ANSWER LOCALIZATION SYSTEM
USING DISCOURSE EVALUATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERTER SUALP

IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

NOVEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Dr. Meltem Turhan Yöndem
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Göktürk Üçoluk (METU, CENG) ____________________________

Dr. Meltem Turhan Yöndem (METU, CENG) ____________________________

Assoc. Prof. Dr. Ali Doğru (METU, CENG) ____________________________

Assist. Prof. Dr. Bilge Say (METU, CENG) ____________________________

Dr. Onur Tolga Şehitoğlu (METU, CENG) ____________________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Merter Sualp

Signature :

ABSTRACT

ANSWER LOCALIZATION SYSTEM USING DISCOURSE EVALUATION

Sualp, Merter

M.S., Department of Computer Engineering

Supervisor: Dr. Meltem Turhan Yöndem

November 2004, 46 pages

The words in a language not only help us to construct the sentences but also contain some other features, which we usually underestimate. Each word relates itself to the remaining ones in some way. In our daily lives, we extensively use these relations in many areas, where question direction is also one of them.

In this work, it is investigated whether the relations between the words can be useful for question direction and an approach for question direction is presented. Besides, a tool is devised in the way of this approach for a course given in Turkish. The relations between the words are represented by a semantic network for nouns and verbs. By passing through the whole course material and using the relations meronymy for only nouns; synonymy, antonymy, hypernymy, coordinated words for both nouns and verbs; entailment and causality for only verbs, the semantic network, which is the backbone of the application, is constructed.

The end product of our research consists of three modules:

- getting the question from the user and constructing the set of words that are related to the words that make up the question
- scoring each course section by comparing the words of the question set and the words in the section
- presenting the sections that may contain the answer

The sections that are evaluated are taken as the sections of the course for granted.

The chat logs that expand three years of the course were taken by permission and questions were extracted from them. They were used for testing purposes of the constructed application.

Keywords: Turkish Answer Localization, Turkish Question Direction, Turkish Semantic Networks, Natural Language Processing

ÖZ

METİN BÖLÜMLERİNİN DEĞERLENDİRMESİ KULLANILARAK
CEVAP YERELLEŞTİRME SİSTEMİ

Sualp, Merter

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Dr. Meltem Turhan Yöndem

Kasım 2004, 46 sayfa

Bir dildeki kelimeler sadece cümle kurmada bize yardımcı olmazlar, ayrıca, genellikle küçümsediğimiz özellikleri da içerirler. Her bir kelime, kendisini diğer kelimlerle bir şekilde ilişkilendirir. Günlük yaşantımızda bu ilişkileri, soru yönlendirmenin de içinde bulunduğu, çok değişik alanlarda kapsamlı olarak kullanırız.

Bu çalışmada, kelimeler arasındaki ilişkilerin soru yönlendirmede faydalı olup olamayacağı incelenmiş ve soru yönlendirme için bir yaklaşım sunulmuştur. Bunun yanısıra, Türkçe verilen bir ders için, bu yöntem doğrultusunda bir araç geliştirilmiştir. Kelimeler arasındaki ilişkiler, isimler ve fiiller için bir anlamsal ağ ile ifade edilmiştir. Dersin tüm içeriğinin incelenmesi ve sadece isimler için üstanlamlık; isimler ile fiillerin her ikisi için de eş anlam, zıt anlam, altanlamlık, yanaşık sözcükler; ve sadece fiiller için de gerektirim ve neden ilişkileri kullanılarak, uygulamanın omurgası olan anlamsal ağ oluşturulmuştur.

Araştırmamızın son ürünü üç modül içermektedir:

- Sorunun kullanıcıdan alınması ve soruyu oluşturan kelimelerle ilişkili kelimelerden oluşan kümenin hazırlanması
- Dersin bölümlerindeki kelimeler ile soru kümesindeki kelimelerin karşılaştırılarak her bir ders bölümünün puanlanması
- Sorunun cevabını içermesi muhtemel bölümlerin sunulması

Değerlendirilen bölümler, doğrudan dersin bölümleri olarak alınmıştır.

Dersin üç yılını kapsayan konuşma kayıtları, izinle, alınmış ve içerdikleri sorular ayıklanmıştır. Bu sorular, hazırlanan uygulamanın testi amacıyla kullanılmıştır.

Anahtar Kelimeler: Türkçe Cevap Yerelleştirme, Türkçe Soru Yönlendirmesi, Türkçe Anlamsal Ağlar, Doğal Dil İşleme

Dedicated to my great family

ACKNOWLEDGMENTS

I am grateful to Meltem Turhan Yöndem for her full assistance and guidance throughout the research.

Also I wish to express my deepest feelings to my family for their continuous encouragement, understanding and motivating support.

I feel so lucky to have friends like Onur Soysal, İlhami Görgün and Mesut Bahadır. Their advices are highly appreciated.

I also would like to send special thanks to Narin Emeksiz. Without her, it would be really hard to complete this daunting task.

And lastly, many thanks go to my colleagues Gül Bayrambaş, Seval Erol and Yeşim Şimşek-Korkmaz. Their self-sacrifice was the most valuable contribution that could have taken place for this work.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATION viii

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTER

1. INTRODUCTION 1

2. LITERATURE SURVEY 4

   2.1. Question Answering Studies 4

   2.2. Semantic Network Construction Studies 7

3. BACKGROUND 9

   3.1. Question Direction 9

   3.2. Semantic Networks 10

      3.2.1. WordNet 1.7.1 11

   3.3. Evolutionary Algorithms 13

4. CONSTRUCTING THE INFRASTRUCTURE 14

   4.1. The File Conversions 14

   4.2. The Database 22

      4.2.1. Tables for Parsing the Question 26

5. THE IMPLEMENTATION OF QUESTION DIRECTION APPLICATION 27

   5.1. Input Module 27

   5.2. Evaluation Module 28

5.3. Output Module 29
6. WEIGHT SET GENERATION 30
6.1. Representation 30
6.2. Implementation of the Training 31
6.3. Discussion 34
7. CONCLUSION 37
REFERENCES 39
APPENDICES 41
A. 41

LIST OF TABLES

4.1 File (7) Format 19
4.2 File (10) Format 24
4.3 File (11) Format 24

6.1 Evaluating the Weights. 31
6.2 The Training Results 33
6.6 The Effects of Criteria 36

A.1 Synonymy Table 41
A.2 Antonymy Table 41
A.3 Meronymy Table 42
A.4 Entailment Table 42
A.5 Hypernymy Table 42
A.6 Causality Table 42
A.7 Coordinated Table 43
A.8 Relation Table 43
A.9 Section-Info Table 43
A.10 Word-Root Table 44
A.11 Lecture-Word Table 44
A.12 Lecture-Meaning Table 44
A.13 Question Table 45
A.14 Question-Answer Table 45

LIST OF FIGURES

3.1 A Sample Turkish Semantic Network 10

4.1 File Conversion Process: Steps 1- 6 18
4.2 File Conversion Process: Steps 6 - 9 21
4.3 File Conversion Process: Steps 10 -11 25

6.1 Fitness Curve 34

A.1 The Entity Relationship Diagram of the Database 46

CHAPTER 1

INTRODUCTION

Since the early days of computer science, the process of answering a question in an automatic way has always attracted the attention of computer scientists. Some tried to provide the exact answer to the user, while the others strived to retrieve the relevant documents from a collection of a larger document set. With the advent of the Internet and its unprecedented progress, the significance of directing a question to the related contexts has been massively increased.

Up until now, many approaches that are profoundly different than each other have been devised for the purpose of question answering and direction. The aim of this study is to investigate whether the relations between the words can contribute to the process of getting a question and searching for the pieces that may contain the answer. Our claim is that, for every word, we can construct a network of words which are connected to each other by semantically and this network can be used to evaluate the pieces for eligibility.

To prove our claim, first, we will devise a mechanism to generate the semantic network for a word, and then, try to construct a tool for getting a question and finding the pieces that may contain the answer to the question by using the semantic networks. Our application domain will be a certificate program of Middle East Technical University, which is called Bilgi Teknolojileri Sertifika Programı -BTSP- (Certificate Program of Information Technologies). This program includes eight online courses. These are:

- Bilgisayar Sistemleri ve Yapıları (Computer Systems and Structures)
- C ile Bilgisayar Programcılığına Giriş (Introduction to Computer Programming with C)
- C ile Veri Yapıları ve Algoritmalar (Structures and Algorithms with C)
- UNIX ile İşletim Sistemi (Operating Systems with UNIX)
- Yazılım Mühendisliği (Software Engineering)
- Veri Tabanı Yönetim Sistemleri (Database Management Systems)
- WEB Programlama (WEB Programming)
- Yazılım Geliştirme Projesi (Software Development Project)

All are about Computer Science. Moreover, all courses are in Turkish. Each course has its own web page. The lessons are given in eight weeks. Therefore, there are eight web pages for each course and sixty-four pages in total. However, we could only have the chance to deal with the following five of them:

- Bilgisayar Sistemleri ve Yapıları (Computer Systems and Structures)
- C ile Veri Yapıları ve Algoritmalar (Structres and Algorithms with C)
- UNIX ile İşletim Sistemi (Operating Systems with UNIX)
- Yazılım Mühendisliği (Software Engineering)
- Veri Tabanı Yönetim Sistemleri (Database Management Systems)

The output of our application will not be the EXACT answer to the questions. It only provides the sections that may contain the answer in an ordered fashion. The sections that are more likely to have the answer are shown first.

In choosing lectures in Turkish, two facts were considered: It is our native language and there is no such Turkish example in the Question Direction topic. It must always be kept in mind that the tool developed for this study strictly depends on the course material. Using for open-domain questions may produce defective results.

The other contribution of the tool is the additions to the Turkish semantic network. Most of the Computer Science specific words are added to the network, which still needs many more contributions.

The evaluation of the application is accomplished by using the questions that were asked by the instructors and students throughout the online chat sessions.

As our conclusions:

1. Expanding the queries by means of semantic networks increased the success of the search.

2. Since this study is domain dependent, adding the domain-specific words into account is inevitable for the success of the system.

The remaining of the thesis is arranged as follows: Literature survey and previous work are presented in Chapter 2. Chapter 3 includes the basic definitions of concepts that are used. The construction of the infrastructure of the application is presented in Chapter 4. Chapter 5 describes the details of semantic network building and the implementation of the application. Finally, Chapter 6 discusses the training of the application.

CHAPTER 2

LITERATURE REVIEW

Throughout this chapter, the researches that helped us in this work will be presented. Section 2.1 tells about the studies about Question Answering. The following section, Section 2.2, includes surveys Semantic Network Construction.

2.1 Question Answering Studies

Question Direction is actually Information Retrieval. It provides the documents that are related to the question; yet, it does not aim for providing the answer. On the other hand, most Question Answering systems [1], [2], [5], [6], [8], [9], [10] uses information retrieval techniques get the sections or documents that may contain the answer and apply detailed searches over them to extract the answers. The Question Answering studies are helpful in that sense.

The focus of all Question Answering Systems is getting the answer of a question from a large collection of documents. [3] The two important inputs of the Question Answering Systems are the question and the documents to be searched for the answer. If the document set remains the same, different questions will most probably generate different answers. Finding an answer to *a* given question can be either easy or difficult depending on the document collection. Hence, a healthy classification of Question

Answering Systems cannot depend on only one of those. Both of them should be thought as an inseparable unit.

Nonetheless, the Question Answering Systems can still be differentiated. Some of them strive to find a set of exact answer while the others struggle to find the passages that may contain the answer to the given question. In fact, the discrimination between the two approaches is quite important. The former is much more akin to Information Retrieval than the latter one, which is Document Retrieval.

Whatever the class, type or the way the Question Answering System follows, the success heavily depends on the ability of the system to combine:

- Application of complex Natural Language Processing techniques on questions, with
- Performing semantic unifications in lexical level on both the question and the collection of documents where the answer is searched

There are three areas that Question Answering Systems may differ:

- Linguistic and Knowledge Resources
- Natural Language and Document Processing Techniques
- Reasoning and Knowledge Coercion Methods

The differences in these three areas lead us to five different Question Answering System groups:

1. *Question Answering Systems for Factual Questions:* These systems look for the answers of the questions like “Where is the capital city of Turkey?” or “Who is the first president of the United States of the America?” The process for finding

out the answer heavily depends on the weight of the question keywords in the answer paragraph. No reasoning mechanism is necessary.

2. *Question Answering Systems with Simple Reasoning:* The question and the answer are not directly related. Some inference rules are applied to extract the answer.

3. *Question Answering Systems that can Search in Multiple Document Collections:* These systems gather partial information from different document collections and formulate an answer by applying advanced semantic rules on these information.

4. *Question Answering Systems with Analogical Reasoning:* These types of systems decompose the given question into multiple queries. The results of the queries are formulated into an answer by analogical reasoning

5. *Interactive Question Answering Systems:* Since the interaction is something like a dialog, the answer to a question depends both on the current question and the previous interaction with the user of the system.

Although Question Answering Systems vary this much, they have more or less the same architectural backbone: a *Question Processing Module*, a *Document Processing Module* and an *Answer Extraction and Formulation Module.*

The Question Processing Module is for determining the *expected answer type* of the given question. For example, suppose the question is “What is the capital of Turkey?” The expected answer type is a LOCATION. So, while searching through the document collection, the system indeed looks for LOCATIONs.

The Question Processing Module cannot directly derive the expected answer type by looking at the question word (in this example, the question word is “what”). The question words can be ambiguous. Therefore, other ambiguity resolution schemas are derived. *Dependency Model* is one of them.

In Dependency Model, there are rules for identifying the head-child of each constituent in a parse tree and then the headword is propagated to its parent. Then, *Answer Type Taxonomy* is used to find out the expected answer type of the given question.

After that, relevant paragraphs should be returned. To get the relevant paragraphs, the systems should build a query by using the question words and the expected answer type. The question words are all the nouns, their adjectival modifiers and the main verbs of the question. Yet, sometimes the basic query may return too much paragraphs (because there are a few words in the query) or too few paragraphs (because there are too many words in the query). By adding or dropping the words from the query, an optimization can be made. Moreover, for further enhancements, the following alternations can be applied to the words in the query:

- Morphological Alternations
- Lexical Alternations
- Semantic Alternations and Paraphrases

Parallel to question processing, the Document Processing Module should prepare the candidate sections or paragraphs from the document collection. This is accomplished by constructing paragraph-windows. A paragraph window is a set of sentences. The number of sentences in a paragraph window is fixed. After paragraph-windows are defined, each of them is evaluated and they are sorted according to the scores they have.

2.2 Semantic Network Construction Studies

The relations between the words cannot be used to effectively without designing a proper structure. There is an accepted method that benefits from WordNet for English.

[7] A word can be expanded to a set of words where each element of this set is acquired by applying predefined relations.

There is a database of words and relations for Turkish [11], [12] but it is not as complete as WordNet. For this work, many domain-specific words are added to that database.

CHAPTER 3

BACKGROUND

The ultimate goal of this study is to prove the claim that the relations between the words can be used in question direction. To establish a strong basis for the proof, Question Direction should be explained, the semantic relations should be identified and the computational structures representing these relations must be described.

For Question Direction, basic techniques designed for passage selection part of Question Answering are used. These could be found in Section 3.1. The semantic relations that are available to our purpose are detailed in Section 3.2.1. The structure of semantic relation representation, i.e. Semantic Networks, is explained in Section 3.2. The last section of this chapter, Section 3.3., mentions about the basics of Evolutionary Algorithms.

3.1 Question Direction

The core of Question Direction is twofold:

- getting the question form user and deriving necessary information from it
- using the derived information for evaluating the candidate sections

The information extracted from the question can be anything that will have an impact on the coming evaluation step. Syntactic and semantic information are the basic two types.

The same type of information gathered from the question must also be gathered from the place that hides the answer. This is because of the comparison purposes. To decide whether a section may contain the answer, the information conveyed from question and the information taken the section must be comparable.

The result of the comparison is not expected, or need not, to be a simple yes / no. The important part is that, the result should imply the eligibility of the section. Even this may not hold true for some systems. After every section is evaluated, the list of those sections, order by their values, may be enough.

Specifically for this work, the information derived from the question is a set of Semantic Networks. The elements of each network are compared with the words in a section and each comparison contributes to the overall value of that section. The end of the evaluation process produces a list of sections order by their values.

3.2 Semantic Networks

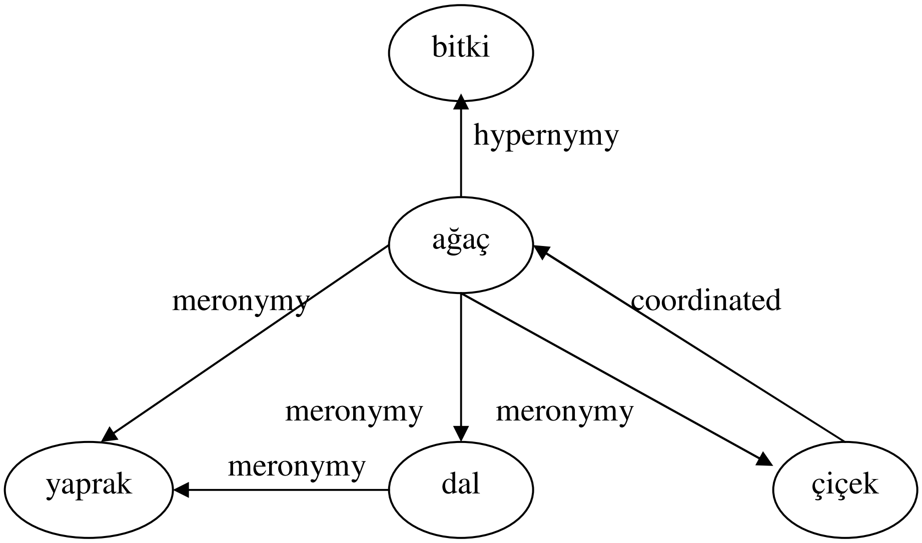

Figure 3.1: A Sample Turkish Semantic Network taken from [12]

Semantic networks are knowledge representation schemes. There exist:

- nodes
- links between the nodes

The nodes are the words and the links are relations between the words. Since not all relations are symmetric, the arguments of a relation are to be differentiated. Therefore, for asymmetric relations, the links are directed. An example semantic network is shown in Figure 3.1. In this network, {yaprak} is a meronymy of {dal}. No line means there is no semantic relation.

For the purpose of this work, Semantic Networks serve as a way to expand the question. Each word in the question corresponds to a Semantic Network. Suppose question is a set Q and $Q = (w_1, w_2, \ldots, w_n)$ where w_i are the words in the question. Then

$$S(w_i) = \bigcup \mathrm{Sr(wi)} \text{ where } w_i \in Q$$

For each relation r, S_r is the set of elements that w_i is related with relation r.

3.2.1 WordNet 1.7.1

Combination of current psycholinguistic theories of human lexical memory and computer technology gave birth to an application that enables to map the English nouns, verbs, adjectives, adverbs and function words into their underlying lexical categories and represent their interactions: WordNet.

The meanings can be thought as synonym sets. In other words, each synonymy word is treated as one set. Each synonym set is related to other synonym sets in different ways. What WordNet does is representing these sets and relations. The relations are the pointers between the synonym sets. Some of these relations are symmetric while some others are not. There may be many relations to be discussed; yet the scope of this thesis limited us to use the only ones explained below.

Synonymy To construct the synonymy sets, we need the basic relation, synonymy, to gather the words having the same meaning as one set. One of the definitions of synonymy states that, *two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made.*

Antonymy The antonymy of a word is the opposite of that word. For example, {good} and {bad} are antonyms. This relation is symmetric.

Hypernymy Hyponymy / hypernymy is a semantic relation between word meanings. It may be thought as Subset / Superset relation. Hyponymy is Subset and Hypernymy is Superset : e.g., {dog} is a hypernymy of {hound}, and {animal} is a hyponymy of {dog} These relations are asymmetric.

Meronymy This is the part-whole relation. The lexical semanticists name them as meronymy-holonymy. The meronymic relation is transitive and asymmetric. For example meronymy of {tree} is {branch}.

Entailment When a verb entails the other, we get this as the entailment relation. For example, {snore} entails {sleep} because the sentence *He is snoring* entails *He is sleeping*. This relation is asymmetric and available only for verbs.

Cause A resultative action is in fact caused by some other action. For example, because I show, you see. Hence, there is a causality relation between see and show. This relation is asymmetric and available only to verbs.

Coordinated Words WordNet is not a dictionary. We cannot find the meanings of words there. But *see also* and *gloss* parts of dictionaries are preserved thanks to this relation. We know that there is a lexical relation between {doctor} and {nurse}. This relation is symmetric.

3.3 Evolutionary Algorithms

The Evolutionary Algorithms imitate nature. They take a handful of parents, breed them and produce the offspring. Over time, the values generated by the algorithm converge to a point.

Throughout this process, Evolutionary Algorithms use two basic operations: cross-over and mutation. The key in the genetic variation is cross-over. While preserving the qualities of both parents, it equips the child with new properties.

Every once in a while, randomly, some part of a chromosome changes itself, which is called mutation. This enhances genetic variation and employs momentum to the generation. The momentum can be in either way, positive or negative.

In this work, the evolutionary algorithms are used for training the system to construct a set of variables that will be used in the evaluation of the discourse segments. The set consists of 9 variables. The exact values for those variables are not known. By using the evolutionary algorithms, their values will be generated.

CHAPTER 4

CONSTRUCTING THE INFRASTRUCTURE

In this chapter, the database that is used to construct the Semantic Networks and evaluate the sections is to be presented. The Section 4.1 gives detailed information on the file conversion processes. The next section, Section 4.2, mentions about the database used by the tool.

4.1 The File Conversions

The courses of BTSP were taken from IDE_A. There are eight online courses; however, the notes of five of them are available for this work. These lessons are all in HTML (Hypertext Markup Language) format.

On the other hand, a basic Turkish Semantic Network Database [12] is obtained. It is intended to be a Turkish version of WordNet. In this database, there are only Turkish verbs and nouns. It consists of seven different, but interrelated, files. The backbone of them is the file that contains the synonym sets. Each word in this file has both a meaning number and a sense number. For instance:

```
1       1       ab
1       2       su
```

is a synonym set taken from the database. It says that “su” or “ab” has the meaning number 1. They are synonyms of each other. First synonym of the first meaning is “ab” and second synonym of the first meaning is “su”.

There are 12077 words in the synonym set file. The other files are:

- Antonymy set file. There are 991 relations.The format of the antonym set file is:

```
1033    2       813     1
```

First and second columns represent a word in the synonym set file and the remaining two columns represent another word. Only meaning number is not sufficient. For example:

ak	X	kara	ANTONYMS
beyaz	X	kara	NOT ANTONYMS

- Meronymy set file. There are 1567. The format of the antonym set file is:

```
1       1202
```

This shows us that meaning 1 is a part of meaning 1202.

- Coordinated words set file. There are 24142 relations. The format of the coordinated words set file is:

```
8869    8868
```

This shows us that meaning 8869 and meaning 8868 are coordianted.

- Hypernymy set file. There are 4823 relations. The format of the hypernymy set file is as:

```
1       1182
```

This shows us that meaning 1182 is a super set of meaning 1.

- Causality set file. There are 130 relations. The format of the causality set file is:

```
1566    1470
```

This shows us that meaning 1566 causes meaning 1470 to occur.

- Entailment set file. There are 96 relations. The format of the entailment set file is:

```
1443    2502
```

This shows us that meaning 1443 entails meaning 2502.

There are a few issues to be addressed about the database that we are given. First, it did not contain the words that are specific to computer science. Second, the database is a collection of seven text files. Because of the performance concerns, these should be converted into tables. And lastly, there may be some inconsistencies in these files, since they are all manually created, maintained, and no consistency check is made.

The work presented here is domain specific. Therefore, storing only the words that are computer science specific will be enough. The structure of these files preserved, but only the words that are related to our domain, which may be called as *target words*, and their relations are inserted into the database.

After addressing these problems, the *target words* must be listed. The roots of the words that appear in the lecture notes are extracted. The Turkish Semantic Network Files contain only the roots of the words; therefore, to make correct comparison, the roots of all the words that constitute the lecture notes must be present. The following steps are taken accordingly.

It may well be too hard to follow only by reading; therefore, the graphical visualization of the steps throughout the process will be available (Figures 4.1-4.3).

The name of the files and their usage are shown in the figures. Their representative numbers reside inside the circles that are left of the boxes. Referring to the files with their given numbers will be easier. The arrows show the conversions. The explanation, if there are any, next to the arrows gives information about the conversion task.

HTML formatted lecture notes (1) are converted into text format (2). The conversion is held by selecting the all text in the web page and pasting them all into Notepad. All images and other stuff that are not related to the lecture itself are deleted, which lead to another set of text files (3). Every lecture had eight HTML files, each, actually contains the notes for a week. Hence, after the conversion, forty text files are prepared.

With these, the manual process is over, but one more step is needed. By using a PHP code, previously built text files are converted into another format (4). The content of the newborn files are exactly the same with the original text files, with one crucial difference: in the new files, there is only one word in each line. In other words, the lecture notes are separated word by word. By passing over this file, one more type of a file (5) is created. This had basically the same format with the aforementioned file, yet there is one slight difference. The sentences are numbered. Each file, again, represented a week of a lecture.

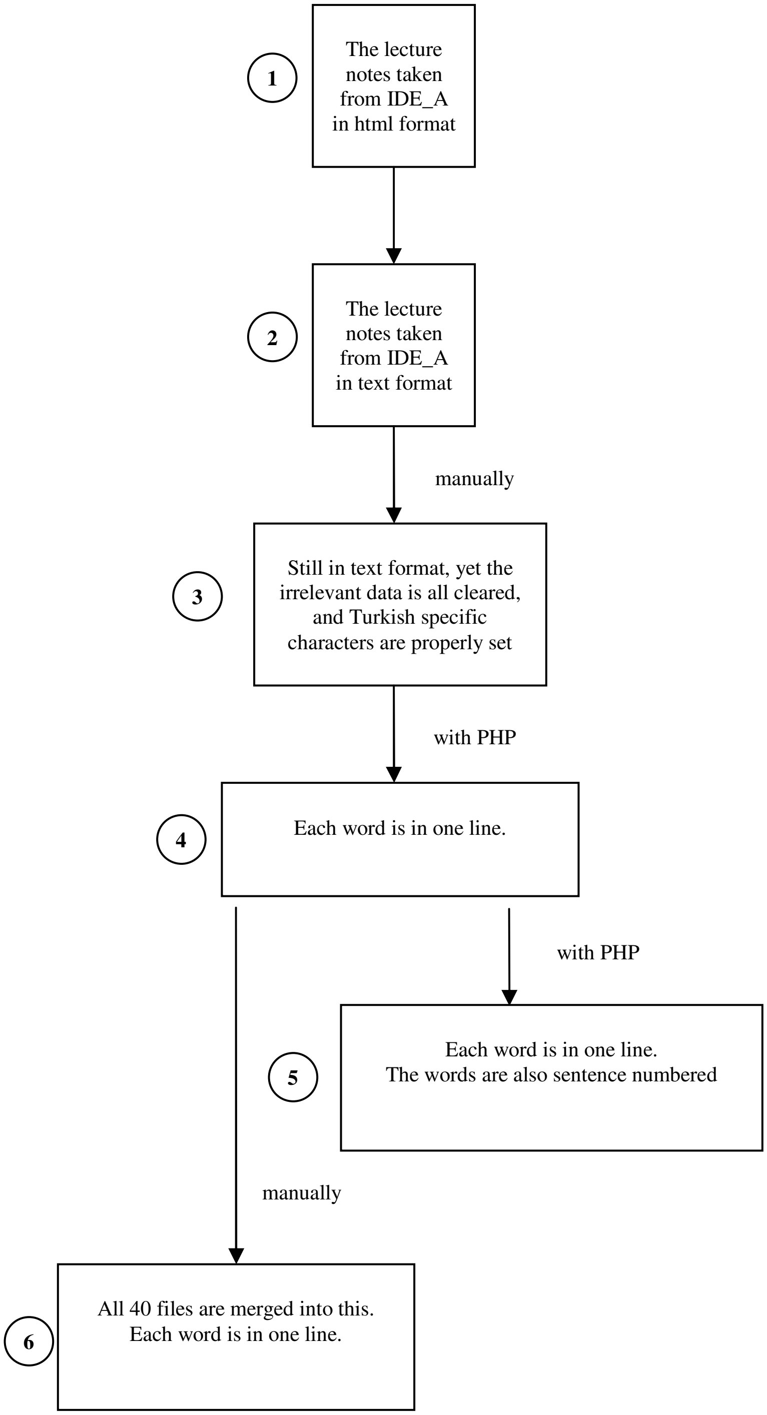
1
The lecture notes taken from IDE_A in html format
2
The lecture notes taken from IDE_A in text format
manually
3
Still in text format, yet the irrelevant data is all cleared, and Turkish specific characters are properly set
with PHP
4
Each word is in one line.
with PHP
5
Each word is in one line.
The words are also sentence numbered
manually
6
All 40 files are merged into this.
Each word is in one line.

Figure 4.1: File Conversion Process: Steps 1-6

The goal of these file conversions is to extract the words that had appeared in the lecture notes but not in the database. To search the lecture note words in the database the roots of the words should be identified; since, in the database, there are only roots. For example, suppose the word “dersler” is in the lecture notes. We are searching for the word “dersler” is futile in the database because “dersler” is actually “ders + Plural (lAr)”

To have the roots of all those words, all files (4) are merged into one file (6) and this is sent, which contained all the words in lecture notes line by line, to be parsed by the tool devised in [5]. By the help of appreciated efforts, all words are turned into “roots + suffixes” form. A sample output from the output file (7) is given below:

Table 4.1: File (7) Format

```
Giriş giriş +Verb+Pos+Imp+A2sg
Giriş gir   +Verb+Pos^DB+Noun+Inf+A3sg+Pnon+Nom
Giriş gir   +Verb^DB+Verb+Recip+Pos+Imp+A2sg

Tarihin     tarihi      +Adj^DB+Noun+Zero+A3sg+P2sg+Nom
Tarihin     tarih +Noun+A3sg+Pnon+Gen
Tarihin     tarih +Noun+A3sg+P2sg+Nom

başlangıcından    başlangıç   +Noun+A3sg+P2sg+Abl
başlangıcından    başlangıç   +Noun+A3sg+P3sg+Abl

beri  beri  +Noun+A3sg+Pnon+Dat
beri  beri  +Noun+A3sg+Pnon+Nom
beri  beri  +Postp+PCAbl
beri  beri  +Adv

insanoğlu   insanoğul   +Noun+A3sg+P3sg+Nom

bilgi bilgi +Noun+A3sg+Pnon+Nom
```

This output contains the word – root – suffix information. From this file, verbs and nouns are picked. These nouns and verbs are searched in the Semantic Network Database and the absent ones are saved for the next stage.

There are also *UNKNOWN* words in this file. This means that there is no available word-root-suffix information for those words. These are mainly non-Turkish words or abbreviations. Our observations have shown that nearly all of these words are related to our domain and they are all designated to be the *target words*.

File (7) is too big to process in one pass. Therefore, it is divided lecture by lecture and then week by week. The format is not changed (8). However, as mentioned before, only the derivations of verbs and nouns are suitable for our purposes, so the words having root types other than those are left off. The Turkish specific characters of the remaining ones are converted in to their capital Latin counterparts.

Before talking about our database and the tables within, seeing the overview of the process and the input / output files after the parser in [5] took the job with visual aids will help a lot.

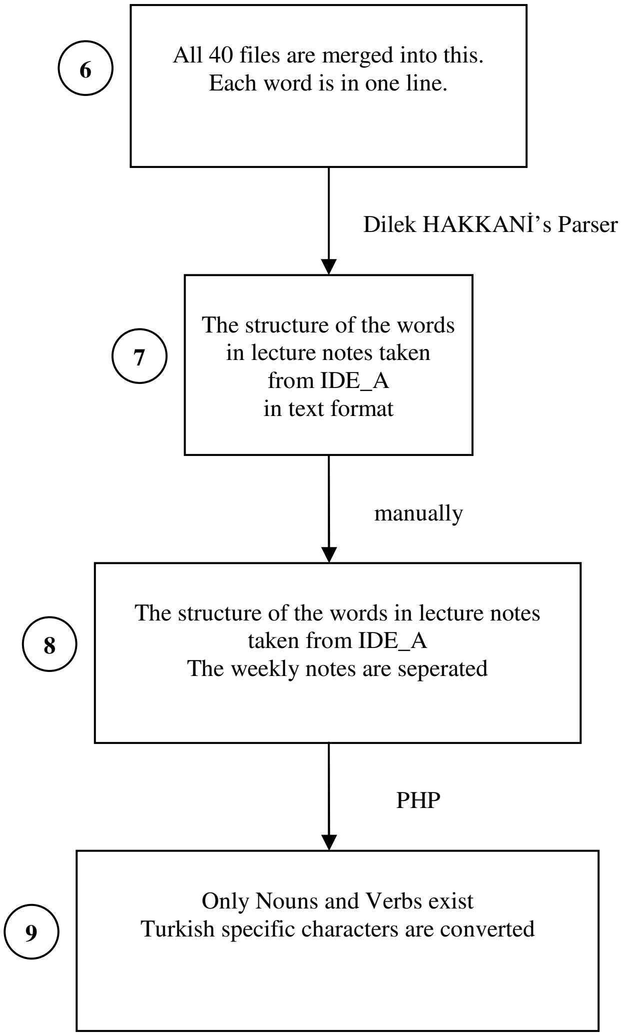
6
All 40 files are merged into this.
Each word is in one line.
Dilek HAKKANİ's Parser
7
The structure of the words
in lecture notes taken
from IDE_A
in text format
manually
8
The structure of the words in lecture notes
taken from IDE_A
The weekly notes are seperated
PHP
9
Only Nouns and Verbs exist
Turkish specific characters are converted

Figure 4.2: File Conversion Process: Steps 6 - 9

4.2. The Database

The structure of the seven files that contain the words and their interrelations are designed to be the basis of a discourse segmentation tool [12]. They are, and still are, invaluable sources for any project that will use Turkish Semantic Networks. Yet, it is clear that file processing would take huge amounts of time. As the design progressed, it is seen that complex data types are thought to be much handier. Moreover, there would be times that, a single step must access more than one data source (which are all files at the beginning).

Directly using files as main data storage would result long waiting times. The end product application is never tested with using the files; however, while reading the records from files and putting them into tables, we have seen that our concerns are true; since accessing files and converting them into tables took our precious time a lot.

As the first step, the *target words* are extracted and their relations are constructed as in the seven text files. They are put directly into tables that consist of the identical columns. The tables are in Appendix A. Each table is designated for one word relation. SYN is for synonymy, ANT for antonymy, MER for meronymy, ENT for entailment, HYP for hypernymy and hyponymy, CAU for causality and COO for coordinated words. In the tables MER, HYP and CAU, the order is important. For example in ENT, en_word_no1 entails en_word_no2. In MER, me_word_no1 is meronymy of me_word_no2, and in HYP, hy_word_no1 is the hypernymy of hy_word_no2. These are all the relations that would be covered.

Other than those tables RELATION, SECTIONINFO, WORDROOT, LECTUREWORD, LECTUREMEANING, QUESTION and QUESTIONANSWER tables are created for storing information about the question, the lecture notes and for comparison and evaluation purposes.

In RELATION table, il_relation_no both serves as uniqueness purposes and gives us the opportunity to process the relations within a single loop. il_relation_adi is for describing the relation's full name. In the field of il_relation_table_name, we store the table name where the relation is represented. SYN, ANT, COO, CAU, MER, HYP and ENT are the only values that will occur in this field. il_relation_code is actually used only for one type of relation: Exactness. This relation has no table. However, we should treat it as a relation, since we will not only look for the related words but also to the exact words that constitute the question. When exact matches are found, we identify them as the words that have the Exactness relation with the question words. The field il_value is the home for the numerical value for the relation. The values obtained in previous works [12] are used. In the section evaluation process, this field is used extensively. All the data in the table RELATION are inserted into the table by hand. There is no automatic task for this step.

There is one more step to be explained before going further. Since the ultimate purpose is to present the sections that most probably contain the answer to a given questions, there must be identified sections. We chose the natural or pre-constructed sections as our sections. These are simply the parts that are set by the lecturers themselves. We have neither divided nor deleted them. They are taken granted as they are. No modification is made. Therefore, the process of preparing and inserting data about discourse segmentation is made manually. Both the lectures in html format (1) and the text files with sentence numbers (5) are passed through.

Table 4.2: File (10) Format

1	284	Baslik1	Link1	0
2	311	Baslik2	Link2	1

First column is the number of the section. The second column is the last line number in that section (inclusive). The last number at the end of the line is the number of the super section. If a section is a subsection of another section, then this field contains the number of the super section. Else it has 0.

The lecture notes do not have those section links inherently. The links are added to the HTML pages by modifying the tags, not the content of the courses.

By means of these files, the opportunity for easily constructing the next file (11) is gained.

Table 4.3: File (11) Format

1	1	giriS
1	1	tarihin
1	1	baSlangIcIndan
1	1	beri
1	1	insanoGlu
1	1	bilgi

The first column is the section number, followed by the sentence number and the word in the lecture notes. There are a few points deserving to talk about. One of them is that there are no punctuation marks in this file. All of them are removed. As the other one, all capital letters are converted into lowercase letters except the Turkish specific letters. These Turkish specific letters are converted into their capital Latin counterparts. So, only capital letters in a word are the Turkish specific characters. Then, all words that contain only numbers are eliminated. And the last, but perhaps the most important note is that, the suffixes that are separated by "'" are recorded as separate entries.

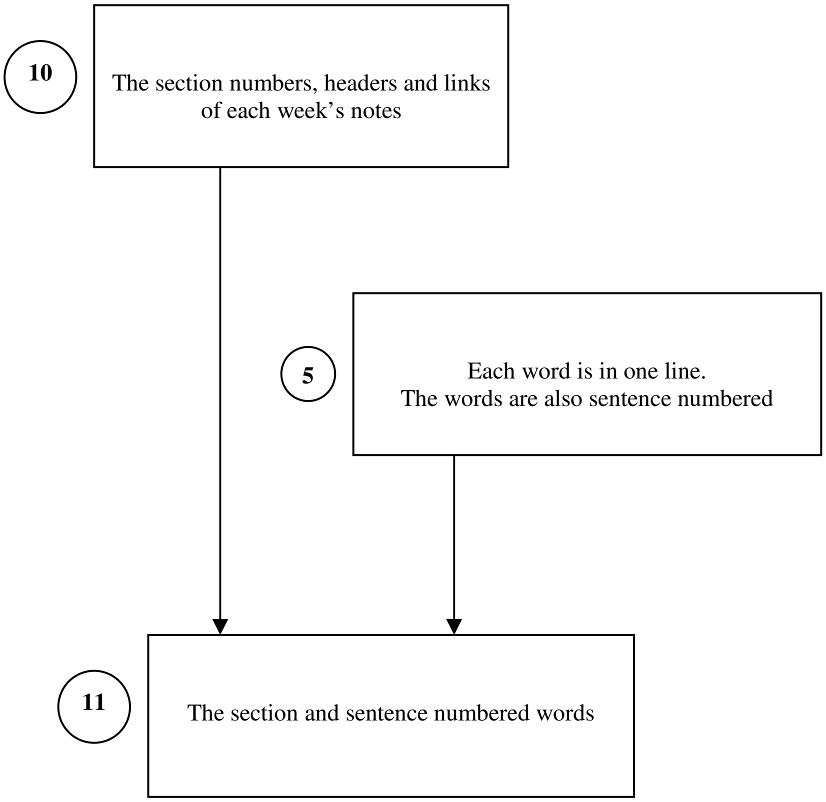
10
The section numbers, headers and links of each week's notes
5
Each word is in one line.
The words are also sentence numbered
11
The section and sentence numbered words

Figure 4.3: File Conversion Process: Steps 10 - 11

While dividing the lectures into sections, the general information contained by the sections must be stored. The table SECTIONINFO keeps these. To prepare the table, two different approaches can be followed. One of them is getting the data by means of a user interface, as in the way the table RELATION is constructed. The other one is getting the same data from a file. In fact, one such file already exists. The format of the files (10) serves both for the previous purpose, which is giving the correct section number to the sentence numbered words, and for supplying the necessary information to SECTIONINFO table.

4.2.1 Tables for Parsing the Question

To use the relations and relation tables, the question sentence should be parsed to get the individual words and extract the roots of each word. These are the information needed for the section evaluation step. The only parsing mechanism we, indirectly, had is in [5]. It is used to parse the whole course material; however, it is not available for our question parsing purpose. We may well write our own morphological parser. However, that would put us too far away from our goal.

It is clear that we cannot use the parser in [5] for sure. But, the input and the output of that parser for the whole course material are readily available as files (6) and (7) respectively. The words that constitute the questions to be asked would likely use these words. Therefore, these word-root pairs in (9) files could be used. WORDROOT table is designed as a look-up table. All word-root pairs and their structures that exist in the output files (9) are stored in the WORDROOT table.

The lecture notes are transformed into two tables for efficient search. First of them is the LECTUREWORD table. It consists of all the section and sentence numbered data in the files (11). The words that constitute the whole course material are stored in LECTUREWORD table. However, each word does not necessarily correspond to a meaning. There may well be concepts that are represented by more than one word. For example access/collision (erişim/çarpışma) identifies a single meaning where two words constitute it. This type of word chunks are broken into their constituents and the consecutive constituents are looked for through the lecture notes. If these structures are found, they are given their respective sense numbers and are stored in another table, LECTUREMEANING. The one word – one meaning entries are also kept and moved to the LECTUREMEANING table.

CHAPTER 5

THE IMPLEMENTATION OF QUESTION DIRECTION APPLICATION

This chapter is intended to give the details of the implementation of the tool. This application consists of four interconnected modules. Each of them heavily depend the one that comes prior to it. The modules are:

- Input Module, described in Section 5.1
- Evaluation Module, described in Section 5.2
- Output Module, described in Section 5.4

The application, with slight modifications, is also used for adjusting the weights of the relations.

5.1 Input Module

This gets the question from the user. Then removes all punctuation marks (".", "?", "!", ",", "\"", """, "(", ")", ":", ";", "{", "}", "[", "]") from the question and creates a *Sentence (Cümle)* object. Each word in the question will be become a *Word (Kelime)* object and be a member of *Sentence* object. The *Sentence* is a list of *Word* objects. Each *Word* object consists of the word itself and the number of times that word is encountered in the question.

But before becoming a *Word*, the letters in the word must be converted into lowercase and the Turkish specific characters should also be converted into their capital counterparts.

5.2 Evaluation Module

The aim of the evaluation module after getting the *Sentence* object as the input is to extract the necessary information from the *Word* objects and apply three different functions to each of the sections. This will result in an ordered list of lecture sections. While evaluating the affect of the meaning relations on Question Direction, the contributions of two other criteria are also investigated. These criteria are:

- The number of exact matches in a section
- The number of irrelevant question words in a section

The size of the section is also an important criterion. However, the size of the sections in our domain varies greatly. The effect of the section size criterion highly biases the overall evaluation step. Henceforth, the section size is left out of the discourse evaluation process.

As the first step, the number of exact matches is found. The words that make up the question and the words that form the lecture notes are directly compared. This comparison is accomplished by means of the LECTUREWORD table. An SQL query is generated and it is used to fetch the number of exact matches in each section. The number of matches is multiplied by the appropriate weight and added to the total value of the section.

In the second step, the words that constitute the question sentence are taken one by one and their roots are extracted. The root of the word is being searched as follows: first, the WORDROOT table is looked at whether the word exists as a root. If so, then it is concluded that the word is the root of itself. Otherwise, it is tried to be verified whether the word exists as a word in WORDROOT table. If it does exist, it is very likely that there is more than one root-structure for the word to be found, which implies that the word may have more than one root. Hence, a choice must be made. Since this work is domain specific, most of the time there will only be one root for each word. Moreover, the syntactic information of the question or the lecture notes is not available. These leads us to the conclusion that morph-sense disambiguation is not applicable, so the longest root is chosen as the root of the word. The other option for finding the root of the word is to look it up in the SYN table. The SYN table is the last place for getting the root of a word.

For each word in the question, a list of other words that are related to the word is constructed. The process of finding the related words, the seven relation tables (SYN, ANT, MER, ENT, HYP, COO, CAU) should be used. The root of the word is searched and corresponding related words are added to a list. After every related word is placed into that list, all of them are searched through the structure constructed by means of the LECTUREMEANING table. The section information list for each and every one of meaning is built and the related words are searched in these lists.

The other function identifies whether the question words have any impact on the evaluation. If a question word does not change the value of a section, then it is considered to be irrelevant of the question – section search and treated to have an affect on the value of the section.

5.3 Output Module

Not only is the question provided by the user but also the number of sections to be seen is given. All sections will be evaluated according to the question. They are ordered by their values in descending order. The links of the best *X* of them will be presented to the user where *X* is the number of sections to be dealt with.

CHAPTER 6

WEIGHT SET GENERATION

This chapter is about the training of the weights that are used in the Evaluation Module of the Question Direction Application. There are 9 weights to be fine tuned. Evolutionary Algorithms are used in this tuning operation. Section 6.1 describes the representation of the weights. The next section, Section 6.2, gives information about the details of the implementation. Finally, Section 6.3 will present the results after the training.

6.1. Representation

The weights are set to be between –100 (exclusive) and 100 (exclusive). 8-bits are assigned for each weight. A set of 9 weights consists of 72 bits and thought to be a “chromosome”. Crossovers and Mutations are applied on this 72-bit array.

The chat logs of the courses ae taken by permission. The logs of the course BSY (Bilgisayar Sistemleri ve Yapıları) for the years 2001, 2002 and 2003 are extensively searched and 100 questions are identified for training purposes. These questions are stored in the QUESTION table. The sections that contain the answer for the questions are also selected and they are inserted into the QUESTIONANSWER table. This table is used for comparison of the answer sections produced by the Question Direction Application.

6.2 Implementation of the Training

There are 100 parents to be trained. 99 parents are created randomly. The relation weights of one of them is preset as the weigths taken from[12]. The weight of Number of Exact Matches is 0 and the weight of Number of Irregular Words is -5.46875. These 100 parents are the set of weights and for each set, the Question Direction Application is executed to generate the answer sections for 100 questions. For training issues, the number of sections to be generated is set to be 3. The generated answer sections and the answer sections that are considered to be true in QUESTIONANSWER are compared and the result of the comparisons are evaluated as follows:

Table 6.1 Evaluating the Weights

- If the generated answer section and its order are both true, a +10 point is given
- If the generated answer section is sub or super section of the real answer section and their orders are the same, a +8 point is given
- If the generated answer section is true; but its order is wrong, a +10 point is given
- If the generated answer section is sub or super section of the real answer section and their orders are different, a +8 point is given
- If the generated answer does not exist as a real answer; but it is in the same lecture and in the same week of some answer section, a –5 point is given
- If the generated answer does not exist as a real answer; but it is in the same lecture of some answer section, a –7 point is given
- If the generated answer is totally different than the real answer sections, a –15 point is given

The first 50 parents getting the best results are paired and bred. The pairing operation is handled tournament-wise. I.e. a parent is chosen randomly. Then another one is chosen. One of them will be selected for the next step of the tournament. The number of steps of the tournament is 4. The selection in one step depends on the points of the competetors. 80% of the time, the parent with better result will be selected.

After chosing one of the parents, the other one is chosen in a same way. As the next step, they are bred. A random point inside 88-bit chromose is selected and the bits after the selection point are interchanged between the parents. While producing the new children, there exists a possibility of mutation. The mutation ratio is defined to be 1 / 1,000. The mutation is simply changing the value of the bit.

The 50 parents that has the least scores are repleace by the new born children, which are the new set of weights. This execute - evaluate – breed – replace sequence is repeated 50 times for tuning the weights as good as possible.
After training the application, the following results are optained:

Table 6.2 The Training Results

Name of the Weight	**Value**
Exactness	4.7244094
Synonymy	57.4803150
Antonymy	0.7874016
Hypernymy	-7.8740157
Meronymy	-6.2992126
Causality	49.6062992
Entailment	-62.2047244
Coordinated	1.5748031
Number of Irrelevant Words	-100.0000000

Throughout the training process, the following fitness curve is obtained. Here, the average of fitness is calculated by summing up all the fitness values of each parent for that iteration and dividing it by the number of parents, 100.

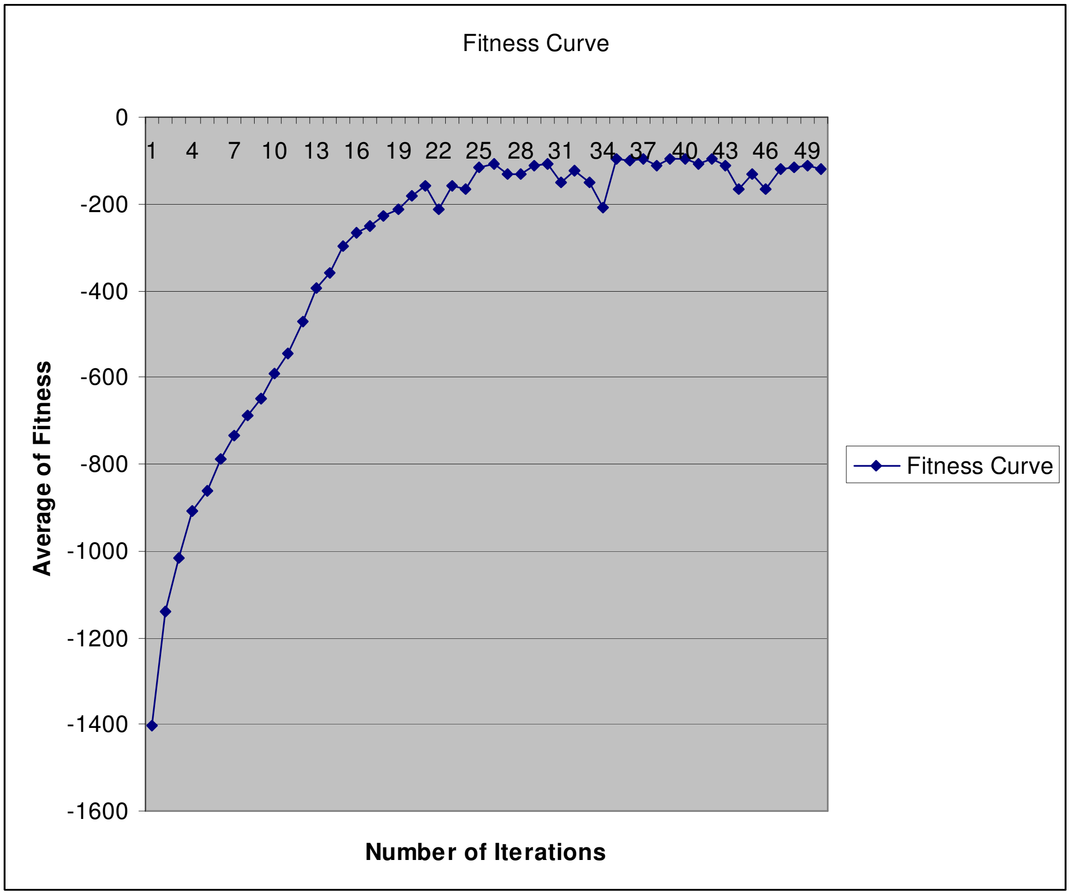

Figure 6.1 Fitness Curve

6.3 Discussion

There are 175 sections in BSY lecture. In exploring the baseline performance, first, the probability of finding a correct answer to one question is calculated. It is defined as:

$$1 - ((172 / 175) * (171 / 174) * (170 / 173))$$

The basic idea behind it is that the probability finding all false sections is subtracted from 1. The result is 5.08%. Since answering each question is independent of each other, it can be deduced that for answering half of the questions correctly for example, 0.00508 will be multiplied by itself by 50 times and the result will be multiplied by $\begin{pmatrix}100\\50\end{pmatrix}$. In this example, this calculation will yield to 1.98e-86

For the selected 100 questions, 198 sections were chosen as the sections that ontain the answer. After running the application, 300 candidate sections are selected.

The success of the system is defined by using the number of questions correctly answered. Moreover, it is also important to identify the results that are not correct but are in the same lecture and in the same week, because they are all in one html page. This type of results give clues about the accuracy of the system. They are considered to be partially correct answers and contribute to the overall success.

In the following table, the results are detailed. There are 9 criteria in evaluating the sections. After training the system, the success of the system is observed. To investigate the effect of each criterion, each criterion is set to 0 one at a time and all 100 questions are answered again. The results are shown in the rows of the following table.

Table 6.3 The Effects of Criteria

MISSING CRITERION	**SUCCESSFUL ANSWERS (Over 100 Questions)**	**PARTIAL SUCCESSFUL ANSWERS (Over 100 Questions)**	**OVERALL (Over 100 Questions)**	**FITNESS VALUE**	**BASELINE PERFORMANCE**
NONE	62	11	73	-98	3.29e-115
Number of Exact Matches	51	14	65	-308	9.87e-89
Synonymy	54	12	66	-447	9.61e-96
Antonymy	62	11	73	-98	3.29e-115
Hypernymy	66	12	78	-26	2.24e-125
Meronymy	62	11	73	-98	3.29e-115
Causality	62	11	73	-98	3.29e-115
Entailment	62	11	73	-98	3.29e-115
Coordinated Words	57	12	69	-294	6.54e-103
Irrelevant Word Count	50	7	57	-897	1.98e-86

CHAPTER 7

CONCLUSION

The aim of this work is to investigate the effect of the relations between the words on question direction. Moreover, the number of exact matches, irrelevant words and the size of the sections. Throughout the research and implementation, it is seen that the relationships between the words, the number of exact matches, the number of irrelevant words and the size of the sections play a significant role in question direction.

The effect of verbs seems to be negligible. This can be derived from the fact that the absence of causality or entailment does not have an effect on the success of the system. There is no meronymy relation between the target words. This is reflected by the results obtained by leaving the meronomy relation out. There are 31 antonymy relations. The results show that they are also not able to alter the sections chosen to contain the answers.

The coordinated words, synonyms and exact matches contributes to the section selection; however, the absence of irrelevant word count deeply lowers the success ratio.

The last result that should be pointed out is that the inclusion of hypernymy relation degrades the precision of section selection.

For improving the accuracy there are three steps to be considered:

1. The set of questions should be enlarged.
2. The weight evaluation mechanism presented in table 6.1 can be reevaluated. It should be revised and some other properties may be added.

REFERENCES

[1] "Question Answering by Passage Selection (MultiText Experiments for TREC-9)", http://citeseer.ist.psu.edu/553723.html, 27.12.2004

[2] "Exploiting Redundancy in Question Answering", http://citeseer.ist.psu.edu/clarke01exploiting.html, 27.12.2004

[3] Harabagiu, M., Moldovan, D., "Question Answering", Oxford Handbook of Computational Linguistics, Chapter 31

[4] Hakkani-Tür, D. Z., "Statistical Language Modeling for Agglutinative Languages". Ph.D. Thesis, Department of Computer Engineering, Bilkent University, August 2000, Ankara, Turkey

[5] "IBM's Statistical Question Answering System", http://citeseer.ist.psu.edu/636512.html, 27.12.2004

[6] "Tequesta: The University of Amsterdam's Textual Question Answering System", http://citeseer.ist.psu.edu/monz01tequesta.html, 27.12.2004

[7] "An Analysis of Ontology-based Query Expansion Strategies", http://www.dcs.shef.ac.uk/~fabio/ATEM03/navigli-ecml03-atem.pdf, 27.12.2004

[8] "High Performance Question / Answering", http://citeseer.ist.psu.edu/pasca01high.html, 27.12.2004

[9] "Ranking Suspected Answers to Natural Language Questions Using Predictive Annotation", http://citeseer.ist.psu.edu/radev00ranking.html, 27.12.2004

[10] "A Question Answering System Supported by Information Extraction", http://citeseer.ist.psu.edu/srihari00question.html, 27.12.2004

[11] Turhan-Yöndem, M., "An Analysis on Turkish Discourse Segmentation" In Proceedings of ESSLLI 2000, pp 265-274, 2000.

[12] Turhan-Yöndem, M., "Identifying the Interactions of Multi-Criteria in Turkish Discourse Segmentation". PhD Thesis, METU, 2001.

APPENDIX A

Table A.1: Synonymy Table

SYN (1423 records)

sy_word_no [int(11)]
sy_sense_no [smallint(6)]
sy_word [char(50)]

Table A.2: Antonymy Table

ANT (31 records)

an_word_no1 [int(11)]
an_sense_no1 [smallint(6)]
an_word_no2 [int(11)]
an_sense_no2 [smallint(6)]

Table A.3: Meronymy Table

MER (0 records)

me_word_no1 [int(11)]
me_word_no2 [int(11)]

Table A.4: Entailment Table

ENT (6 records)

en_word_no1 [int(11)]
en_word_no2 [int(11)]

Table A.5: Hypernymy Table

HYP (340 records)

hy_word_no1 [int(11)]
hy_word_no2 [int(11)]

Table A.6: Causality Table

CAU (3 records)

ca_word_no1 [int(11)]
ca_word_no2 [int(11)]

Table A.7: Coordinated Table

COO (1240 records)

co_word_no1 [int(11)]
co_word_no2 [int(11)]

Table A.8: Relation Table

RELATION (9 records)

il_relation_no [tinyint(4)]
il_relation_name [varchar(30)]
il_relation_table_name [varchar(4)]
il_relation_code [char(3)]
il_value [decimal(17, 7)]

Table A.9: Section-Info Table

SECTIONINFO (175 records)

bb_section_no [int(11)]
bb_supersection_no [int(11)]
bb_course_code [char(5)]
bb_week [smallint(6)]
bb_header [char(150)]
bb_link [char(150)]
bb_meaning_count [int(11)]

Table A.10: Word-Root Table

WORDROOT (11585 records)

kk_word [char(50)]
kk_root [char(50)]
kk_structure [char(200)]

Table A.11: Lecture-Word Table

LECTUREWORD (24779 records)

dk_section_no [int(11)]
dk_sentence_no [int(11)]
dk_word_order_no [int(11)]
dk_meaning_no [int(11)]
dk_word [char(100)]

Table A.12: Lecture-Meaning Table

LECTUREMEANING (10555 records)

da_section_no [int(11)]
da_sentence_no [int(11)]
da_meaning_order_no [int(11)]
da_word_no [int(11)]
da_sense_no [int(11)]

Table A.13: Question Table

QUESTION (100 records)

qu_question_no [int(11)]
qu_question [char(200)]

Table A.14: Question-Answer Table

QUESTIONANSWER (198 records)

qa_question_no [int(11)]
qa_order_no [int(11)]
qa_section_no [int(11)]

The bold field names represent the primary keys of the tables.

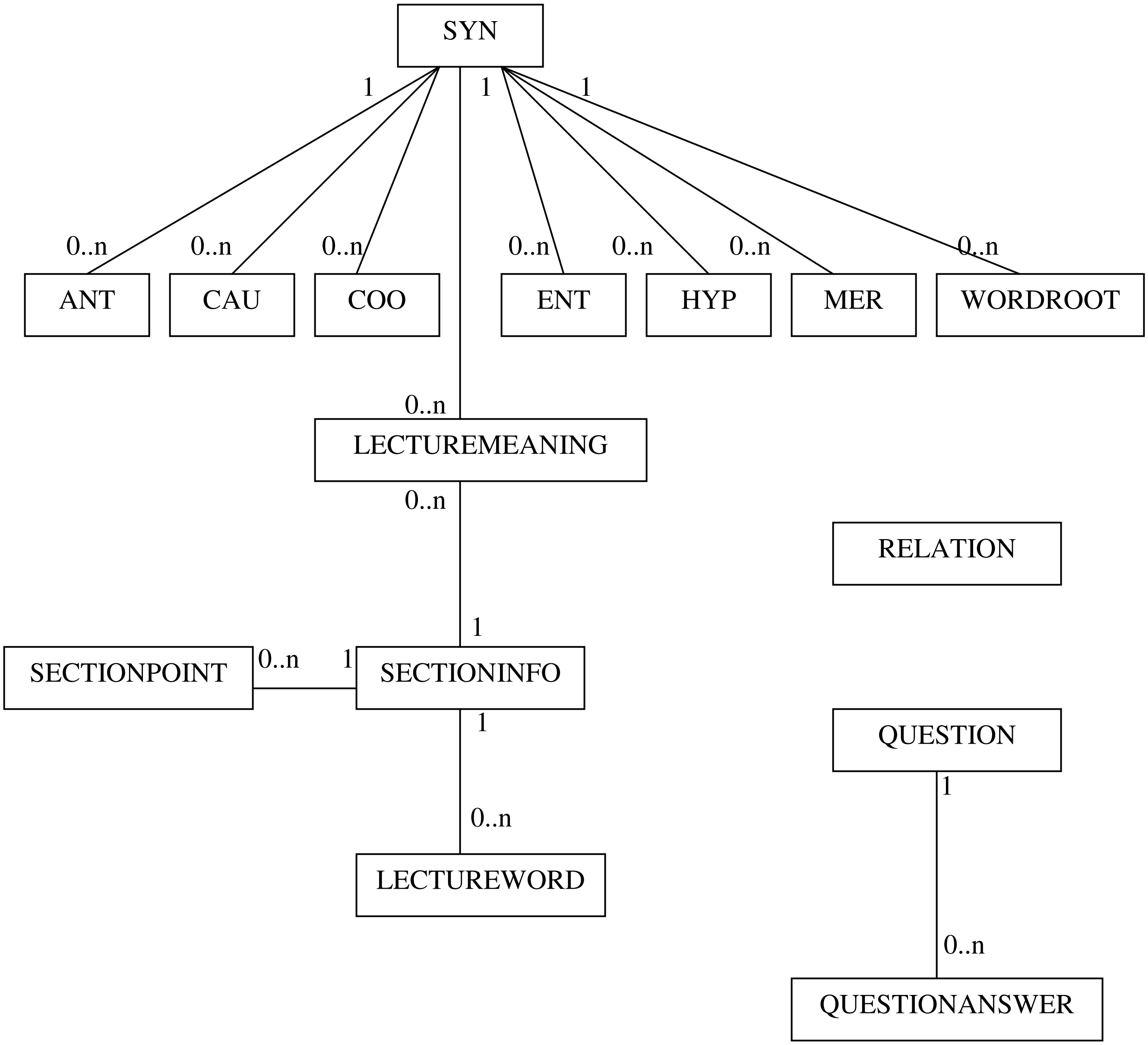

Figure A.1 The Entity Relationship Diagram of the Database