weight), which is incorporated in the evaluation function.

Most Case-based reasoners such as REMIND (Cognitive, 1992), MEDIATOR (Kolodner and Simpson, 1989), PERSUADER (Sycara, 1987) etc., use a generalized weighted distance function (NN) such as,

$$\mathbf{Dist} (C_i, C_j) = \sum_{K=1}^n w_k \times \mathbf{atr_dist} (C_{ik}, C_{jk})$$

Some others measures have also been defined in the literature, (Leake et al. 1997, Osborne and Bridge, 1998, Sanchez-Marre et al, 1998).

2.5.2 Case Adaptation (Reuse)

When the best partial-matching case selected from the case library does not match perfectly with the new case, the old solution needs to be adapted to fit more accurately the new case solution. This reusing process can happen during the solution formulation (adaptation), or after some feedback has pointed out some problem in the evaluation step, which needs to be fixed (repair).

There are lots of strategies that have been used by Case-based reasoners. All these techniques can be grouped as null adaptation, structural adaptation and derivational adaptation, although in most Case-based reasoners, several mixture kinds of adaptation methods are implemented (Kolodner, 1993; Riesbeck and Schank, 1989).

2.5.3 Case Revision

When a case solution generated by the reuse phase is not correct, an opportunity for learning from failure arises (Aamdot and Plaza, 1994). This phase is called case revision and consists of two tasks:

1. Evaluate the case solution generated by reuse. If successful, learning from the success (case retainment)
2. Otherwise repair the case solution using domain-specific knowledge

According to Aamdot and Plaza (1994), the evaluation task takes the result from applying the solution in the real environment (asking a teacher or performing the task in the real world). This is usually a step outside the CBR system, since it at least for a system in normal operation involves the application of a suggested solution to the real problem.

Case repair involves detecting the errors of the current solution and generating explanations for them.

2.5.4 Case Retainment (Learning)

This is the process of incorporating what is useful to retain from the new problem-solving episode into the existing knowledge. The learning from success or failure of the proposed solution is triggered by the outcome of the evaluation and possible repair. It involves selecting which information from the case to retain, in what form to retain it, how to store the case for later retrieval for similar problems, and how to integrate the new case in the memory structure (Aamdot and Plaza, 1994).

2.6 Case-based Reasoning in the Domain of Law

An important task of law practitioners is to find suitable cases, which can be referred to in court to make their argumentation strong. Before starting a search for suitable case reports, practitioners have a legal issue and their client's factual situation in mind. During the search they try to find cases related to a particular issue which has been decided under similar circumstances. In addition to this they also consider some other factors including whether they are arguing for or against the result indicated by legal rules, whether the point at the issue has been decided in ones favor or not in a precedent, and whether there

are alternative paths to desired result. Retrieval of suitable cases for these purposes is an important issue in computer based legal research (Gelbart and Smith, 1991). Two well cited difficulties are query formation and irrelevant retrieval.

The need is therefore, for systems which allow users to see the details of the primary sources (Pasha and Soper, 1999). Legal professionals, for example, need a system, which, on the one hand, analyses their request, compares it with the available data items and retrieves only those data items, which are relevant to their request.

Legal reasoning as stated by Bench-Capon (2001), is seen as relating the current decisions to past decisions, and arguing that some past decision should govern the present case. As literature reveals, automation is particularly important in legal research, the essence of which is reasoning based on relevant decided cases and related legal information.

According to Gelbart and Smith (1992), A Case-based reasoning legal expert system generates an opinion about a hypothetical fact pattern on the bases of relevantly similar decided cases stored in its database. Accordingly, during a consultation with a Case-based system, a profile of the relevant facts of the hypothetical case is formulated and used as a search query to retrieve matching cases from the case library of legal decisions. Information stored in those matching cases forms the basis for the system's prediction of the outcome of the user's hypothetical case. The system assigns higher weights to information retrieved from more significant judgments such as recent cases, higher-level courts, and applicable jurisdictions.

Case-based advisory systems have many advantages for legal professionals. Among which Gelbart and Smith (1992) noted the following:

- They provide information which the lawyer must acquire by examining a large number of legal documents, in a manner analogous to the reasoning process of a legal professional who has studied the relevant case law
- Changes in the law are automatically reflected in new cases added to the database.

CHAPTER THREE

AMHARIC LEGAL PRECEDENTS

3.1 Introduction

As the main objective of this study is developing Amharic legal precedent retrieval system, describing the role of legal precedents with particular local emphasis is appropriate. This chapter therefore, highlights, the role of legal precedents, structure of Amharic legal cases and provides an overview of the Amharic language writing system and software.

3.2 Legal Precedents

The function of a precedent (court's opinion) is to apply one or more rules of law to the facts involved in the legal dispute before the court. According to Statsky (1998), there are two broad categories of rules of law that a court interpret and apply:

- Enacted law
- Common law

Enacted Law is any law that is not created within litigation. Here is a list of the major categories of enacted laws and their definitions:

- **Constitution:** The fundamental law that creates the branches of government and identifies based rights and liberties. The legislature often writes the constitution, sometimes with a vote of the people.
- **Statute:** A law that declares, commands, or prohibits something. The legislature writes statutes.

- **Administrative regulation:** A law designed to explain or carry out the statutes and executive orders that govern an administrative agency. The agency writes its own administrative regulations.
- **Ordinance:** A law that declares, commands, or prohibits something. It is written by the local legislature such as the city council.
- **Rule of Court:** The procedural laws that govern the mechanics of litigation before a court. They are written by the legislature and/or by the highest court of the jurisdiction.

Common law is judge-made law created by the courts to resolve a dispute within a particular litigation in the absence of controlling enacted law such as a statute or constitutional provision that governs the dispute (Daniel and Rissland, 1999). When a court has a dispute before it in litigation, it seeks to resolve the dispute. For example most of the law of negligence was initially created as common law by the courts because the legislatures had not created (enacted) any statutes in this area of the law.

According to Costa et al. (2001), one of the important aspects of law is its open texture that sometimes leads to conflicting interpretations of legal norms. The most common way of resolving these conflicts is to invoke past interpretations, especially if stated by hierarchically superior courts; given their privileged position, it is presumed that the jurisprudence established by them in their decisions is adopted in future similar cases. Therefore, by studying previous cases similar to their own, legal agents can gather elements to guide their actions, judges can make consistent rulings, and lawyers can seek arguments that favor their cases.

Assume that you are in a law library researching the facts of a client's problem. You find an opinion that looks promising, but you need to analyze it carefully before concluding that it applies to the facts you are researching. The opinion reached a certain result, called a holding. The conclusion of your legal analysis will be your assessment of whether this holding applies to the facts of your research problem. How do you make this assessment? You go through two separate comparisons:

- First, you compare the rule of law (enacted law or common law) that was interpreted and applied in the opinion with the rule of law (enacted law or common law) that you have uncovered elsewhere in your research as potentially applicable to the facts of your research problem
- Second, you compare the key facts of the opinion (i.e., those that were essential or very important to its holding) with the facts of your research problem.

3.2.1 Rule Comparison

If the client has a negligence case, for example, you search for opinions that interpret the law of negligence. Rule comparison in the analysis of opinions, therefore, is fairly simple. The general principle is: you compare the rule involved in your client's case with the rule interpreted and applied in the opinion, and you proceed only if the rule is exactly the same.

3.2.2 Fact Comparison

Here is the heart of the analysis. Before the holding of an opinion can apply, it is necessary to demonstrate that the key facts of the opinion are substantially the same as the facts in the client's case. If the facts are exactly the same or almost exactly the same the opinion is said to be on all fours with the facts at hand. If so, then you will have little difficulty convincing someone that the holding of the opinion applies to the facts. It is rare, however, to find an opinion on all fours. Consequently, careful analysis of factual similarities and differences must be made. In general, if the facts are substantially similar, the ruling applies; if they are substantially different, it does not.

Key facts in the opinion need to be determined, because these facts alone are the bases of the comparison. As indicated earlier, a key fact is a fact that was essential or very important to the holding of the court. In a divorce opinion, for example, it will probably not be key that a plaintiff was thirty-three years old. The court would have reached the same result if the plaintiff were thirty-two or thirty-four. Age may have been irrelevant or of very minor importance to the holding. What may have been key is that the plaintiff beat his wife, because without this fact the court may not have reached the conclusion that the ground of cruelty existed. The opinion is carefully combed to read what the judge said about the various facts. Did the court emphasize certain facts? Repeat them? Label them as crucial or important? These are the kinds of questions one must ask to determine which facts in the opinion were keys. In this regard, with the aim of assisting legal professionals in precedent retrieval, this study attempted to develop a CBR-system. Accordingly the required sample decided cases were taken from the Federal Supreme Court of Ethiopia. The next section provides an overview of the court.

3.3 The Federal Supreme Court of Ethiopia

The Supreme Court of Ethiopia was established in 1975 having exclusive first instance

Jurisdiction over the following:¹⁰

- Offences for which officials of the Federal Government are held liable in connection with their official responsibility
- Without prejudice to international diplomatic law and custom, offences for which foreign ambassadors, consuls as well as representatives of international organizations and foreign states are held liable, and
- Application for change of venue from one Federal High Court to another or to itself, in accordance with the law.

The Federal Supreme Court has also appellate jurisdiction over:

- Decisions of the Federal High Court rendered in its first instance jurisdiction;
- Decisions of the Federal High Court rendered in its appellate jurisdiction in variation of the decision of the Federal First Instance Court.

In addition to these, in cases where they contain fundamental error of law, the federal Supreme Court, have the power of cassation over the following final decisions of:

- The Federal High Court rendered in its appellate jurisdiction;
- The regular division of the Federal Supreme Court;
- The Regional Supreme Court rendered as a regular division or in its appellate jurisdiction.

The Federal Supreme Court has Civil, Criminal and Labor as major divisions of cases.

¹⁰ Jurisdiction of the Federal Supreme Court. Proclamation No. 25/1996. Federal Negarit Gazeta, 1996.

In any type of case, the court process starts when the applicant submits application with all relevant information and supporting documents to the Registrar's office. The registrar accepts a valid application and the applicant is conducted to effect payment accordingly. Upon the applicant submits the receipt; a file is opened for the case by the file opener and the following is recorded on the file:¹¹

- Date of application
- Plaintiff name and details
- Defendant name and details
- Case type (Civil, Criminal, Labor)
- Victims
- Amount paid and cash receipt number and
- Schedules.

Any type of case at any level of court passes through a series of logical and cyclic activities. The process starts when the file is opened for the case and first appointment is set. The case can go through a series of repeated hearings and adjournment, ends when the case is disposed. A disposed case can be reactivated due to various reasons that might imply entering again into same appointment-hearing-adjournment cycle. Thus, in addition to the items listed above as components of the case file, adjournments, judgments and court orders will be added.

As the working language of the Federal Government of Ethiopia, the legal cases considered for this research are also written in Amharic. Accordingly, it is appropriate to describe the Amharic writing system. The next section addresses this issue.

¹¹ Case Management System (CMS) Software Development Project: Requirement Analysis Report

3.4 The Amharic Writing System

3.4.1 The Amharic Characters

Amharic is the working language of the Federal Government of Ethiopia. The present writing system of Amharic is taken from Ge'ez alphabet, which was the language of literature in Ethiopia in the early time. The Amharic writing system consists of a core of 33 characters each of which occurs in a basic form and in six other forms known as orders. Each graphic symbol represents a consonant together with its vowel. The vocalic symbol cannot be detached from the consonant element. That is, Amharic does not use independent symbols for vowels. In other words, as Leslau (1965) discusses, the Amharic script is a syllabic rather than an alphabet.

The seven orders represent the different forms of a consonant. Each form is made in accordance with the sound that goes with the symbol. The non-basic forms are derived from the basic forms by more-or-less regular modifications. For example, using the consonants **ሀ** (hä), **ለ** (lä) and **መ** (mä) the order is:

1 st	2 nd	3 rd	4 th	5 th	6 th	7 th
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
hä	hu	hi	ha	he	h	ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
lä	lu	li	la	le	l	lo
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ

mä mu mi ma me m mo

In addition to these, there are four so-called labio-vellars, which have five orders (e.g. ቂ ቃ ቄ ቅ ቆ), and eighteen additional labialized consonants (e.g. ቧ ባ ባጫ ባጭ ...). The language has also its own numeral symbols (without a symbol for 0), but not mostly used except for writing dates. A listing of the whole Amharic character set, also called fidel (ፊደል) (attached in Appendix I).

The Amharic alphabet, all in all has around 290 letters. The alphabet does not have any distinction between capital and lower case letters.

3.4.2 Punctuation Marks

The individual words in a sentence are separated by two dots ‘:’, although this practice is not exercised especially in type-written texts. The end of a sentence is marked by a square-formed four dots ‘::’. The symbol ‘፣’ represents a comma, while ‘፤’ corresponded to a semicolon. Apart from this, the language has borrowed some punctuation marks (e.g., ‘!’ and ‘?’) from foreign languages.

3.4.3 Problems with the Amharic Writing System

There are many problems observed regarding the writing system of Amharic language. Some of them are summarized below.

3.4.3.1 Problems Regarding Consonants with Different form

As discussed above, Amharic borrowed most of its scripts from Geez. However, it did not select from Geez alphabet those symbols that are only necessary for its consonants. As a result, there are certain phonemes with different symbols, where they have meaning in Geez, but their meaning is not known in Amharic. The following table shows the list of consonants that have different forms.

Consonant	Other forms of the Consonant
ሀ (hä)	ሐ and ኀ
ሠ (sä)	ሰ
ኧ (ä)	ዐ
ጸ (tsä)	ፀ

Table 3.1 Amharic Alphabets with Different Forms

The pronunciation of these paired characters is the same and each of them has their own orders.

The distinction between these symbols in Geez is when spelling certain words, however there is no rule as for their usage in Amharic language, and as Getachew (1967) stated, the proper use of these symbols is not studied exhaustively and also there is no standard dictionary to refer to. Therefore, it is not clear whether one should write “häyl” (power) as **ሀይሉ**, **ሐይሉ** or **ኀይሉ**, also “tsähäy” (sun) as **ፀሐይ** or **ጸሐይ**. As a result there arises some confusion and inconsistencies in Amharic spelling, and these redundant consonants are assumed surplus.

3.4.3.2 Problems Related to Certain Interchangeably Used Orders

Beletu (1982) mentioned the confusion regarding the first order and the fourth order of some consonants. For instance, it is not clear which one to choose **ሀ** (as **ሀይሉ**) or **ሃ** (as **ሃይሉ**) to spell ‘Häylu’ – name of a person. As a result, one can find the same word ‘Häylu’, spelled differently in six forms, which are **ሀይሉ**, **ሃይሉ**, **ሐይሉ**, **ከይሉ**, **ኅይሉ**, and **ኃይሉ**. Similarly, “eye” can have three forms, which are **ዐይን**, **ዓይን**, and **አይን**, with pronunciation “äyn”.

It has also been found that the second order of the consonant **ወ**, which is **ወ**, and its sixth order, which is **ወ** are interchangeably used and there is no consistency. Because of this, one can find the word ‘dog’ spelled as **ወሻ** and **ወሻ** (“wshä” and “wushä” respectively).

3.4.3.3 Problem regarding compound words

Bender and Ferguson (1976) has mentioned another problem regarding the division of compound words. For example, it is not clear which one, **ወጥቤት** “wät’bet” or **ወጥ ቤት** “wät’ bet” is the correct spelling for ‘kitchen’. They also mentioned another problem of the writing system, which has something to do with regularizing spellings and regularizing punctuation. For example, the word “sämtoäl” (‘he has heard’) may be spelled as **ሰምቶአል**, **ሰምቷል** or **ሰምትዋል**. This problem exists in different languages that have words of different forms of writing. For example, the words ‘recognize’ and ‘recognise’ in English language are two variants of the same word.

Translation from foreign words into Amharic is also another problem mentioned by Getachew (1967). As a result, one can find the word ‘television’ translated into different forms, which include: ቱሌቭዥን, ቱሌቭዥን, ቱሌቪዥን.

Therefore, any automatic Amharic text processing should consider the aforementioned problems.

3.4.3.4 Problem regarding abbreviations

As seen from documents reviewed, there is also no consistency when spelling abbreviations. For example, when abbreviating the phrase ማህተም (in the year AD), one can find ማ.ም., ማ.ም or ማም as possible abbreviations. So these kinds of words should come into a common word. Similarly, the use of hyphen is not consistent.

3.5 Amharic Software

Since Amharic is not known in the ASCII code table, various software experts have tried to develop their own keyboard driver program, which converts the English keyboard into Amharic keyboard. Technically speaking, the software convert the default code table where each key is associated with English symbols by Amharic code table, so that users can use the same keyboard to write and edit Amharic letters. In other words, these programs associate the keyboard buttons with Amharic symbols.

In fact, this is done at the screen level. That is, the symbols stored inside files are the associated ASCII symbols of the default code table, not the Amharic symbols. The software converts these symbols into associated Amharic symbols when they are read

from files into memory. As a result users see Amharic symbols on their screen and printouts.

All Amharic software have succeeded in helping users to enter and edit Amharic text inside the computer (especially for word processing purpose). However, all face some problems especially when they are used for database purpose.

The number of the Amharic characters is beyond the number of ASCII codes and the available buttons on the keyboard. As a result, all Amharic software implement the assignment of Amharic symbols (especially the non-basic forms) as two characters. For example, the fidel "ሀ", for example, is stored as two characters "ሀ" and the diacritic mark " · " inside the computer. The codes assigned to the characters did not also consider their precedence in the language. Because of this, for example, when we sort the data (ሀገር, ሂዳብ, ሀዳብ), we will find ሂዳብ (not ሀገር) first. Therefore, any text processing should consider this as well. That is, consider these characters as two symbols, not as is seen outside (one).

On the other hand, when the software implement the representation of symbols borrowed from other languages, most of them have changed the codes of these symbols. For example, the ASCII code for '?' (Question mark) is 63 in the default code page? However, Visual Geez version 2.0 has implemented it by giving code 41.

Therefore, any string searching and matching procedure should consider the codes, as implemented in each software, not the symbol when searching for these symbols. In other

words, the search should not be done, for example by using “?” to search question mark symbol as it is done in any string searching program.

As mentioned above, when searching for Amharic symbols we can use their equivalent in the default code page. For example, the code for ‘ሐ’ in Visual Geez is the code for ‘/’ in the default code table. This means, ‘ሐ’ is stored inside files as ‘/’. Accordingly, to search for ‘ሐ’ one can use either its ASCII code or use “/” in the search statement.

CHAPTER FOUR

DESIGN OF THE PRECEDENT RETRIEVAL SYSTEM

4.1 Introduction

The explicit request for reuse of decided legal cases, called for the application of Case-Based Reasoning (CBR). The requirement for the system was hence to build a knowledge base in which complete decided cases could be entered and then recalled when similar cases arose again. According to Hafner and Berman (2000), research in case-based legal reasoning has a more restricted meaning: formal judicial decisions (i.e. legal cases) play the role of remembered experiences, and the problem solving goal is the construction of precedent based adversarial legal arguments.

This research is aimed at retrieval of cases based on the similarity between the facts and legal issues of a prior case and those of a new case. For this purpose domain dependant, legally relevant factors are defined, and each prior case is considered more or less on point depending on the similarity between its factors and those of the new case. So as to extend the usability of the system an online interface has been developed.

The remaining part of this chapter describes the different phases involved in the development of the CBR based precedent retrieval system, the experimentation and discusses the result.

4.2 Selection of CBR Tool

For this study to focus on the issues involved in case representation and retrieval, using an existing CBR shell was appropriate. After, a review of a variety of available CBR tools, CBR-Works has been selected. CBR-Works is a shell for case-based application building owned by Tecinno. It is a software development package suited for intelligent solutions in a variety of domains and environments. Besides the retrieval of cases, it supports modeling the cases' structure and maintaining the case base. Though CBR-Works is designed as a complete environment, it also acts as a CBR-Server for several clients by the use of CQL (Case Query Language). Last but not least, CBR-Works offers an open interface (ODBC) to build a case-based application from existing data stored in common database systems. Hence, any source (e.g., sheet or database) which contains the domain data can be connected to CBR-Works to import structure and data to build up a CBR-System. Some of the merits of CBR-Work can be summarized as follows:

- Available on all major operating systems
- Support full power of object-oriented modeling
- Allows to assign attribute weights which express the importance of concept attributes
- Direct ODBC to databases
- Query weights can be assigned that express user preferences
- Query filters control the search for suitable alternatives

4.3 The Legal Precedent Retrieval System

An important routine task of law practitioners is to find suitable cases, which can be referred to in court to make their argumentation strong. Before starting a search for suitable case reports, practitioners have a legal issue and their client's factual situation in

mind. During the search they try to find cases related to the issue at hand, i.e., cases decided under similar circumstances. In addition to this they also consider some other factors including whether the point at the issue has been decided in ones favor or not in a precedent (Pasha and Soper, 1999). Retrieval of suitable cases for this purpose is an important issue in computer based legal research. Two well cited difficulties are query formation and irrelevant retrieval (Mulder and Combrink-Kuiters, 1996).

This research, therefore, attempted to develop a CBR-System which analyses the legal professionals request, compares it with the available data items in the case base and retrieves those data items which are relevant to their request. To demonstrate the system a prototype has been developed.

The issues addressed in the designing of the CBR-System are the definition and scoping of the domain, collecting and representing data, examining data and identifying significant features to design the case structure. The following sections provide a description of the different phases involved in the CBR-system development process.

4.4 Setting up the Experiment

4.4.1 Domain Definition

Before the search routine for the relevant cases could be put into operation, it was necessary to define the specific legal subject of the research. After some preliminary

investigation (i.e., discussion with legal professionals and review of cases), it was decided to focus on labor law.

Labor law has the generic objective of safeguarding employees and employers in the work environment. These objective express government policy and the principles, which bring labor regulations into existence. These objectives or principles are translated into more detailed issues in the different subject areas: remuneration, suspension, termination, reduction of workers, severance pay and compensation, etc. These issues or requirements are very general and expressions are highly abstract. The argument surrounding each appeal decision is often complex, involving the balancing of a variety of assertions by the parties concerned.

Legal experts make a decision after consulting all information relevant to a given case. In addition to statute and case histories relevant to the given case, the experts should refer to all information available to provide justification for their decision. The problem is therefore to support the consultation of these loosely expressed legal requirements without being forced to make explicit all the information needed to arrive at a decision.

Further more, discussions conducted with domain experts revealed the fact that the cases the labor court handles tend to be redundant and increase in number very rapidly. Consequently, provision of a system to retrieve precedents in the domain could have a paramount role in assisting legal professionals.

4.4.2 Collection of Decided Cases

The heart of a CBR-System is the case base containing the knowledge of the domain to be represented. Accordingly, the first step in the system development was to collect a homogeneous set of relevant court cases. Building a CBR-System from scratch is necessary and appropriate for domains that are not available in electronic form (Leake, 1996). For information being stored in, e.g., a database, a CBR-Tool must be able to reuse such data rather than having the user to remodel the domain and manually add all information to the case base. Since the Federal Supreme Court of Ethiopia maintains a database of cases it was selected as source of the required data for this research. However, at a later time during the experiment, the attributes used by the existing database were not found sufficient for the purpose of this research. As a result, building the CBR-system from scratch was reconsidered.

For this research a total of 39 labor cases were selected. In view of the volume of work involved in developing the CBR based precedent retrieval system from scratch, it was only possible to prepare 39 cases. However, effort was made to make the cases as representative as possible and in a manner to support the findings and conclusions drawn.

4.4.3 Feature Selection

A CBR implementation, in any domain, requires a detailed analysis of the environment, as it is strongly related to the type of problems being solved or decisions being supported (Haque, 1999). After review the selected cases and several discussions with sixteen attorneys (all of them having experience on labor cases) from the Federal Supreme Court

of Ethiopia, important features of a labor case have been identified. Table 4.1, presents the identified features:

Feature	
ከሳሽ (käsäsh)	Plaintiff
ተከሳሽ (täkäsäsh)	Defendant
ጉዳይ (gudayu)	Case type
ያሸነፈው አካል (yashänäfwu akal)	Winner
የተከፈተበት ቀን (yätäkäfätäbät qän)	Case opening date
የተዘጋበት ቀን (yätägäbät qän)	Case closing date
ወሳኔ (wusani)	Verdict

Table 4.1 Selected Features for the Precedent Description

4.4.4 Concept Definition

Here, the term *concept* defines the structure of the cases. They are defined in hierarchy similar to a class-model hierarchy including inheritance. Each *concept* consists of attributes, which can be either atomic (defined by a *type*) or complex (has-part relationship to another *concept*). Every concept has its own attributes. For retrieval purposes, attributes have three additional, functional properties: one for defining its weight, i.e., its importance in respect to the other attributes of the concept, a property for defining whether an attribute is discriminant for retrieval or will be ignored, and another property defining if an attribute is mandatory for a case to be valid. Using the expert's knowledge made the assignment of weight to the attributes and defining them as discriminant or mandatory. Moreover, for some attributes a question has been given that can be used by clients when querying the system. Figure 4.1 shows the concept definition for labor legal cases.

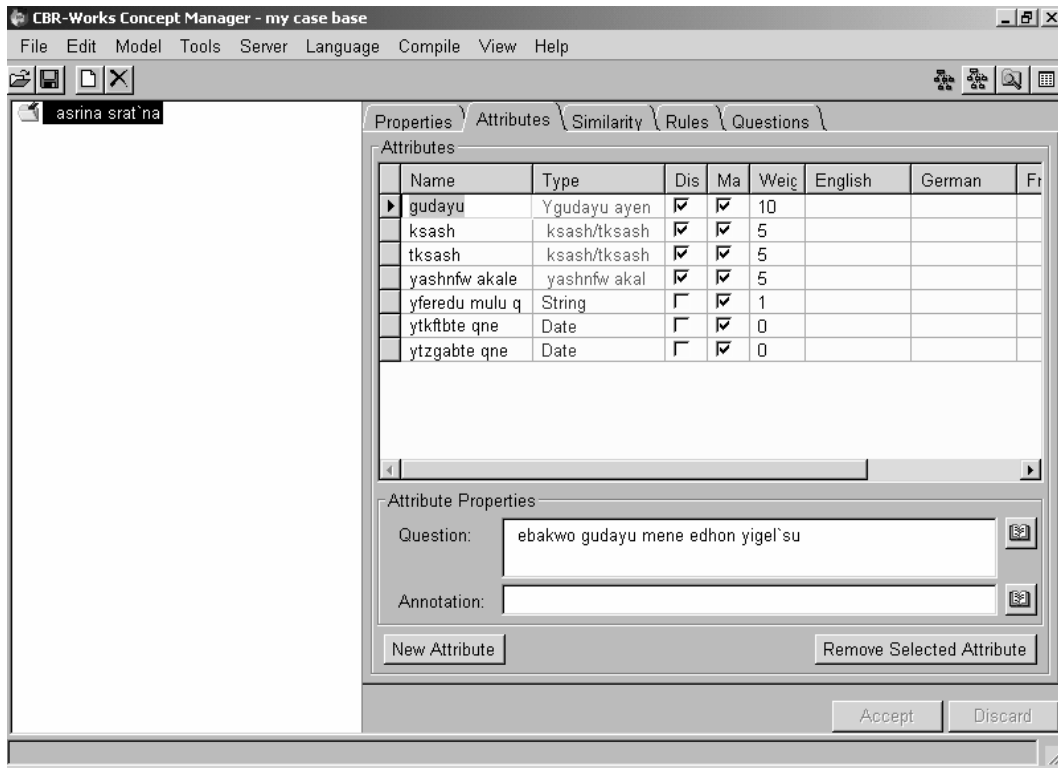


Figure 4.1 Concept Definition

As depicted in the above figure, the concept 'asärina säratäña' has been defined to represent the structure of labor legal cases. The case features identified in section 4.4.3 have been used as attributes to describe the concept. As shown in the figure, ገዳዩ 'gudayu', ካሳሽ 'käsash', ተካሳሽ 'täkäsäsh', and ያሽነፊ.ወ. አካል 'yashänäfäwu akal' are discriminant attributes i.e., they are used for similarity assessment between a new case and cases in the case library during retrieval. All the seven attributes are also defined as mandatory i.e., in order to add a new case into the case base their values should be given. In addition to this, every attribute has a defined data type. The next section describes attribute type definition.

4.4.5 Attribute Type Definition

Similar to concepts, types are defined hierarchically. New types are defined by building subtypes of the existing elementary types (e.g., integer, symbol, string etc). A type may be used *immediate* or *derived*. While *immediate* types cover the whole range of possible values of a type, *derived* types get restricted in their range by defining an enumeration of elements of its elementary type or, in case of numeric types, by specifying an interval. For this study, the data types for attributes ክሳሽ (käsäsh); ተክሳሽ (täkäsäsh); ጉዳዩ (gudayu); ያሕናፊዉ አካል (yashänäfäwu akal) are defined as derived types. While the remaining attributes are immediate types.

The respective range for the discriminant attribute's type has been defined in consultation with domain experts and by referring the Labor proclamation No. 42/1993. Table 4.2 shows the derived atomic types and respective ranges for the first three discriminant attributes. The derived hierarchical type and the associated range for the remaining discriminant attribute ጉዳዩ 'gudayu' is depicted on Table 4.2.

Attributes	Defined Data Type	Rang
ክሳሽ (käsäsh)	ksash/tksash	አሰሪ 'asäri', ሰራተኛ sätatäña
ተክሳሽ (täkäsäsh)	ksash/tksash	አሰሪ 'asäri', ሰራተኛ sätatäña
ያሕናፊ ጸሀፊ (yashänäfäwu akal)	yshnfwu akale	ክሳሽ (käsäsh), ተክሳሽ (täkäsäsh)

Table 4.2 Attribute types and the respective range

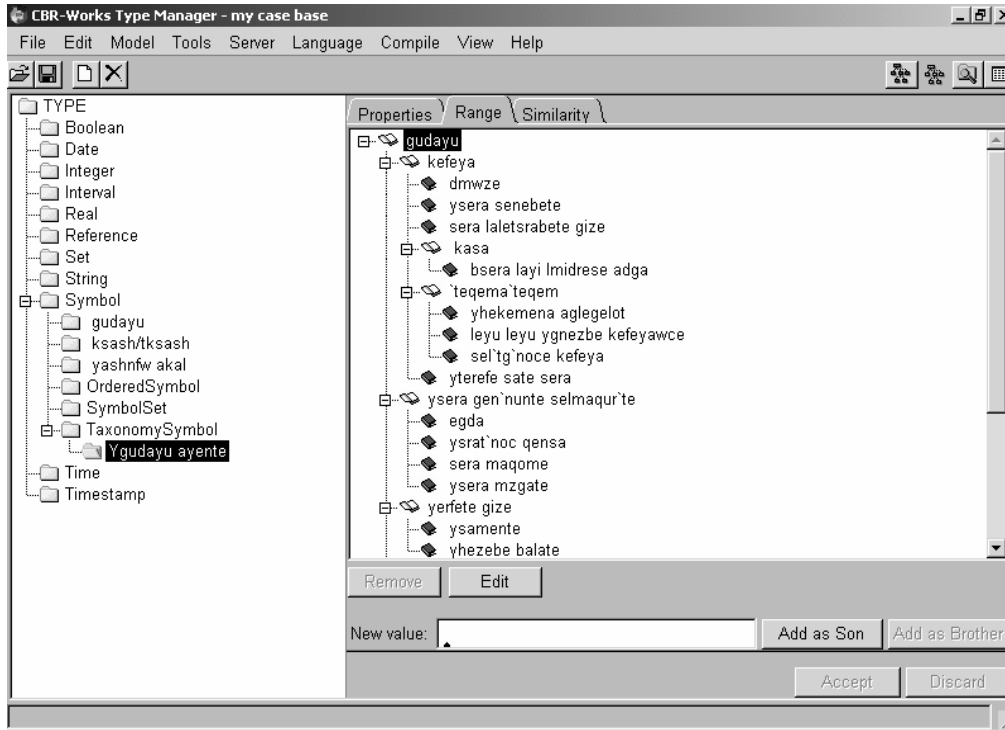


Figure 4.2 The Defined Data Type and The Ranges for the Attribute 'gudayu'

4.4.5.1 Type Similarity Specification

Selecting the best case requires being able to match with the case in the case base. In general the matching is not exact matching, because on the one hand, the values of the features of the new case and previous cases may not exactly the same and on the other hand there are usually missing values for some or many of the features. In addition to these, there might be a similarity of some level between two or more different values of an attribute. Therefore, in this study for each type derived from elementary types, similarities among the possible values of the attribute (range) were defined describing major parts of the experts' knowledge. In order to get the similarity metric, sample attorneys were asked to give a value ranging from 0(none similar) up to 100(exact match). The average of there responses was calculated and used for the similarity metric

definition. An example of a similarity metric for the attribute ‘gudayu’ is given in figure 4.3. Labor case type being different in a retrieved case to a specified value in the query.

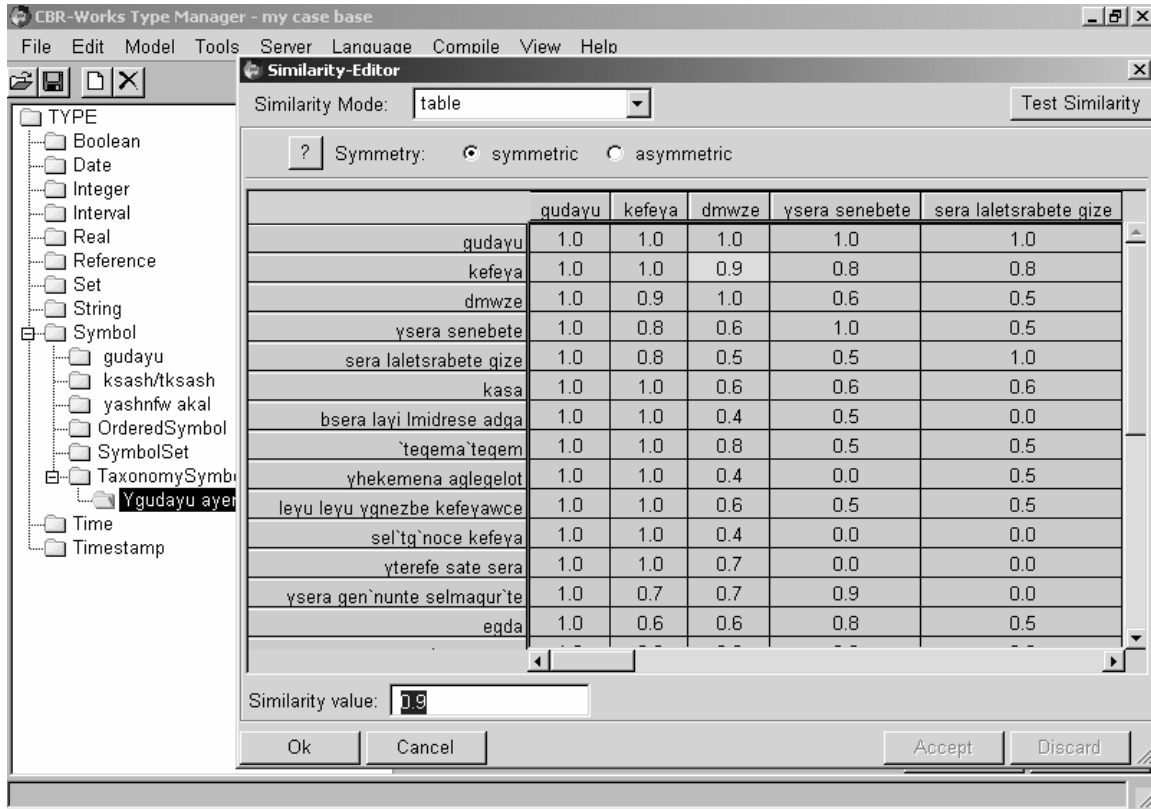


Figure 4.3 Example of a similarity table for the case type ‘ygudayu ayente’

4.4.6 Rule Definition

General background knowledge (rule) is necessary to better explore and interpret the available cases. Such general knowledge may state a strict dependency of one feature of a case on several other features of the same case. This allows to infer additional, previously unknown features from the known ones.

A CBR system may define two types of rules namely: completion rules and adaptation rules.

- *Completion rules* infer additional features out of known features of an old case or the query. Thereby, these rules complete description of a case.

- *Adaptation rules* describe how an old case can be adapted to fit to the current query

If some attribute values depend on each other, completion rules ease handling by automatically setting appropriate values. Adaptation rules get activated only after retrieval and they are used to combine attribute values of the query and retrieved cases and to apply the result to a target case. Since the goal of this research is to develop interpretive CBR-system, there was no need for defining adaptation rules.

Important for a consistent case base is the maintenance of its cases concerning validity of values and changes to the underlying model of the domain (Harrison, 1997). Accordingly, defining completion rules ensures that each case is confirmed or protected to be valid regarding modeled type-ranges after inserting or modifying a case in the case base. Figure 4.4. Shows an example of a compilation rule defined for the system.

The completion rule on the above figure helps the user of the system both in adding a case to the case base and query formation. In several situations, features of a case description are directly dependent on several other features. As a result, when the user enters some of the features in the query, she/he should generally not be demanded to enter the values of features which are absolutely determined by the information she/he has already entered. For instance, if the user entered the value ሰራተኛ 'säratänn' for the attribute ከሳሽ 'käsäsh', the system will automatically assign the value አሰሪ 'asäri' to the attribute ተከሳሽ 'täkäsäsh'.

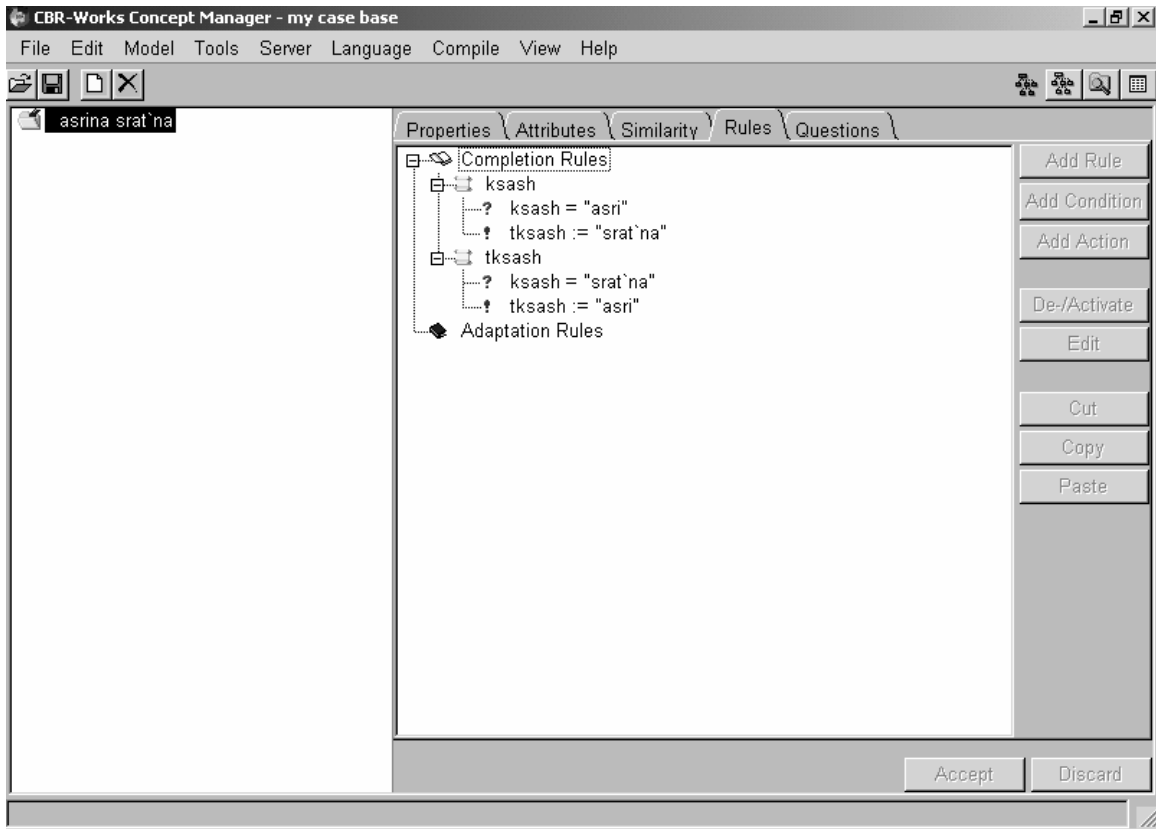


Figure 4.4 A Sample Completion Rule

4.5 The Case-Base

As it has been stated in the first chapter of this thesis, the objective of the study is to provide the users with the most useful cases to support their current legal problem. Accordingly, the legal case-base is supposed to be an effective form of knowledge disseminating and reuse in the legal domain, helping to overcome the cumbersome task required in accessing legal precedents.

The case-base is constructed in such a way to make the insertion and removal of cases a simple task. It is actually an incremental process, with no need to rebuild the case-base each time a case is added or removed. This way, the case-base can be regularly updated

by people not specialized in computer science (e.g., the judges, lawyers). Allowing the maintenance of a case-base to be done by domain experts and not by computer specialists is an important aspect for the acceptance and viability of the system. Figure 4.5 shows the case base of the CBR-system.

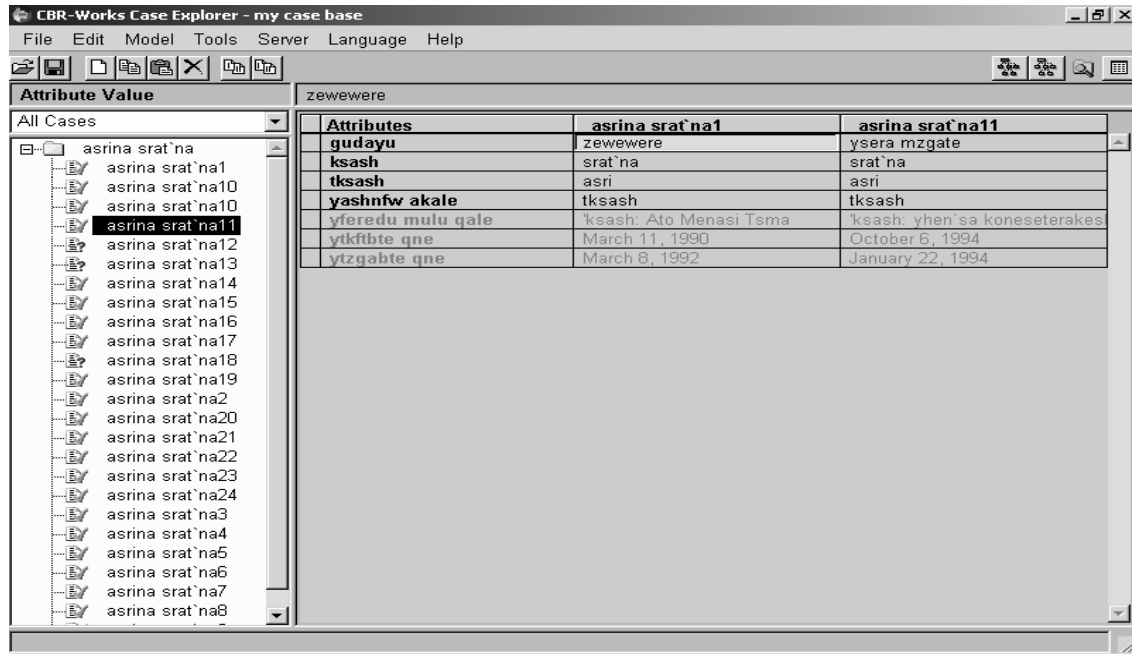


Figure 4.5 The Case Base

A case in the case base has four possible states: *unconfirmed*, *confirmed*, *protected*, and *obsolete*. Usually, new cases become *unconfirmed* being unrefined or incomplete cases not valid for retrieval. Particularly as this research provides an online interface to add new cases in to the case base, there is a need for new cases to remain unconfirmed until the system administrator revise them. Revised cases become either *confirmed* which allows for retrieval or *protected* which additionally protects the case from changes. Old cases, no longer valid for retrieval but probably useful for further statistics, become *obsolete*. In figure 4.5, those cases with a right mark indicate the cases are *confirmed* while; those with a question mark are *unconfirmed* cases.

4.5.1 The Case Representation

The case base in the CBR-system is presented as a table comprising N columns representing case attributes ($\{A_1, A_2, A_3, \dots, A_N\}$), and M rows representing individual cases ($\{C_1, C_2, C_3, \dots, C_M\}$). Consequently, each attribute has a sequence of possible values associated with it: $A = \{V_1, V_2, V_3, \dots, V_K\}$. Of course, knowledge of some attribute values may be unavailable and thus some fields may be left blank.

Each attribute is therefore, a dimension in N -dimensional space (the case space), and consequently each case is N -dimensional object (a case object) in this space whose location and shape is defined by the nature of the values for its associated attributes. Given a case for which all attribute values are known, these values act as coordinates which uniquely define a cell within the case space. Whether a value for an attribute is known or not, all possible values for the attribute are included when defining the nature of the case object.

4.6 Case Retrieval

The retrieval task starts with a new case description, and ends when a best matching precedent has been found from the case base. The following two points noted by Costa et al. (2001), served as a guideline for the retrieval aspect of the CBR system described in this thesis:

- The retrieval process must be complete; i.e., all relevant cases must be retrieved. Users would lose faith in the system if they detect that it was overlooking important cases.

- The retrieval process must be flexible, allowing the users to conduct a search with whatever information they possess at the moment. Forcing the users to always provide the same information to seed a search would be very restrictive, since different pieces of information are known at different times.

4.6.1 Case Similarity, Matching and Ranking

Selecting the best similar case(s) is usually performed in most Case-based reasoning systems by means of some evaluation heuristic functions or distances, which are possibly domain dependent. CBR-Works uses numeric schema called ‘Nearest Neighbor’ in this technique each dimension of representation is given an importance rating. Accordingly, the similarity between two cases is calculated pair-wise, between pairs of fields. The similarity between a query and a case from the case base is computed in a bottom up fashion. For each attribute, a local similarity measure (the similarity metric) determines the similarity between two attribute values and for each object (the case and the query) based on the local similarities of the belonging attributes. Figure 4.6 depicts how the nearest neighbor-matching algorithm works.

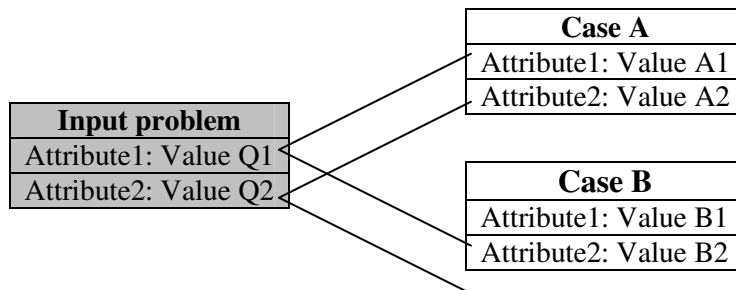


Figure 4.6 Nearest Neighbor Matching Algorithm

The final goal of the CBR system described in this thesis is to find the case which has the maximum similarity to the input case. Thus, a mechanism to rank the similarity of cases is needed. As it has been described in section 4.4.5.1, a vector of importance values (weights) for all the features of a case was defined. As a user provides a query for the system, each feature of the input case is compared with the corresponding feature in the stored case and a similarity score for that feature is calculated. An overall measure of similarity between the two cases is then calculated as weighted average of the feature scores. This measure of similarity is assigned as a rank for the retrieved case(s).

4.7 User Interface Development

The Internet (or more specifically the World Wide Web) has become the forum for information gathering and will surely be an essential tool of all modern lawyers. In this research an online interface has been developed and implemented. Three interfaces were developed for easy usage of domain models. Among the three interfaces, one is for adding a case to the case base while the remaining two are for the purpose of querying. The screen depicted on Figure 4.7 shows the interface which enable the users to query the system by entering the values of the discriminant attributes of the new case at hand. In addition to this, it allows the user to specify the similarity threshold value. Accordingly, it allows setting different threshold similarity values to broaden or narrow the search result.

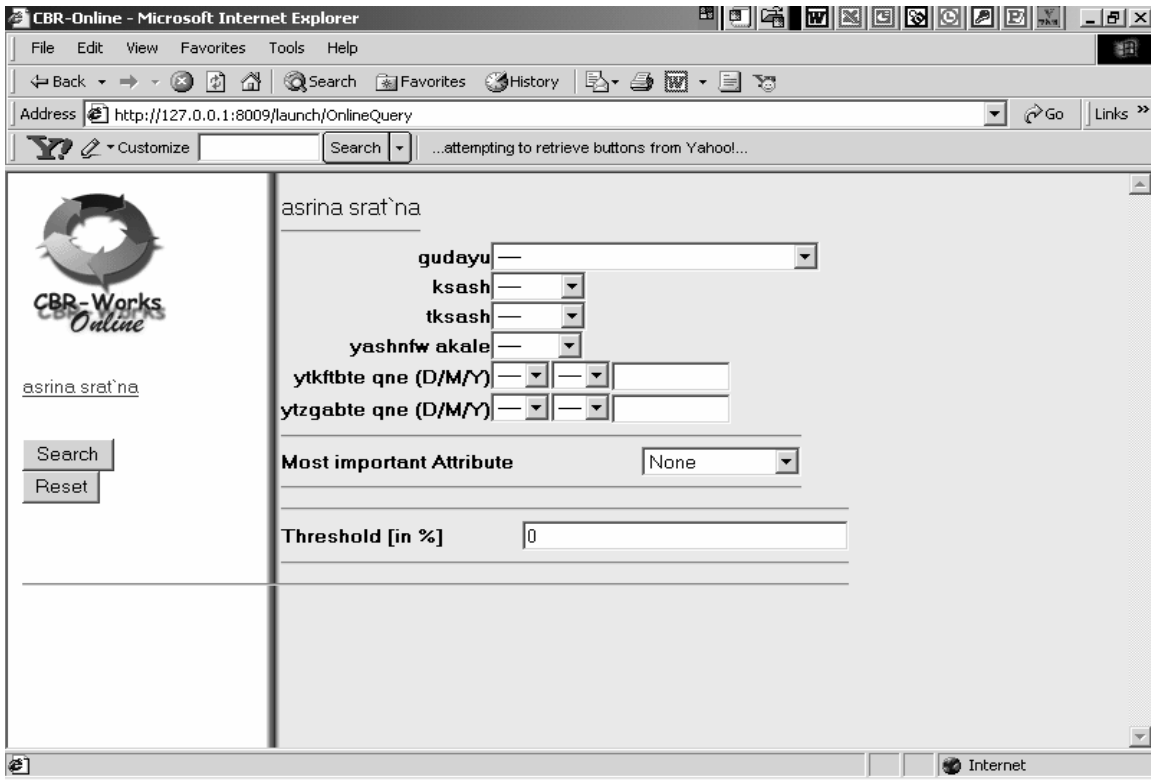


Figure 4.7 User interface window for online query (attribute: value)

The other interface provided to the user supports searching for case by answering questions related to the discriminant attributes. As it has been specified in section 4.3.3 during concept definition for every discriminant attribute a question is attached that can be used by clients when asking for a related case. Accordingly, these questions are presented to the user. Then the system will consider the answer given to the questions as the value of the respective attribute to which the question is associated. As a result of the search the system provides a ranked list of cases from the case base. Figure 4.8. and 4.9 show the wizard interface screen and the search result screen respectively.

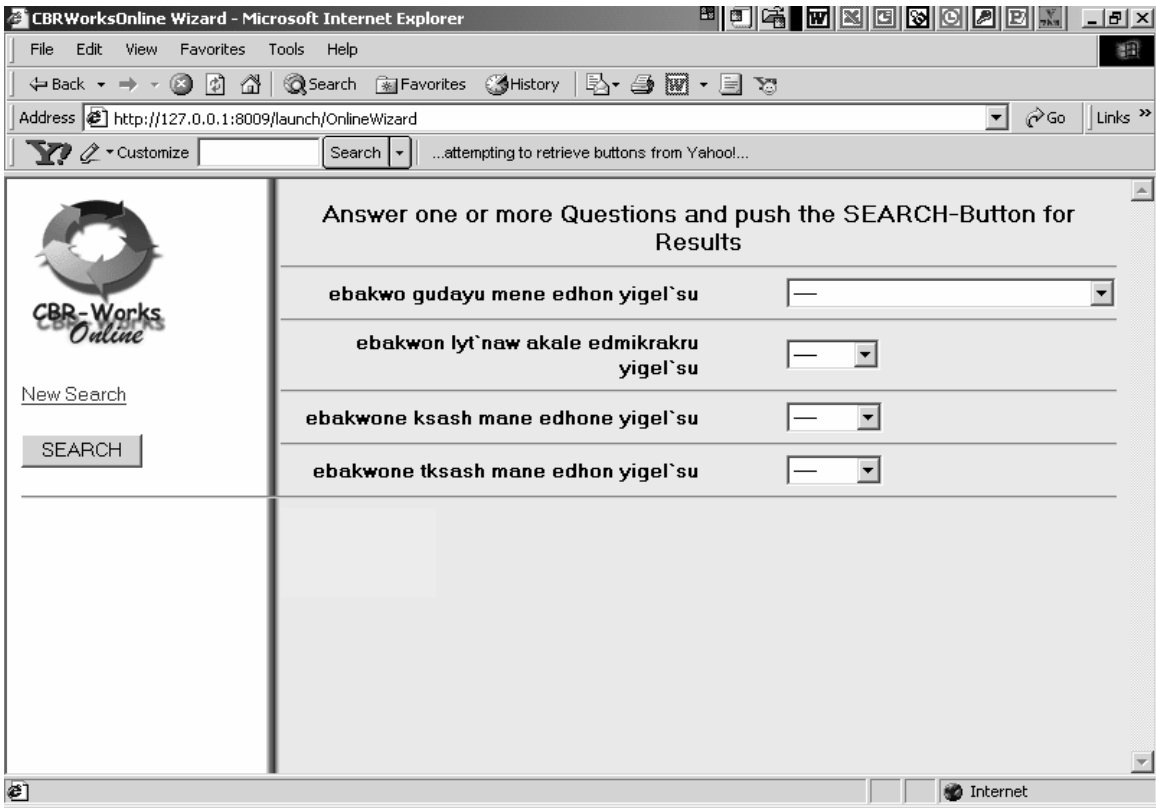


Figure 4.8 User interface window for online query (question: answer) wizard

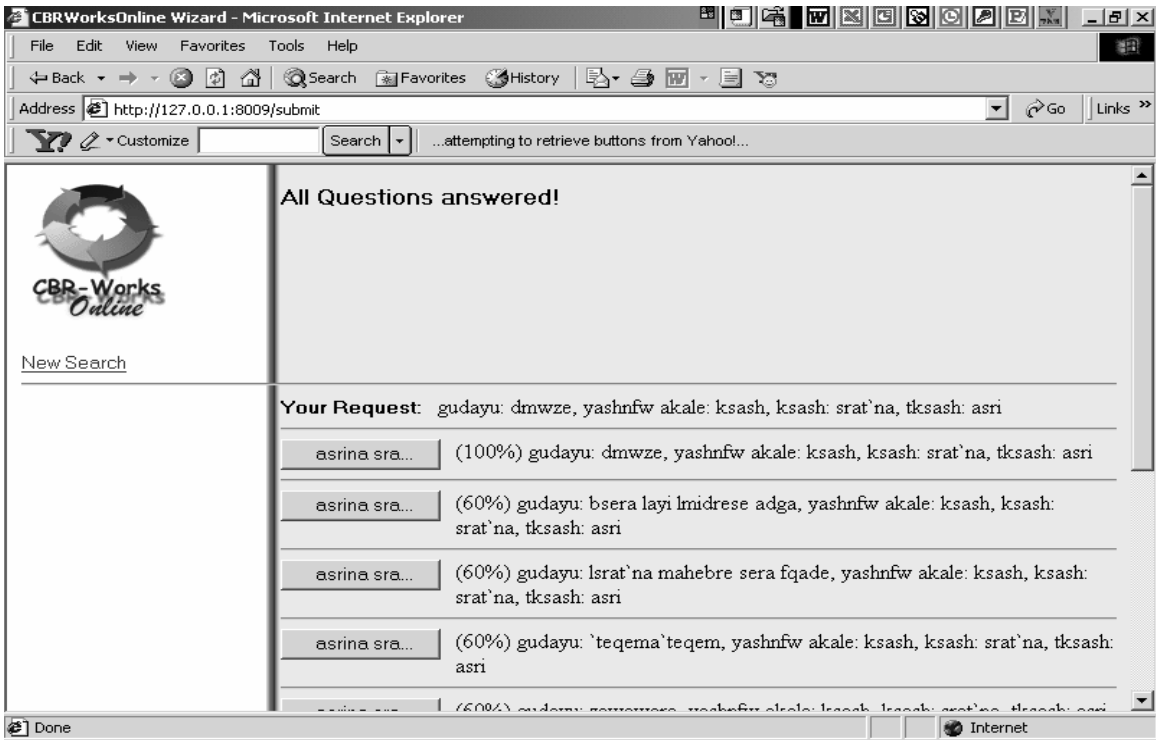


Figure 4.9 Query Result Window

The third type of interface provided, enables the user to add new cases into the case base. As it has been indicated in section 4.5.1 a case in CBR-Works has four possible states: *unconfirmed*, *confirmed*, *protected*, and *obsolete*. New cases become *unconfirmed* being unrevised or incomplete cases not valid for retrieval. Figure 4.10 depicts the online interface for adding new cases into the case base.

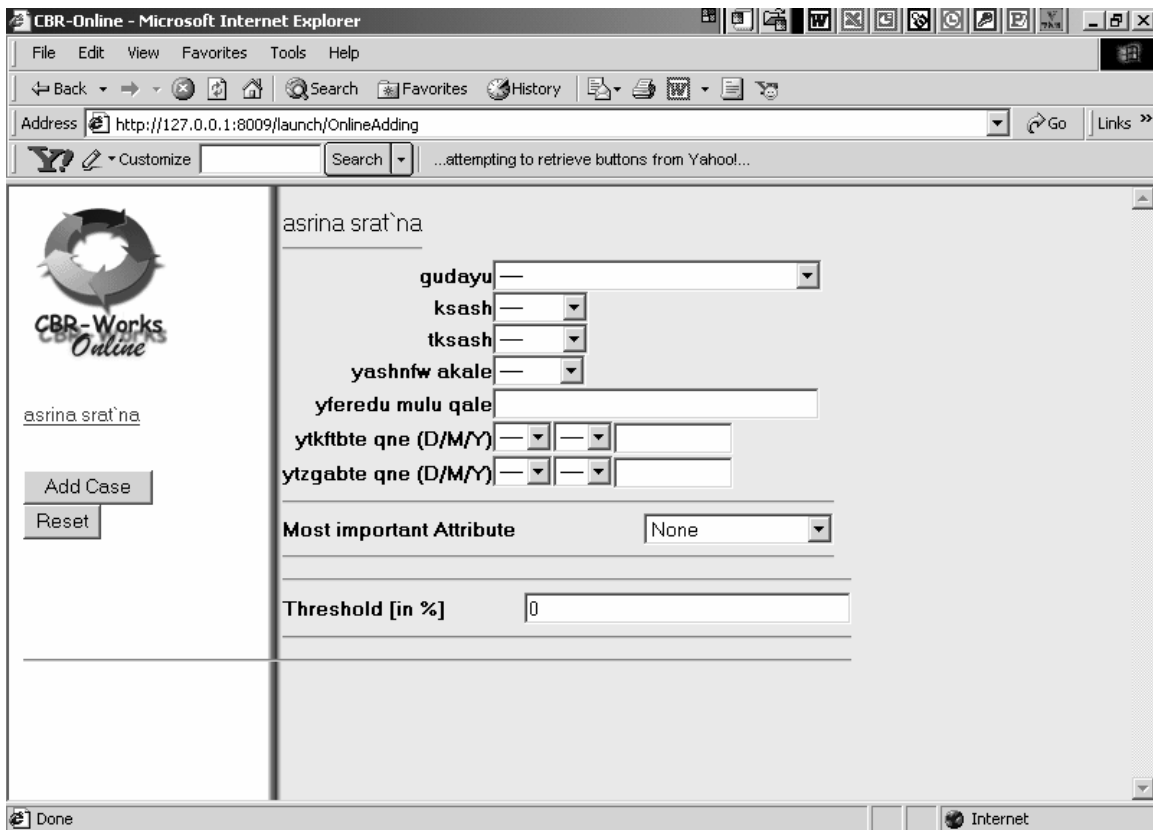


Figure 4.10 User interface Window for Case Entry into the Case Base

4.8 Empirical Evaluation of the Prototype

To evaluate the successes of the CBR-system two types of criteria has been used. The first one is on increasing users satisfaction in terms of the following issues:

- The users feedback, regarding:

- The relevance of the information retrieved (how suitable is the precedent to the case at hand)
- The relevance of the fields (Attributes) in the case structure
- The ease of use of the system, in terms of the entry forms layout and level of detail

The second one is based on precision and recall. The following sections present the test procedures and discuss the result.

4.8.1 Selection of Domain Experts

In order to evaluate the performance of the system domain experts¹² were selected. As it has been indicated in section 4.4.5.1, for attribute definition and the corresponding range similarity assignment, several discussions were held with 16 lawyers. As a result this group of users were believed to have better knowledge of the system. By taking this into consideration, the users for system evaluation were categorized into two groups. The first group constitutes, those who participated during system development. While those who do not have prior knowledge of the system were placed in the second group. The two groups constitute 7 members each. The reason for considering 7 out of the 16 attorneys in the first group was that it was only 7 attorneys who have been working on a labor case at the time of this research. Accordingly, the descriptions of the cases they were handling have been used as a query in order to evaluate the performance of the system.

¹² Domain experts here refer to lawyers who handled several labor cases

4.8.2 The Test Case

The two groups of users were asked to provide a description of the labor cases (query) they were handling at the time of this research. As a result a description of 14 new cases was given as a query for the system. Since the content of some of the queries was identical, there were only 10 distinct queries. The ranked search results from the system for each of the queries were given to the respective users. Then the users were asked to evaluate the performance of the system in terms of the relevance of the information retrieved (how suitable is the precedent to the case at hand), the relevance of the fields (Attributes) in the case structure, and the ease of use of the system, in terms of the entry forms layout and level of detail. The users were asked to assign values (Poor, Fair, Good, Very good, and Excellent). Table 4.3 depicts the result.

Evaluation Criteria	Group I	Group II
Relevance of the search result	Very good	Good
Relevance of the attributes describing the case	Very good	Good
Level of detail provided	Very good	Good
Ease of use of the system	Very good	Very good
Speed	Excellent	Excellent

Table 4.3 The Performance Evaluation of the System by Domain Experts

As shown on the table above, there is deviation in the assessment value given to the CBR-system performance by the two groups. It was also noted that the deviation is resulted from the variation of system familiarity of users. Because, after some orientation on the system has been given to the users in the second group, there assessment resembles with those in the first group. And yet, members of the second group emphasized the need to add more attributes to describe labor cases.

The evaluation technique described in this section provided more of a crude assessment of the performance of the CBR-system. Accordingly, the standard measures of relevance in the information retrieval community (precision and recall) were used. The next section describes the test result with precision and recall.

4.8.3 Evaluation with Recall and Precision

Recall and precision values are the most commonly used measures of the effectiveness of an IR system. Recall measures the proportion of relevant documents that have been retrieved (recalled) from all the relevant documents in the collection. Precision on the other hand, measures the proportion of relevant documents from those that have been retrieved (Salton and McGill, 1983). Both recall and precision, being ratios, give values between 1 and 0. The computation equations of recall and precision values for the retrieval of a number of documents for a query are given by:

$$\text{Recall} = \frac{\text{RelRet}}{\text{TotRel}}$$

$$\text{Precision} = \frac{\text{RelRet}}{\text{TotRet}}$$

Where:

RelRet -Number of relevant documents retrieved

TotRel- Total number of relevant documents in the collection

RelRet- Number of relevant documents retrieved

TotRet- Total number of documents retrieved

The ten queries described in the previous section are again used as a test set. Precision and recall generally assume that relevance judgment is known for the queries test sets. Accordingly, there was a need to acquire the case/query relevance judgment from the domain experts. For this purpose, five students from Addis Ababa University, Law School identified the relevant cases from the case base to each of the ten queries. Table 4.4 shows the case/query relevance judgment assigned by the students.

Query	Case
1	asrinatna3, asrinatna7, asrinatna21, asrinatna27, asrinatna32, asrinatna34, asrinatna35
2	asrinatna5, asrinatna6, asrinatna9, asrinatna17, asrinatna24, asrinatna31
3	asrinatna1, asrinatna4, asrinatna18, asrinatna22, asrinatna33, asrinatna39 asrinatna21
4	asrinatna11, asrinatna13, asrinatna15, asrinatna19, asrinatna23, asrinatna27, asrinatna29
5	asrinatna2, asrinatna8, asrinatna12, asrinatna13, asrinatna22 asrinatna25, asrinatna31, asrinatna38
6	asrinatna3, asrinatna5, asrinatna10, asrinatna25, asrinatna27, asrinatna36, asrinatna37
7	asrinatna6, asrinatna9, asrinatna24, asrinatna26 asrinatna30 asrinatna35
8	asrinatna8, asrinatna9, asrinatna13, asrinatna19, asrinatna20, asrinatna22, asrinatna29, asrinatna34
9	asrinatna1, asrinatna5, asrinatna12, asrinatna21, asrinatna36, asrinatna38
10	asrinatna14, asrinatna16, asrinatna20, asrinatna21, asrinatna38, asrinatna39

Table 4.4 Case/Query Relevance Judgment

4.8.3.1 Testing with Attribute Similarity and Weight

The next step is to define the experimental group. Three experimental groups have been defined, each group representing a modification in one or more of the system component so as to observe the effect on the system performance. There were two components- similarity and weight.

In the first group the domain expert similarities of all attribute values were not defined. And the weights of all attributes were assigned a uniform value of one. On the other hand, in the second group in addition to the domain expert similarities of all attribute values the respective attribute weights have been specified.

The recall/precision calculation result is as follows. In the first experimental group, all the cases indicated in the case/query relevance judgment table were retrieved (recall average is 100%) for every run. However, the precision was 13.56%. In the second experimental group, the average of recall was 95.05%, with average precision 82%. The recall and precision result of the second group is depicted in table 4.5 and Figure 4.11

Query	Recall	Precision
1	1	0.583333
10	1	0.857143
2	1	0.666667
3	1	0.666667
4	1	0.818182
5	1	1
6	1	1
7	0.5	0.428571
8	0.875	0.777778
9	1	0.857143

Table 4.5 Recall/

Precision Table

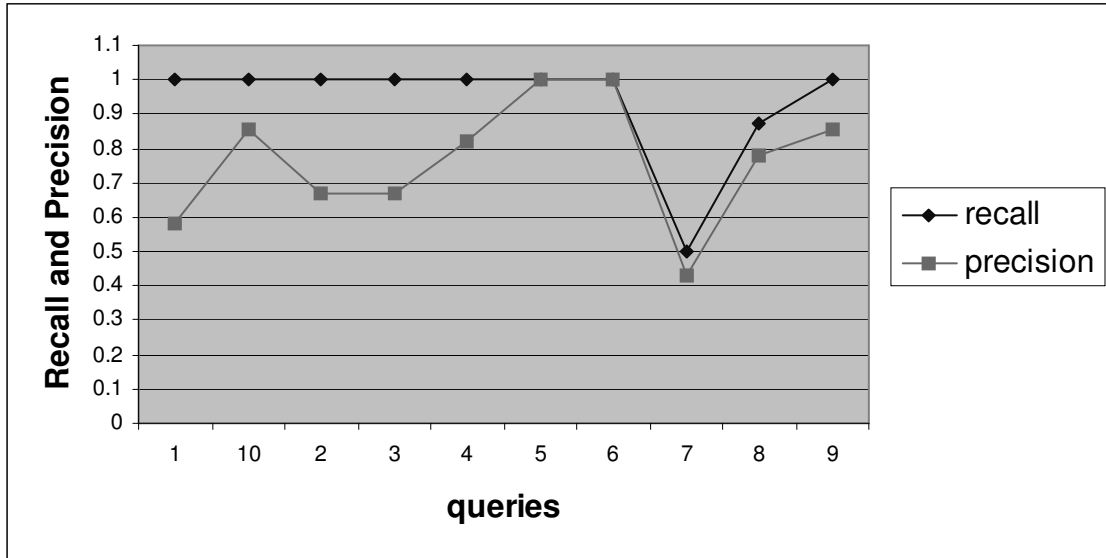


Figure 4.11 Precision/ Recall Graph

4.8.3.2 Testing with Varying Similarity Rank Threshold

A recursive testing has been made by varying the similarity rank threshold value. The test with the maximum threshold value (a database retrieval approach) resulted 100% precision but with 35.05% of recall. This did not satisfy the CBR-system objective-the retrieval of most similar cases. This test revealed that, the increase in threshold value increases the precision, while decreasing the recall. On the other hand decreasing the threshold increases recall, while decreasing precision.

The result from the two experimental groups revealed that the performance of the system increases proportionally with the level of domain knowledge integrated into the CBR-System.

4.9 Analysis of Results

The above results have provided some indications on the factors affecting the performance of the CBR-system. Some functional components of the prototype, such as specification of domain knowledge based similarity for all ranges of the attributes and assignment of weight for the attributes shown good performance. In other words, the performance of the CBR-system degrades if less domain knowledge is integrated.

The result of the experiment with varying threshold showed that the increase in threshold value increases the precision, while decreasing the recall. On the other hand decreasing the threshold increases recall, while decreasing precision. This is more of a database approach than CBR, as searching in a database is exact match, while that of CBR is partial similarity.

The other dimension that the experiment made to see is that, the usability and user satisfaction of a CBR-system is directly related to the system familiarity level of the user. Accordingly efforts must be exerted in providing appropriate user orientation.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

This thesis demonstrated the feasibility and promising case-based reasoning techniques to legal precedent retrieval. In this chapter, a summary of the problems addressed and preliminary results, and recommendation for future improvements are provided.

5.1 Conclusions

This thesis begins with the aim of developing a legal precedent retrieval system in the domain of Ethiopian Labor Law. The explicit request for reuses of knowledge and experience in the domain called for the application of Case Base Reasoning (CBR). The system described in this thesis enables the storage and access to legal precedents and extend knowledge in a natural and straightforward manner. Standard case representation to the original knowledge source (legal precedents) is used to store legal cases. Legal cases have a predefined case structure with a fixed number of features. The features are selected to reflect the important aspects of a legal case. The feature values are used to do the exhaustive search of the case-base.

In order to maintain the consistency of cases in the case base and to assist users during query formation, completion rules have been defined and incorporated in the system.

The retrieval task starts with a new case description, and ends when a best matching precedent has been found from the case base. In order to determine the similarity, for

each type derived from elementary types, similarities among the possible values of the attribute (range) were defined describing major parts of the experts' knowledge. In order to get the similarity metric, sample attorneys were asked to give a value ranging from 0(none similar) up to 1(exact match). The average of their responses was calculated and used for the similarity metric definition.

At last, given a query (a description of a new case), the system provides a ranked list of similar case which can be used by legal professional to handle the case at hand. In doing so, the system allows the user to specify a threshold value of his/her choice to narrow or broaden the scope of the search result. In order to make the system more usable an online interface was developed.

So as to evaluate the system's performance user satisfaction, precision and recall have been used. Given the limited number of cases and queries, the results are found to be promising. Based on the literature retrieve, experiments and the results presented in the previous chapter, the following conclusions may be drawn:

- Legal precedents are important elements that are supposed to be used during legal argument. Thus it is valuable to build a system to help professionals in the legal research process.
- Because the knowledge space for the legal precedent domain is extremely incomplete and context dependant, it is difficult to formalize general rules to help lawyers solve problems. In contrast, by using CBR techniques, a set of legal

precedents can be stored in a case library to guide the lawyer during argument writing.

- The prototype system also provides the flexibility to allow the user to broaden and narrow the search result by assigning different threshold value for the similarity rank.
- The prototype system developed as part of this work can be used as a base for future CBR systems, which can be expanded to other domains simply by adding domain knowledge.
- The fact that cases in the case library are assigned different status helps easy maintenance of the case base.
- The online interface developed for the system enables users to have remote access to the system. Such provision enables lawyers to focus upon offering advice and litigation support, rather than searching for precedents.
- This system accumulates case histories of interpretation of regulations used to establish precedents. These precedents can be used when revising statutory regulation and enrich the resulting new versions of regulations. They also provide relevant information for the experts to interpret and make decisions about cases coming to appeal.

- As the system helps in the retrieval of legal precedents, by studying previous similar to their own, legal agents can gather elements to guide their actions; judges can make consistent rulings, and lawyers can seek arguments that favor their case.

5.2 Recommendations

Although the results of this study are promising, further work needs to be done in order to upgrade the prototype system to a real system. The field of law offers a fertile domain for the study of important problems in artificial intelligence, such as using prior experiences to solve new problems and making persuasive arguments. Promising new directions for research include the development of formal legal ontology¹³ and their use in knowledge-based systems.

Besides, given the inheritance characteristic of concepts as an object in the CBR-system developed, other case types (in addition to labor cases) can easily be defined and added into the existing case structure. In doing so, future research can minimize the burden of structure definition, and can emphasize on the level of domain knowledge to support the retrieval process. As the results of this research revealed the more domain knowledge is available, the higher will be the performance of the system.

¹³ Ontology is a rigorous specification of a set of specialized vocabulary terms sufficient to describe and reason about the range of situations of interest in a domain.

A continuation of this research also can assume on extending the scope of the system to develop legal arguments by adapting the retrieved precedent cases. This will increase the usability of the system in assisting legal professionals

The CBR approach may be employed at several levels. Retrieved cases can be used to form the basis of an argument to a solution, or used as the input into algorithms for constructing legal arguments using cases.

REFERENCES

- Aamodt, A and Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AICom-Artificial Intelligence Communications*. 7, 39-59.
http://www.iiia.csic.es/People/enric/AICom_ToC.html
- Aha, D. W. (1991). Case-Based Learning Algorithms. In *Proceedings of the DARPA Case-Based Reasoning Workshop*. Washington, D.C.: Morgan Kaufmann.
<http://www.aic.nrl.navy.mil/~aha/papers/aha-cbr91.ps>
- Ashley, K. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. PhD thesis, Department of Computer and Information Science, University of Massachusetts Amherst.
<http://home.nvu.edu/~ss74/ashley.html>
- A-TEAM (2001). *AI Tools: CBR Implementation Report*. Information Societies Technology. <http://cl4.org/projects/A-REAM/interim.doc>
- Beletu Reda. (1982). *A Graphemic Analysis of the Writing System of Amharic*. Paper for the Requirement of the Degree of Bachelor of Art in Linguistics. Addis Ababa University.
- Bench-Capon, (2001). *Issues in Argument and Legal Reasoning..* Liverpool:University of Liverpool
<http://www.csd.abdn.ac.uk/~tnorman/sac/Bench-Capon.rtf>
- Bender, M., and Ferguson, C. Eds. (1976). The Ethiopian Writing System. In *Languages in Ethiopia*. London: Oxford University Press.
- Berman, D. H. and Hafner, C. D. (1991) Incorporating Procedural Context into a Model of Case-Based Legal Reasoning. In *Proceedings of the 3rd International Conference on Artificial Intelligence and Law*. New York: ACM Press. <http://www.compapp.dcu.ie/~tonyv/MIND/dan.html>
- Branting, L. K. and Porter, B. W. (1993). *Rules and precedents as Complementary Warrants*. <http://citeseer.nj.nec.com/branting91rules.html>
- Burstein, M. (1989). Analogy vs. CBR: The Purpose of Mapping. In *Proceedings of the DARPA Case-Based Reasoning Workshop*. San Mateo: Morgan Kaufmann.
- Civil Procedure Code of the Empire of Ethiopia* (1965).

- Costa, M. et al. (2001). *Case Retrieval Nets in AI and Law*. Port, Portugal: Instituto Superior de Engenharia do Porto, Departamento de Engenharia Informatica.
- Curran, K. and Higgins, L. (2000). A legal Retrieval Information System. *The Journal of Information, Law and Technology*. <http://elj.warwick.ac.uk/jilt/00-3/curran.html>
- Daniels, J. and Rissland, E. L. (1999). *Finding Legally Relevant passages in Case Opinions*. <http://citeseer.nj.nec.com/cache/papers/cs/15752.html>
- De Mantaras, R. and Plaza, E. (1994). Case-Based Reasoning: An Overview. *AI Communications*.10, 21-29 <http://eia.udg.es/~quimmel/varis/seminars-exit/cbr-seminar/material/lecture1/aicom2.pdf>
- De Mulder, R. V. and Combrink-Kuiters, C. J. (1996). Is Computer Capable of Interpreting Case Law? *The Journal of Information, Law and Technology*. <http://elj.warwick.ac.uk/elj/jilt/issue2/11demlar>
- Gelbart, D and Smith, J. C. (1991). Beyond Boolean Search: FLEXICON, A Legal Text-Based Intelligent System', in *Conference Proceedings of the Third International Conference on Artificial Intelligence and Law*, pp. 225--234. <http://www.iiia.csic.es/people/enric/AICom-ToC.html>
- Gelbart, D and Smith, J. C. (1992). Current Issues in Text Retrieval. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*. <http://www.iiia.csic.es/Reports/1996/ReviewCBR.ps>
- Gelbart, D and J. C. Smith (1999). Towards Combining Automated Text Retrieval and Case-Based Expert Legal Advice. In *Proceedings of International Conference on Database and Expert Systems Applications*, Vienna. <http://www.law.warwick.ac.uk/1tj/1-29.html>
- Getachew Haile. (1967). *The Problems of Amharic Writing System*. Unpublished
- Hafner C. D. and D. H. Berman (2000) *The Role of Context in Case-Based Legal Reasoning: Teleological, Temporal, and Procedural*. Boston: Northeastern University. <http://www.hafner.net/papers/think.ps>
- Harrison, I. (1997). *Case-based Reasoning*. <http://www.aiai.ed.ac.uk/links/cbr.html>
- Haque, et al. (1999). *Towards the Application of Case Based Reasoning to Decision-Making in Concurrent Product Development (Concurrent Engineering)*. Nottingham: University of Nottingham. <http://citeseer.nj.nec.com/cache/papers/cs/15752.html>
- Jurisica, I. (1995). *Representation and Management Issues for Case-Based Reasoning Systems*. Toronto: University of Toronto, Department of Computer Science.

- Kolodner, J. and Simpson, R. (1989). The MEDIATOR: Analysis of an Early Case-Based Program. *Cognitive Science*. 13:507-549.
- Kolodner, J. (1993) *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann
<http://www.iiia.csic.es/People/enric/AICom.html>
- Labor proclamation No. 42/1993
- Leake, D. B., ed. (1996). *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA: AAAI Press/MIT Press, Menlo Park, CA.
<http://www.cs.indiana.edu/hyplan/leake/papers/p-96-01.ps.z>
- Leake et al. 1997, Adaptive Similarity Assessment for Case-Based Explanation. *International Journal of Expert Systems*. 8:165-194.
- Leslau, W. (1965). *An Amharic Text Book of Everyday Usage*. University of California, Los Angeles.
- Osborn, J et al. (1999). Automated Concept Identification within Legal Cases. *The Journal of Information, Law and Technology*.
<http://www.elj.warwick.ac.uk/jilt/99-1/osborn.html>
- Osborne, J and Bridge, 1998,
- Pasha, M. and P. J. Soper (1999) *Combining the Strengths of Information Management Technologies to Meet the Needs of Legal Professionals*.
<http://elj.warwick.ac.uk/jilt/itpract/2pasha/2pasha.TXT>
- Proclamation No.25/1996. *Federal Negarit Gazeta*
- Riesbeck, C. K. and R. C. Schank (1989) *Inside Case-Based Reasoning*. New Jersey: Lawrence Erlbaum Associates Inc.
- Risland, E. L. and Ashley, K. D. (1987) A Case-Based System for Trade Secrets Law. *In Proceeding of the First International Conference on Artificial Intelligence and Law*
- Salton, G. and McGill, J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill Book Company.
- Schank, R (1982). *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge, England: Cambridge University Press

- Schjolberg, S. (2002). Judicial Decision Support Systems from a Judge's Perspective. *A Presentation at the Seventh National Court Technology Conference*. Baltimore, USA. <http://www.globalcourts.com/Foredrag-CTC7.htm>
- Statsky, W. P. (1998). *Legal Research and Writing: Some Starting Points*. Albany: International Thomson Publishing Company.
- Stefik, M (1995) *Introduction to Knowledge Systems*. San Francisco: Morgan Kaufmann Publishers.
- Sycara, K. P. (1987). *Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods*. Technical Report no. GIT-ICS-87/26
- Watson, I and F. Marir, (1994) *Case-Based Reasoning: A Review*. UK, Salford: University of Salford. <http://www.ai-cbr.org/classroom/cbr-review.html>
- Williamson, R. (1996) The Law Reports CD-ROM: A Case Study. *Journal of Information, Law and Technology*. <http://elj.warwick.ac.uk/jilt/BILETA/1996/3william/3william.txt>
- Yang, S. et al.(1994) Use of Case-Based Reasoning in the Domain of Building Regulations. *In Proceedings of the Second European Workshop on Case-Based Reasoning*. Chantilly, France. <http://fas.sfu.ca/cs/research/groups/CBR>

APPENDICES

Appendix I

The Amharic Character Set (Leslau, W.,1965).

Order							Labialized				
1 st	2 nd	3 rd	4 th	5 th	6 th	7 th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ l ^{wa}				
ሐ	ሑ	ሒ	ሓ	ሔ	ሐ	ሐ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ m ^{wa}				
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሪ	ር	ሮ	ሯ r ^{wa}				
ሰ	ሱ	ሲ	ሳ	ሴ	ሰ	ሶ	ሷ s ^{wa}				
ሸ	ሻ	ሼ	ሽ	ሾ	ሻ	ሿ	ሺ ሻ ሼ ሽ ሾ ሿ				
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ q ^{wä}	቉ q ^{wi}	ቊ q ^{wa}	ቋ q ^{wē}	ቌ q ^{wō}
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ b ^{wa}				
ተ	ቱ	ቲ	ታ	ቲ	ተ	ቶ	ቧ t ^{wa}				
ቸ	ቹ	ቺ	ቻ	ቼ	ቸ	ቻ	ቼ ሸ ሻ ሼ ሽ ሾ ሿ				
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ h ^{wä}	኉ h ^{wi}	ኊ h ^{wa}	ኋ h ^{wē}	ኌ h ^{wō}
ነ	ኑ	ኒ	ና	ኔ	ነ	ኖ	ኗ n ^{wa}				
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ	ኟ ሸ ሻ ሼ ሽ ሾ ሿ				
አ	አ	አ	አ	አ	አ	አ	አ ሰ አ ሰ አ ሰ አ ሰ				
ወ	ወ	ወ	ወ	ወ	ወ	ወ	ወ w ሠ wo				
ዐ	ዑ	ዒ	ዓ	ዔ	ዐ	ዑ	ዐ ሰ ዑ ሰ ዒ ሰ ዓ ሰ ዔ ሰ ዕ ሰ ዖ				
ከ	ከ	ከ	ከ	ከ	ከ	ከ	ከ k ^{wä}	ኩ k ^{wi}	ኪ k ^{wa}	ካ k ^{wē}	ኬ k ^{wō}
ኸ	ኸ	ኸ	ኸ	ኸ	ኸ	ኸ	ኸ h ሸ ሻ ሼ ሽ ሾ ሿ				
ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ	ዘ z ሠ zo				
ዠ	ዡ	ዢ	ዣ	ዤ	ዠ	ዡ	ዠ ሸ ሻ ሼ ሽ ሾ ሿ				
የ	የ	የ	የ	የ	የ	የ	የ y ሠ yo				
ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ g ^{wä}	ገ g ^{wi}	ገ g ^{wa}	ገ g ^{wē}	ገ g ^{wō}
ደ	ደ	ደ	ደ	ደ	ደ	ደ	ደ d ሠ do				
ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ	ጀ ሸ ሻ ሼ ሽ ሾ ሿ				
ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ	ጠ ሸ ሻ ሼ ሽ ሾ ሿ				
ጨ	ጨ	ጨ	ጨ	ጨ	ጨ	ጨ	ጨ ሸ ሻ ሼ ሽ ሾ ሿ				
ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ ሸ ሻ ሼ ሽ ሾ ሿ				
ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ ሸ ሻ ሼ ሽ ሾ ሿ				
ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ p ሠ po				
ፈ	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ	ፈ f ሠ fo				
ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ	ፐ ሸ ሻ ሼ ሽ ሾ ሿ				
ቨ	ቨ	ቨ	ቨ	ቨ	ቨ	ቨ	ቨ v ሠ vo				

Appendix II

Sample Form for Similarity Metric for the Different Values of the Attribute “gudayu”

	ደመወዝ	የስራ ስንብት ክፍያ	ስራ ላልተሰራበት ጊዜ ክፍያ	በስራ ላይ ለሚደርስ አደጋ ካሳ	የኸክምና አገልግሎት	ልዩ ልዩ የገንዘብ ክፍያዎች	ስለጥገኞች ክፍያ	የትርፍ ሰነት ክፍያ	እገዳ	የሰራተኞች ቅነሳ	ስራ ማቆም	የስራ መዘጋት	የእረፍት ጊዜ	ዝውውር
ደመወዝ														
የስራ ስንብት ክፍያ														
ስራ ላልተሰራበት ጊዜ ክፍያ														
በስራ ላይ ለሚደርስ አደጋ ካሳ														
የኸክምና አገልግሎት														
ልዩ ልዩ የገንዘብ ክፍያዎች														
ስለጥገኞች ክፍያ														
የትርፍ ሰነት ክፍያ														
እገዳ														
የሰራተኞች ቅነሳ														
ስራ ማቆም														
የስራ መዘጋት														
የእረፍት ጊዜ														
ዝውውር														