AN INTELLIGENT DIAGNOSTIC SYSTEM FROM THE CLINICAL NARRATIVES IN TURKISH

by

Muhammed Oğuzhan Külekci

B.S. in Computer Engineering, Boğaziçi University, 1998

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University

May, 2000

AN INTELLIGENT DIAGNOSTIC SYSTEM FROM THE CLINICAL NARRATIVES IN TURKISH

APPROVED BY:

Doç. Dr. Mehmed Özkan ....

(Thesis Superviser)

Doç. Dr. Ahmet Ademoğlu ....

Prof. Dr. Fikret Gürgen ....

DATE OF APPROVAL: ....

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assoc. Prof. Dr. Mehmed Özkan for his great support and guidance during this thesis study. I also thank to my sister Tuba Külekci who is currently a medical student and has helped for the translation of medical terms into daily life Turkish language. I would like to express my gratitude to my friends Kübra Balcı and Osman Burak Güran for their help while writing my thesis.

AN INTELLIGENT DIAGNOSTIC SYSTEM FROM THE CLINICAL NARRATIVES IN TURKISH

ABSTRACT

Expert system usage emerges in many areas as well as in medicine. Different methodologies of artificial intelligence have been employed to set up such systems. In this study an intelligent diagnostic system is build that accepts Turkish medical narrative texts as inputs and generates a decision based on keyword analysis. Natural language processing has to be implemented to make such a system work. With this purpose a new morphological analysis methodology named as KOZ is developed for agglutinative languages and a Turkish morphological analyzer is implemented with this technique. Among the different techniques of morphological analysis, KOZ is the only one that makes a group vise suffix search. A word-parsing algorithm, which is designed to use with this method, is also introduced and given name "Left-Right-Middle (LRM)" search. The overall view of the system includes finding the roots of the words in the input Turkish medical text by the morphological analyzer deployed, and matching the roots of the input with the knowledge database that keeps the keywords of the illnesses. According to the detected keywords a report is prepared explaining the results obtained. The system is tested with 69 patient records having four different classes of illnesses as respiratory, heart, blood and vessel diseases. The application classifies these 69 patients' medical narratives with 96 % accuracy. The usage of the developed morphological analyzer is not restricted with this study and is discussed at the conclusion for future benefits.

Keywords : Natural language processing, morphological analysis, medical keyword analysis, Turkish morphological analyzer

TÜRKÇE HASTA HİKAYELERİ ÜZERİNDE AKILLI TEŞHİS/TANI SİSTEMİ

ÖZET

Akıllı sistemler birçok alanda olduğu gibi medikal sahada da geniş ilgi uyandırmakta ve bu tip uzman sistemlerin oluşturulmasında yapay zekanın çeşitli metotları uygulanmaktadır. Bu çalışma kapsamında Türkçe yazılmış hasta hikayelerinden anahtar sözcükleri tespit ederek teşhis yapabilen akıllı bir sistem geliştirilmiştir. Böyle bir uygulamada doğal dil işleme tekniklerinin kullanılması gerekmektedir. Bu amaçtan hareketle, sondan eklemeli diller için KOZ adı verilen yeni bir biçimbirimsel çözümleme tekniği tasarlanmış ve bu metodoloji ile çalışan Türkçe biçimbirimsel çözümleyici meydana getirilmiştir. KOZ şu ana kadar morfolojik analiz için geliştirilen uygulamalar içerisinde ekleri grup bazında ele alan ilk ve tek sistemdir. Sözcüklerdeki eklerin araştırılmasını geliştirilen yolla yapabilmek üzere yeni bir kelime çözümleyici algoritması (LRM) da geliştirilmiştir. Sistemin çalışma prensibi Türkçe metin girdisi içerisindeki kelimelerin köklerini tespit ederek bilgi bankasındaki anahtar sözcüklerle karşılaştırmak ve bu anahtarlardan faydalanarak olası rahatsızlıkları tespit etmektir. Meydana getirilen uygulama, rahatsızlıkları kalp, solunum, kan ve damar olarak dört değişik grupta toplanan 69 hasta hikayesi üzerinde denenmiştir. Bu test kümesi üzerinde uygulama %96 doğruluk oranıyla sonuç vermiştir. Geliştirilen sistem dahilindeki Türkçe biçimbirimsel çözümleyicinin değişik kullanım sahaları da sonuç kısmında değerlendirilmiştir.

Anahtar Sözcükler : Doğal dil işleme, biçimbirimsel çözümleme, tıbbi anahtar kelime analizi,Türkçe biçimbirimsel çözümleyici iii

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

NLP : Natural Language Processing
AI : Artificial Intelligence
AIM : Artificial Intelligence in Medicine
EMR : Electronic Medical Record
HIS : Hospital Information System
FSA : Finite State Automaton
FSTN : Finite State Transition Network
RTN : Recursive Transition Network
ATN : Augmented Transition Network
LRM : Left Right Middle algorithm
KOZ : Kulekci&Ozkans' morphological analysis methodology
AGGT: Affix Group Generation Table
€ : Empty affix set

1. INTRODUCTION

Natural language processing (NLP) has been a scientific enterprise for 40 years and in the last decade applied NLP has gained great popularity [1]. The progress in both language engineering and speech processing aims to supply intelligent interfaces for the human-computer interaction. The application areas of such a ‘natural’ interaction is so wide that when this goal is reached, it will be a real revolution in computer related technologies.

The stages of natural language processing are [2]:

1. **Phonetic Level:** Concerned with the compositions of speech sounds.
2. **Phonological Level:** Concerns the way sounds combine to form words, word groups and sentences.
3. **Morphological Level:** Deals with the form of words in a sentence such as verb conjugations, plural of common nouns and so on. This level also deals with more complex aspects such as the way words can be formed by affixes, appendixes or from other words, nominalizations of verbs and so on called derivational morphology. The set of words of a language is stored in what is known as lexicon. The different morphological forms of the word are accessible via morphological rules. By application of these rules inflected or derived forms of the words can be generated or analyzed. It might be the case that different forms of the words are actually stored in the lexicon with the information associated with it (like tense of the verb, type of the word, number of noun, root of the word etc.). Alternatively, the lexicon may only list roots or stems, and there will be an associated set of procedures for getting from stems the full forms of words and vice versa.

Syntactic Level: Concerned with the way words combine to form sentences.

4. **Semantic Level:** Deals with the meanings of words, word groups and sentences. Each word at the lexical level has associated with it one or more internal or semantic representations, which can vary from theory to theory. The rules of semantic composition are associated with the syntactic rules and describe how the internal representation of a construction can be deduced or

calculated from the substructures it is made up of (principle of compositionality).

5. **Pragmatic Level:** Deals with the relationship between an utterance (or sentence) and its socio-cultural context. This component makes use of very varied and highly complex elements.

From the very earliest moments in the modern history of the computer, scientists have dreamed of creating an ‘electronic brain’ by means of an artificially intelligent computer system. Artificial intelligence has many challenging opportunities in many areas as well as in medicine. For thirty years there had been great achievements in ‘Artificial Intelligence in Medicine (AIM)’. In reviewing this new field Clancy and Shortliffe provided the following definition in 1984 [3]:

'Medical artificial intelligence is primarily concerned with the construction of AI programs that perform diagnosis and make therapy recommendations. Unlike medical applications based on other programming methods, such as purely statistical and probabilistic methods, medical AI programs are based on symbolic models of disease entities and their relationship to patient factors and clinical manifestations. '

Although much has been changed since then, that revolutionary definition emphasized the importance of expert systems in medical area. Knowledge based decision support systems had been the leading research field AIM. Expert systems are the commonest type of AIM systems in routine clinical use [3]. Some of these expert systems are *Dxplain, Menelas, Iliad, Jeremiah, Promnet, Orthaplanner*, and *Rapid*. Intelligent computers able to store and process clinical data would become perfect ‘doctors in box’, assisting clinicians in tasks like diagnosis.

A large number of AI techniques can be used and combined in different development stages of a knowledge based decision support system. Coiera stated that the entry and subsequent machine understanding of freely entered clinical notes into electronic systems will be a practical issue for the next decade and also there will be a value in using techniques from NLP to interpret archival printed notes and make them electronically accessible [4]. Coiera also stated that two of the reasons for the extent of use of decision support systems has not been as wide as originally expected are the lack of informational infrastructure to support these systems and the resistance of health care professionals to use the technology [5]. So using NLP to get the knowledge from freely entered medical narratives will establish a solution basis for the lack of informational infrastructure and will be helpful in medical recording. One of the reasons for the clinicians' resistance is thought to be the restrictions that had to be obeyed during the entrance of the knowledge to the expert system. Again by using NLP there will be a daily life media for the usage of the system and so will be much more easier to use as just writing the medical narrative will be enough at the end user side.

There has been a great effort on structuring the medical records for flexible access to free text medical data. One of the reason for the development and dissemination of electronic medical records (EMR) is impeded, is the 'unruliness' of medical narrative data [6]. In the literature, three major challenges were addressed: facilitation of direct data entry, achieving unambiguous understandability of data, and improvement of data presentation. Promising approaches to tackle the first and second challenge are the use of dynamic data entry forms that anticipate sensible input, and free-text data entry followed by natural language interpretation [7].

Natural language processing and medical concept representation can function to mutual benefit. NLP tools need a concept representation system for their semantics, whilst compositional medical concept representation systems need NLP of readable expressions for composed concepts [8]. In literature one can find medical records access systems using natural language. For example 'Menelas' is implemented in French and the goal is to provide better access to the information contained in patient discharge summaries using natural language [9].

A large number of hospital applications are closely related to natural language processing since they heavily depend on an efficient use of huge amounts of textual information. Natural language processing components offers a large variety of medical services according to free input [10]. So the NLP part of the decision support system is not restricted with the use in this expert systems. For example in the context of hospital information systems (HIS) medical free text analysis is reviewed with respect to literature retrieval, case retrieval and fact retrieval from textual data in the patient record [11].

In the literature there had been works in this way of thought. Carl-F. Bassoe had established a system, which does automated diagnoses from clinical narratives by computerized medical records, natural language processing and neural network technology called PROMNET [12].

2. REVIEW OF LITERATURE AND WORK DONE ON TURKISH

The morphological analysis is the most important part of an NLP system on a language. That is because construction of further steps of NLP (syntax, semantics, etc.) depends heavily on morphological analysis. Different formalisms have been used for this analysis and implementations are given for various languages. There exist four main formalisms, which construct a basis for the morphological analyzers [13]:

1. Finite State Transition Networks

A transition network is a parser that has a number of distinct states and proceeds from state to state in a manner controlled by the input string. As the number of states is finite, that is called Finite State Transition Network (FSTN) or Finite State Automaton (FSA). A FSTN can be regarded as a neutral description of a language (a set of sequences of symbols). It can also be interpreted, for instance, as a specification of an FSA to recognize elements of the language or as a specification of an FSA to generate elements of the language [14].

2. Finite State Transducers

A Finite State Transducer (FST) is an FSTN in which each of the transitions can produce output as well as accepting input. An FSTN is actually a recognizer that tests whether the input is well formed or not, where a FST generates outputs during the state transitions.

3. Recursive Transition Networks

A Recursive Transition Network (RTN) is one in which a state transition can either accept an input or execute (call) another entire network. That means a sub network call is possible through an arc. It also means that the number of states available to the parser is no longer finite, because any number of recursive invocations of the same network could be used at the same time.

4. Augmented Transition Networks

An Augmented Transition Network is like an RTN with the additional features that:

- Each subnetwork can have registers (memory locations) in which information can be stored.
- Each arc can have actions associated with it. These actions include storing, retrieving, and testing register values, and adding items to, or retrieving items from, a holding list.
- Data can be transferred between the registers of a subnetwork and the registers of the network from which it was called.

Morphological analysis is a complex task for the agglutinative languages where affixation is theoretically infinite. Some general characteristics of agglutinative languages may be summarized as:

- There are many suffixes or prefixes;
- Suffixes are single syllables in general;
- The meaning of a word and its role in a sentence heavily depends on the affixes;
- The syntax of the root of the word does not change much due to the added affix. Small changes such as dropping or change of a letter are not problematic;
- There exists strict morphophonemic rules defining the concatenation of syllables;
- Suffixes also obey morphophonemic rules (new syllables can be added for vowel/consonant harmony rules);
- There exist rules that formulate the formation of any word by inflection or derivation. Mainly those describe which type of affixes can or cannot come after or before another type of affix. These rules are called morphotactic rules.

All the affixes in a language form a finite set. Although affixation in an agglutinative language is theoretically infinite, all the meaningful affix combinations can be generated forming a finite set by specifying borders in that infinite space with the morphotactic and morphophonemic rules [15,16].

Until now it has been a general trend to isolate the affixes from roots by stripping away affixes one by one in agglutinative languages. For the direction of the analysis both left-to-right [17] (root matching) and right-to-left (affix stripping) parsing algorithms have been used. In these approaches, root-matching process involves first finding the root and then searching the affixes to match with some part of the rest of the word. In affix stripping, affixes are taken away one by one until a root is obtained. If an affix is matched that part is stripped and the search goes on for the remaining part of the word in the same manner until a successful parsing is achieved.

The morphological analysis of Turkish has been studied with different formalisms of finite state techniques. An FSTN implementation is given by Hankamer [18], and an ATN formalism is applied by Güngör [14,19]. The two-level morphological analysis introduced by Koskenniemmi [20] had gained popularity especially for the analysis of agglutinative languages and Oflazer [21] had given a two-level morphological description of Turkish. The general way of thought for these implementations is based on a left-to-right search and can be best understood with an example. To parse “evdekilerin” in Turkish first the root “ev” is found and then the rest is evaluated by searching for each suffix which produces “de-ki-ler-in”. However, while conducting the affix search if limited morphotactic rules are employed the system can end up in infinite loops searching for meaningless possibilities as stated in the Ph.D thesis of Güngör[14].

In this proposed study, instead of searching affix by affix, root matching and suffix stripping are performed in a hybrid manner using the affix groups, which are compiled in a database. The preparation of the affix-groups is the most crucial part of the study. By this approach the above example will be processed as “ev-dekilerin” by first locating the root and then finding the rest (“dekilerin”) as a single entity in the affix-group database. To achieve this a different search technique, which is a combination of left-to-right and right-to-left approach is proposed here.

3. DESCRIPTION OF THE PROPOSED MORPHOLOGICAL ANALYSIS METHODOLOGY (KOZ)

The proposed morphological analysis methodology performs the parsing of affixes in groups instead of traditional single suffix search. To accomplish that group vise investigation, first definitions of some new concepts are introduced and then the method is described.

3.1. Definition of New Concepts:

Note that during the whole study, all different surface forms of any affix are taken into consideration as a separate affix.

3.1.1. Affix-Set

The affixes that are in the same grammar classification and have the same morphophonemic characteristic of the language form an affix-set. The main criteria for the right formation of an affix set is that; if an affix can be concatenated to one of the elements in an affix-set, then it should be concatenated to the other elements of the same set. As an example {im, in, imiz, iniz} forms an affix-set in Turkish as all of them are in class of possessive suffixes, and have the same morphophonemic characteristics due to the vowels they contain. An affix that can be added to one of the elements in this set, can also be added to other elements of the same set.

3.1.2. Affix-Group

The concatenation of affix-sets defined by morphotactic rules forms an affix-group. As an example the two affix-sets in Turkish, {im, in, imiz, iniz} and {de}, may be concatenated due to Turkish morphotactics rule indicating that possessive suffixes can be followed by locative suffixes. This results the ‘imde’, ‘inde’, ‘imizde’, ‘inizde’ affix-groups. Nominal case of each affix-set is also an affix- group.

3.1.3. Affix-Group Generation Table

Morphotactic rules of a language specifies the order of affixation such as an affix of some class can or cannot come after or before another affix class. Affix-Group Generation Table (AGGT) is a table that summarizes these rules by which generation of all possible affixes in a language can be performed automatically. The three properties of an AGGT are:

1. The cells of an AGGT represent affix-sets.
2. The cells in the same column of an AGGT are of the same affix class
3. The cells in the same row have the same morphophonemic characteristics that can be concatenated to each other forming an affix-group.

Because in agglutinative languages the number of affix-groups is very high, the possible combinations are generated automatically using the AGGTs. Noun and verb inflection AGGTs are represented in Appendix B.

The following example states an overall view of the proposed concept and methodology. With the traditional formalisms, the Turkish word "ellerimde " (in my hands) is analyzed as first detecting the root "el" (hand), and next identifying the suffixes "ler" (plural) , "im" (possessive),and "de" (locative) respectively. With the proposed methodology, the identification of the suffixes is performed by groups. The suffixes of the word ("ler", "im", "de") are elements of related affix-sets. The concatenation of them, which is already produced by the preprocessing using the related AGGT, forms the affix-group "lerimde" which is stored in the database with all the other affix-groups. Now, combined suffixes of the word, "lerimde", are recognized in one shot. Note that, generating all meaningful affix-groups in the language and keeping them in a database requires a preprocessing that will be described in the next section.

3.2. Method

The affix-groups in the language are generated and kept in a database by a five step preprocessing. The Left-Right-Middle (LRM) search algorithm depends on these meaningful affix-groups for the morphological analysis.

3.2.1. Definition of Morphophonemic Rules

The rules that define the concatenation of syllables are stated. Generally, vowel or consonant harmony rules, change or drop of letters, and addition of auxiliary letters between syllables are in scope of morphophonemic. These rules are used for two purposes. First, different surface forms of an affix are specified, and second, the affixes are grouped due to their morphophonemic characteristics to set up the affix-sets

3.2.2. Definition of Morphotactic Rules

The rules that define word inflection or derivation in the language are stated. These specify which type of affix can or cannot come before or after another type of affix. The preparations of AGGTs (described in 3.2.4) are based on the general word formation in the language, which is considered in this step.

3.2.3. Classification of Affixes and Affix-Set Formation

The affixes with all their different surface forms in the language are collected and classified. After the affix classification, using the rules of step one, affixes are arranged to form the affix-sets. The resulting classification schema is as shown in Figure 1. It must be noted that steps 1,2 and 3 are not disjoint and must be considered concurrently.

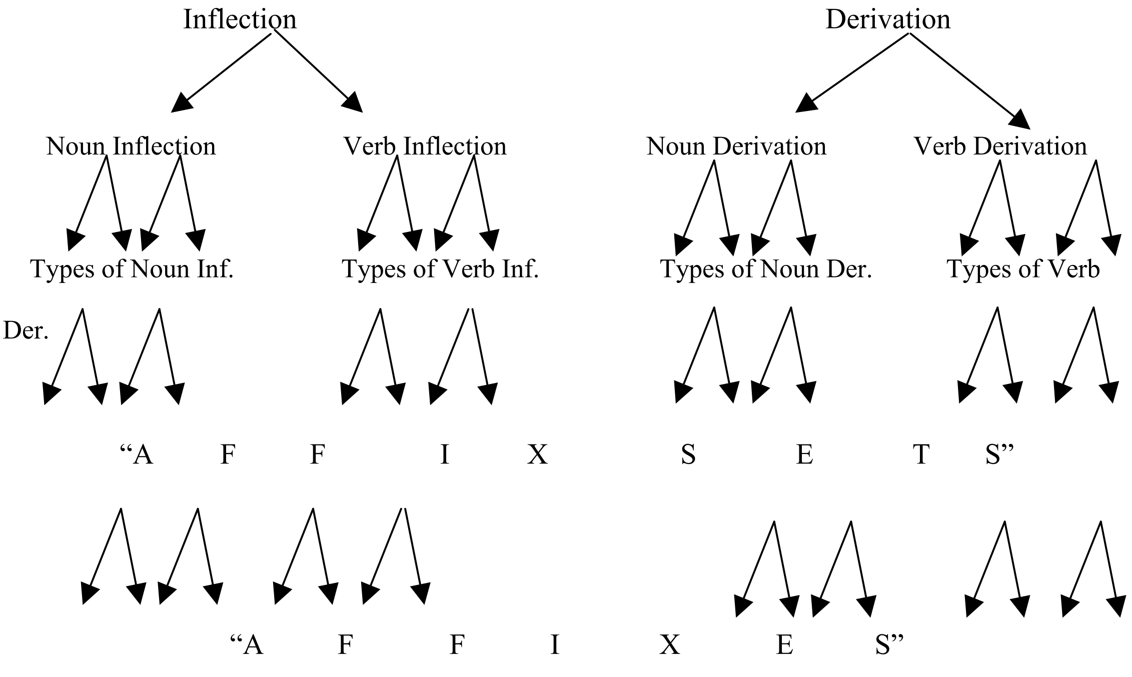

Figure 1. Classification of affixes for the proposed morphological analysis

3.2.4. Preparation of Affix-Group Generation Tables

The affix-group generation tables (AGGT) that define the way of concatenation method of affix sets are created by taking the results of steps 1, 2 and 3 into consideration. These look up tables define which affix-sets will be combined with which other affix-sets.

3.2.5. Generation of Affix-Groups

After the completion of step 4 the infra structure to generate any possible meaningful combination of all the affixes is available. What comes next is the column multiplication of rows of each table to generate the affix-groups and to store them in a database. During the addition of the generated affix-groups to the database, the related information of each group i.e., personal information, type of inflection/derivation information, are also added as separate fields of the generated group.

The preprocessing for the proposed morphological analyzer is achieved as steps 3.1-3.5 are completed. All the affix-groups are generated and kept in the database with all the necessary information. Now the words of the language are in lexicon and all the possible meaningful affix groups are available. The morphological analysis of a given word is done by LRM algorithm. LRM searches the generated affix-groups from the databases and the roots from the lexicon for a suitable parsing of the word.

3.2.6. Left-Right-Middle (LRM) Algorithm

The search is based on a combination of left-to-right, and right-to-left analysis. The word is analyzed in three groups as root, derivations and inflections. The candidates for the root and affix-groups are ranked in a decreasing order of their lengths. First the root matching from the left is done. When there exists a possibility of letter change or drop at the root candidate due to a morphophonemic rule, that surface form of it is also taken into consideration with a special tag. If the root candidate is equal to the word analyzed, then one likely parsing is obtained. Next, an affix-group candidate from the affix-group database is matched with the right portion. If any unmatched part remains in between, that can be one of a derivation group or an inflection group. This part is searched for a suitable parse with the affix-groups in database. If a suitable parse is obtained, then the parsing is complete for this iteration. When there is no more affix-group candidates, then iteration continues for the next root candidate.

4. IMPLEMENTATION OF KOZ ON TURKISH

4.1. Definition of Morphophonemic Rules in Turkish

In Turkish suffixes are concatenated to the word (and also to other suffixes) with obeying the two vowel harmony rules. These rules state that which types of vowels can exist together in a word. In Table 1 the morphophonemic classification of eight vowels in Turkish is depicted.

Table 1. Morphophonemic classification of Turkish vowels

	UNROUNDED		ROUNDED	
HIGH	ı	i	u	ü
LOW	a	e	o	ö
	BACK	FRONT	BACK	FRONT

The two rules of vowel harmony that define suffix to suffix, or suffix to word concatenation are :

a) **Strong Vowel Harmony:** Only back vowels can come after back vowels and only front vowels can come after front vowels.

b) **Weak Vowel Harmony:** Only unrounded vowels can come after unrounded vowels, and only high-rounded or low-unrounded vowels can come after rounded vowels.

Note that in Turkish there exists some letter changes, additions, or drops when the suffixes are concatenated to the word. The rules governing such changes are:

a) **Softening of hard consonants:** If a word ends with one of the hard consonants {p,ç,t,k} and a suffix beginning with a vowel is concatenated to it, these hard consonants change to soft consonants {b,c,d,g}.

b) **Resembling of hard consonants:** If a word ends with one of the consonants {p,ç,t,k} (hard consonants) and concatenates a suffix beginning with a {b,c,d,g} (soft consonants) , the soft consonant {b,c,d,g} at the beginning of the suffix changes to a hard consonant {p,ç,t,k}.

- **c) Auxiliary letter concatenation:** During the concatenation of words and suffixes sometimes a new consonant or vowel, which is called auxiliary letter, is added in between. Auxiliary vowels {ı,i,u,ü} come between two consonants (a morpheme ending with a consonant plus a morph also beginning with a consonant), and auxiliary consonants {y,n} come between two vowels (a morpheme ending with a vowel plus a morph also beginning with a vowel).
- **d) Letter drop:** Last vowel of some special words drops when a suffix beginning with a vowel is concatenated (ex: ağız + ı = ağzı). Especially these are the words defining the organs of the body.

Some different surface forms of a suffix may be observed due to the morphophonemic rules. The two-level morphology [19] handles these situations by introducing free variables for the letters of possible changes. In this work, all of the different forms of suffixes will be defined in the related suffix sets. For example both “du” and “tu”, which are mainly the different surface forms of same suffix in Turkish, will be existent in the same suffix set. While building the affix-group generation tables, with morphotactic and morphophonemic rules, only the correct affixations would be made possible for the generation of affix-groups. Such as; “muştu” (not “muşdu”) and “yordu” (not “yortu”) affix groups, which are the correct forms, will be generated from the related affix group generation tables.

4.2. Definition of Morphotactic Rules in Turkish

The general morphotactic rules in Turkish can be summarized in seven main titles:

1) The word formation in Turkish is as:

root + derivational suffixes + inflectional suffixes

The possible combinations of word derivation in Turkish can be traced using the map shown in Figure 2.

Noun-to-noun or verb-to-verb derivations may be recursive, but verb-to-noun or noun-to-verb derivations cannot be consecutive. That means there may be one or more suffixes concatenated in verb-to-verb or noun-to-noun derivations, where only one suffix affixation is allowed in verb-to-noun or noun-to-verb derivations. There exists an exception where noun-to-verb derivation comes after noun inflection for the formation of “noun verbs” in Turkish (ex: evdeymiş). That is because, these “noun verbs” are concatenations of the inflected nouns and a special type of suffix verb “-i”. The example given above may also be written as “evde imiş”. Another exception for the way of affixation is that; sometimes plurality suffixes may come after possessive suffixes such as in “baba-m-lar”.

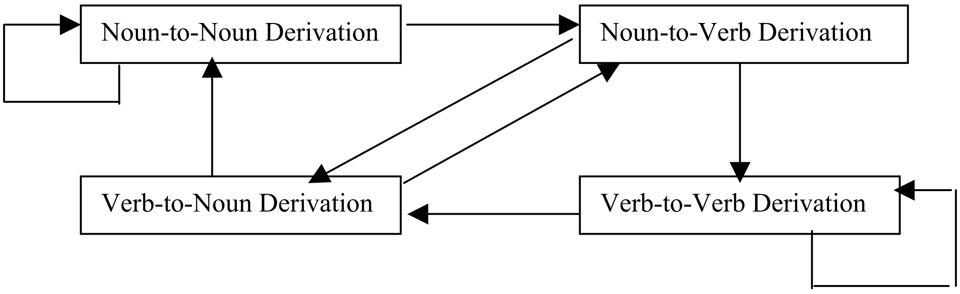

Figure 2. Noun/Verb derivation map in Turkish

Note that in an agglutinative language theoretically infinite number of suffixes may be concatenated. But in real life the meaningful combinations are finite and the difficulty is specifying the optimum boarders of the formation. The scheme proposed above is thought to be the most general way of word formation in Turkish.

2) In noun inflection two suffixes of the same type cannot be concatenated except for some limited number of usage.

3) There are three variables defining noun inflection as; Plurality Suffixes + Possessive Suffixes + Case Suffixes. Plurality Suffixes and Possessive Suffixes can take one of two values as being present or not. When we consider the 7 subgroups of case suffixes (genitive, accusative, dative, ablative, locative, instrumental, equality), the total number of different combinations is 32 and may be shown as in Table 2:

4) There exists a special suffix ‘ki’ in Turkish that behaves like a pronoun. Generally, it can appear after the genitive and locative case suffixes. The only suffix-sets, which can appear after ‘ki’ suffix, are labeled as Poss_14(si), Gen_5(ni), Dat_4(ne), Abl_6(nden), Lok_6(nde), Ins_4(yle), and Equ_4(nce, ncesine) in Table 3. Nested ‘ki’ suffix usage is very rare and considered as an exception.

5) In verb inflection, the first class of suffixes indicating personality comes after Present Tense, Present Continuous Tense, Future Tense, Past Tense Type 2 (miş’li Geçmiş Zaman), and requirement modal (Gereklilik Kipi) inflections.

Table 2. Possible combinations of noun inflection suffixes in Turkish

PLURALITY	POSSESSIVE	CASE
0	0	0
0	0	Genitive
0	0	Accusative
0	0	Dative
0	0	Ablative
0	0	Locative
0	0	Instrumental
0	0	Eguality
0	1	0
0	1	Genitive
0	1	Accusative
0	1	Dative
0	1	Ablative
0	1	Locative
0	1	Instrumental
0	1	Eguality
1	0	0
1	0	Genitive
1	0	Accusative
1	0	Dative
1	0	Ablative
1	0	Locative
1	0	Instrumental
1	0	Eguality
1	1	0
1	1	Genitive
1	1	Accusative
1	1	Dative
1	1	Ablative
1	1	Locative
1	1	Instrumental
1	1	Eguality

6) Second class of suffixes, again indicating personality comes after Past Tense Type 1 (di’li Geçmiş Zaman) and conditional modal (Şart Kipi) inflections. The personality knowledge for the ordering (Emir) and wish (İstek) modals and the negative inflection of the Present Tense are embedded to the suffixes in the related affix-sets in this work.
7) In Turkish, joint and double joint inflection of verbs are possible. The joint inflections are of type two as story joint (hikaye), and hearsay joint (rivayet). Inflecting these story or hearsay with conditional modal does the double joint inflection. The hearsay joint inflection takes first type of personality suffixes to its end, and all the other joint and double joint inflections take second type of personality suffixes.
8) The story and hearsay inflection of ordering modal do not exist. The double joint conditional inflection of both story and the third singular person inflection of a verb do not take any personality suffix.

4.3. Classification of Affixes and Formation of Affix Sets in Turkish:

The classification of Turkish suffixes may be summarized as in Figure 3.After the suffix classification, the inflection or derivation suffix sets are created with all the possible surface forms subject to morphophonemic changes. As a result, the noun and verb inflection suffix sets are produced as shown in Appendix A. Each cell of these tables corresponds to a suffix set.

Both noun and verb suffixes are kept in records whose structures are given below:

```
type
  ek= record
     val:string[8]; {holds the suffix itself }
     tip :integer;    {holds the type information due to the bit graph in Table 3 and Table 4}
    sahis:integer; {holds the person information if exists. If not, it is set to 0.}
  end;
```

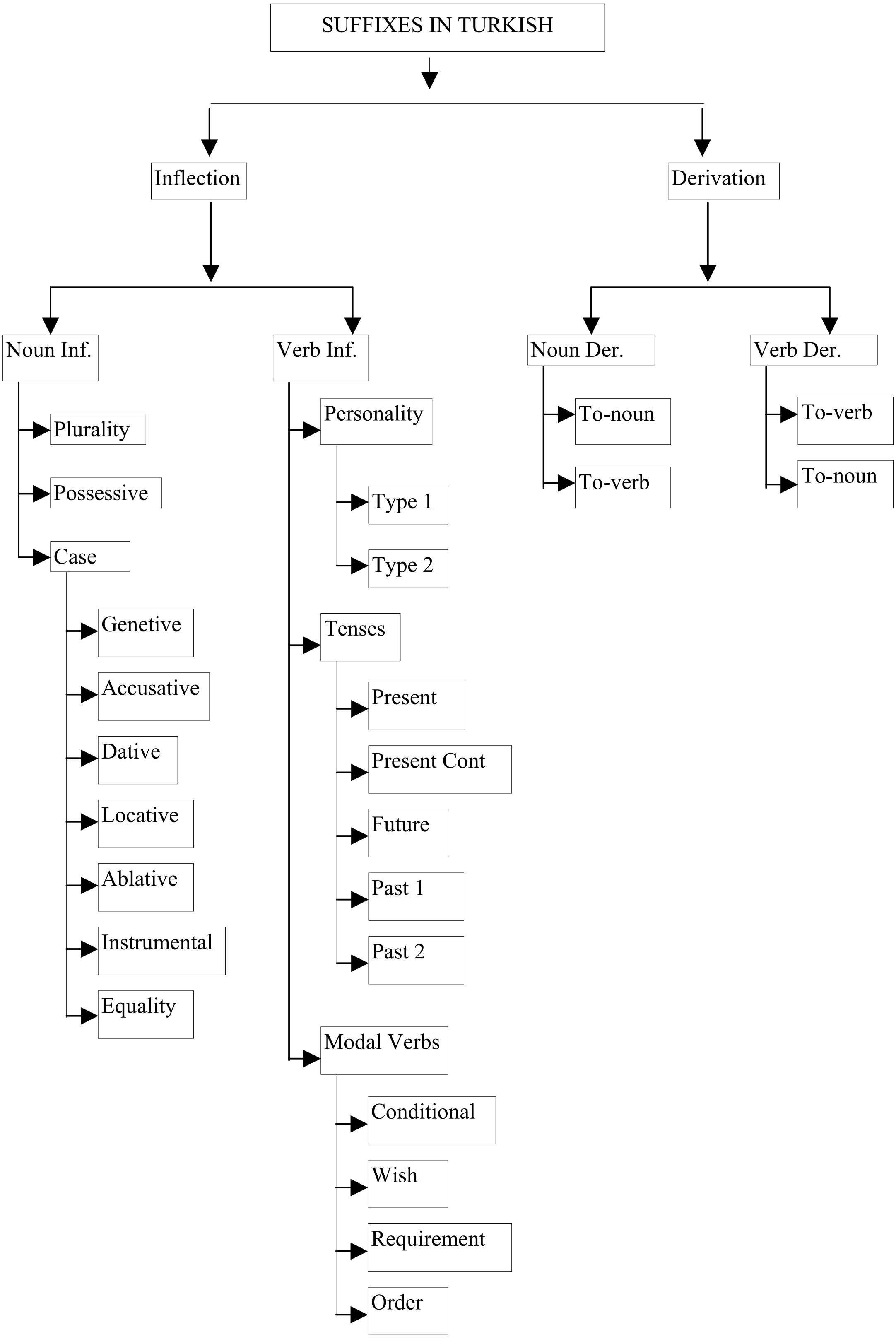

Figure 3. The classification of suffixes in Turkish

Table 3. Bit graph of the ‘tip’ attribute of Turkish noun suffixes

Main Type		**Sub Type**			**Set Number**				
9	8	7	6	5	4	3	2	1	0
00 (Not Used)		**000** (Not used)			The number of the suffix-set that encapsulates the suffix.				
01 (Plurality)		**001** (Genetive)							
10 (Possessive)		**010** (Accusative)							
11 (Case)		**011** (Dative)							
		100 (Ablative)							
		101 (Locative)							
		110 (Instrumental)							
		111 (Equality)							

Table 4. Bit graph of the ‘tip’ attribute of Turkish verb suffixes

Main Type			**Sub-Type**				**Set Number**				**Positive/ Negative**
12	11	10	9	8	7	6	5	4	3	2	1
000 (Personality)			0000 (First type of personality suffix)				The number of the suffix-set that encapsulates the suffix.				States whether the suffix implements a positive or negative meaning.
001 (Inflection)			0001 (Second type of personality suffix)								
010 (Partisip)			0010 (Present Tense)								
011 (Gerendium)			0011 (Present Continuous Tense)								
100 (Compound)			0100 (Past Tense Type 1)								
101 (Noun -based)			0101 (Past Tense Type 2)								
			0110 (Future Tense)								
			0111 (Order model)								
			1000 (Conditional model)								
			1001 (Wish model)								
			1011 (Requirement model)								

4.4 Preparation of Affix-Group Generation Tables for Turkish

The suffix-sets for noun and verb inflection, noun-to-noun derivation, noun-to-verb derivation, verb-to-verb derivation and verb-to-noun derivation are defined in step 3. The look up tables, that include the identification numbers of these suffix sets in their cells, are created in this fourth step. Within these suffix-group generation tables the elements on the same row has the same morphophonemic characteristics and can be concatenated. The morphotactic rules explained in previous steps indicates which columns can be affixed to each other. The affix-group generation tables can be seen Appendix B. These tables are used for the generation of noun inflection suffix-groups as described in the next step.

One suffix-group generation table is enough to show all the possible noun inflections in Turkish, but for the verb inflection it is impractical to include all verb inflections in just one table. That is because verb inflection is much more complex and detailed in Turkish. To overcome the problem 19 different suffix-generation tables are used. Five of these are used to show normal tense inflections, 4 of them are for normal modal verb inflections, another 6 are for joint verb inflections, and the remaining 4 are for double joint verb inflections. The interrelations of the cells in the Verb AGGT (affix-group generation table) are as the same as in Noun AGGT.

4.5. Generation of Suffix-Groups for Turkish Language

The number of possible meaningful suffix-groups for the inflections is very high in Turkish. This task is automated with a software. The affix-group generation tables are constant look up tables. The program takes the columns that will be affixed as arguments and does the affixation of each suffix-sets in the same row of the columns.

As an example, for the noun inflection, Table 2 defines the way of concatenation of suffix classes. The last row of Table 2 indicates the possibility that a plurality suffix can be followed by a possessive suffix, which can also be followed by an equality case suffix. The program must multiply the “plural”, “possessive” and “equality” columns of Table 4, yielding all the possible combination of these 3 columns in Table 4, and iterates on each row of these columns to affix the elements of suffix-sets in the defined order (plural + possessive + equality). For example the first row of these columns has the suffix-set numbers 1, 1, 3 respectively. The program takes the first suffix-set of *plurality*, first suffix-set of *possessive* suffix-set and the third of *equality* case suffix-set and concatenates them. This step produces {ler-imiz-ce, ler-imiz-cesine, ler-iniz-ce, ler-iniz-cesin, lerince, lerincesine} suffix-groups and each of these groups are stored in the noun-inflection suffix-groups database with the necessary information they include. That process is repeated for all the rows in Table 2.

In a similar way, all the noun and verb inflections are generated according to the related AGGT. The result is several suffix-groups forming the database.

In the derivational part, the same procedure is used for the noun-to-noun and verb-to-verb derivations. For noun-to-noun derivations a maximum of three and for verb-to-verb derivations a maximum of four suffixes are allowed to be affixed consecutively. It is believed that the most efficient classification of derivational suffix-groups will be obtained with these maximums, as concatenations of more suffixes are rare in Turkish. Even that rare situation is not problematic with the Left-Right-Middle (LRM) search algorithm described below.

As described in Step 2, more than one noun-to-verb and verb-to-noun suffix concatenations are not allowed in Turkish. That means these type of suffixes cannot produce suffix-groups. Existence of all morphophonemic forms of the single suffixes in the database is sufficient. There is no need for an extra affix-group generation for them.

With the completion of all 5 steps, following suffix-groups become available:

1. Noun inflection suffix groups
2. Verb inflection suffix groups
3. Noun-to-noun derivation suffix groups
4. Verb-to-verb derivation suffix groups
5. Noun-to-verb derivation suffix groups
6. Verb-to-noun derivation suffix groups

The database structure holding the generated noun and verb suffix groups may be summarized as:

Noun Suffix-Group Database Structure:

The attributes of the noun suffix-groups are :

Dummy : An auto incremented number.

EK : The string holding the suffix.
TIPI : An integer holding which kinds of suffixes are included in the EK.
KI : An integer specifying whether a ‘ki’ suffix exists. If exists what types of suffixes are appended to it.
SAHİS : An integer specifying whether a person information is meant.

TIPI field:

One digit is reserved for types of suffixes Plurality, Possessive, and Case

Cogul	**Iyelik**	**Hal**
0	0	0
1	1	1
		2
		3
		4
		5
		6
		7

0 means selected type does not exists, and 1 means existance. For the ‘Case’ suffixes 0 means non-existance where a non-zero value holds the type of the ‘Case’ suffix.

KI field:

Similar to the TIPI field, KI field reserves one digit for each

Ki-suffix	**İyelik**	**Cogul**	**Hal**
0	0	0	0
1	1	1	1
			2
			3
			4
			5
			6
			7

0 means selected type does not exists, and 1 means existance. For the ‘Case’ suffixes 0 means non-existence where a non-zero value holds the type of the ‘Case’ suffix.

Verb Suffix-Group Database Structure:

Dummy : An auto incremented number.
EK : The string holding the suffix.
ZAMAN : An integer indicating the type and tense of suffix..
SAHİS : An integer indicating the person information.
KADEME : If double tense inflection is used , indicates it.

4.6. Left-Right-Middle Search Algorithm for Turkish Word Parsing

LRM search algorithm to parse a word basically consists of four main steps:

1) Root candidates from the word lexicon are chosen and listed in decreasing length of match with the left side of the analyzed word. Note that if there exists a possibility of letter change or drop at the root candidate due to a morphophonemic rule, that surface form of it is also added to the list with a special tag. The top of the list is the first root candidate. If it is equal to the word analyzed, then one of the solutions of the morphological analysis is obtained as a nominal case. If so, we take the next candidate from the list for the analysis, otherwise we move to step 2.

2) The root candidate is stripped away from the word. The remaining part is searched for an inflection at the right end side. Again candidates of inflection suffix-groups are ranked with the maximum character match criteria. If no inflection is found, then proceed from step 4.

3) If any inflection suffix-group is detected, then this suffix-group candidate is stripped away from the right hand side.

4) At this stage, what is left is either a derivation group of suffixes or an empty set. If the remainder is not an empty set, this part is searched for a suitable parse of derivation with the derivational suffix-groups in database according to Figure 2. If a suitable parse is obtained then that is a solution. In that case same procedure is repeated starting from step 2 for the next inflection group candidate. If there is no more candidate for the inflection, then iteration continues starting from step 1 again for the next root candidate.

5. KEYWORD ANALYSIS TECHNIQUE

Template matching and keyword analysis methodologies are the popular techniques that are used extensively in NLP applications. In 1950, Alan Turing proposed that a machine should be considered intelligent if a human being, communicating with it by teletype, could not distinguish it from another human being. In 1966, Joseph Weizenbaum published a computer program called ELIZA that seemed to hold Turing criteria [13]. ELIZA was a template matching system that gives canned responses to the user by recognizing certain patterns of word in the input text. Actually it did not understand much of its input, but produced appropriate answers by detecting some definite templates. For example, if user typed ‘You are X’, ELIZA could respond ‘What makes you think I am X’, where X is any adjective. The first example of a template matching system proved that templates are powerful enough for quite realistic natural language dialogue.

The problem of the template matching system is that it may fail in slight variations of patterns (words). To overcome that problem, instead of matching whole sentence to a template, a keyword system looks for specific words in the sentence and responds to each word in a specific way [13]. Blum (1966) created one of the first keyword systems, which was basically a natural language interface to the operating system. The system was able to understand sentences like ‘Copy 1 file from X to Y in binary’ and convert them to system commands.

Keyword analysis has been used since that time in a wide area where the input is known to contain certain kinds of information. Unlike template matching, slight variations of wording do not throw off the system, and unrecognized words are simply skipped. When a powerful morphological analyzer is used and the keywords are represented as word roots, the number of unrecognized words will lower which results a gain in information.

Both keyword analysis and template matching systems have ‘simplify’ and ‘translate’ stages. First the input is simplified to information needed, and then this knowledge is used by the translate session to derive a decision.

6. SYSTEM ARCHITECTURE

The intelligent diagnostic system from the clinical narratives is implemented with keyword analysis technique that relies on the morphological analyzer (KOZ) developed. An overall view of the application is like in Figure 4.

The word lexicon used in the system is maintained from the Turkish Orthography Guide [22] of Turkish Language Council (Türk Dil Kurumu). Some processing has been done to make this lexicon available for the morphological analyzer deployed.

The symptoms of diseases actually form the keywords database. These definitions are taken from ‘Semptomdan Teshise’ [23]. Four areas of illnesses (respiratory, heart, blood and vessel) are considered for the test of the application and these diseases are matched with their identifying keywords. The application is designed in a way that new disease definitions and keyword entrance are possible via databases.

The system starts with the input of the medical text defining the symptoms of a disease. This narrative may be either loaded from a text file or freely entered by the user to the rich text area on the user interface (see the Appendix C for the graphical user interface of the application). The words of the input text are fed into the morphological analyzer to find their roots. When the roots of the narrative are resolved, the matching keywords are searched from the keyword database. The symptoms in the text are detected at the end of this stage. These recognized keywords (symptoms) are searched in the disease database to rank the possibility of illnesses. At the end a report is displayed that shows the possible diseases in order of the number of keywords matched with each illness.

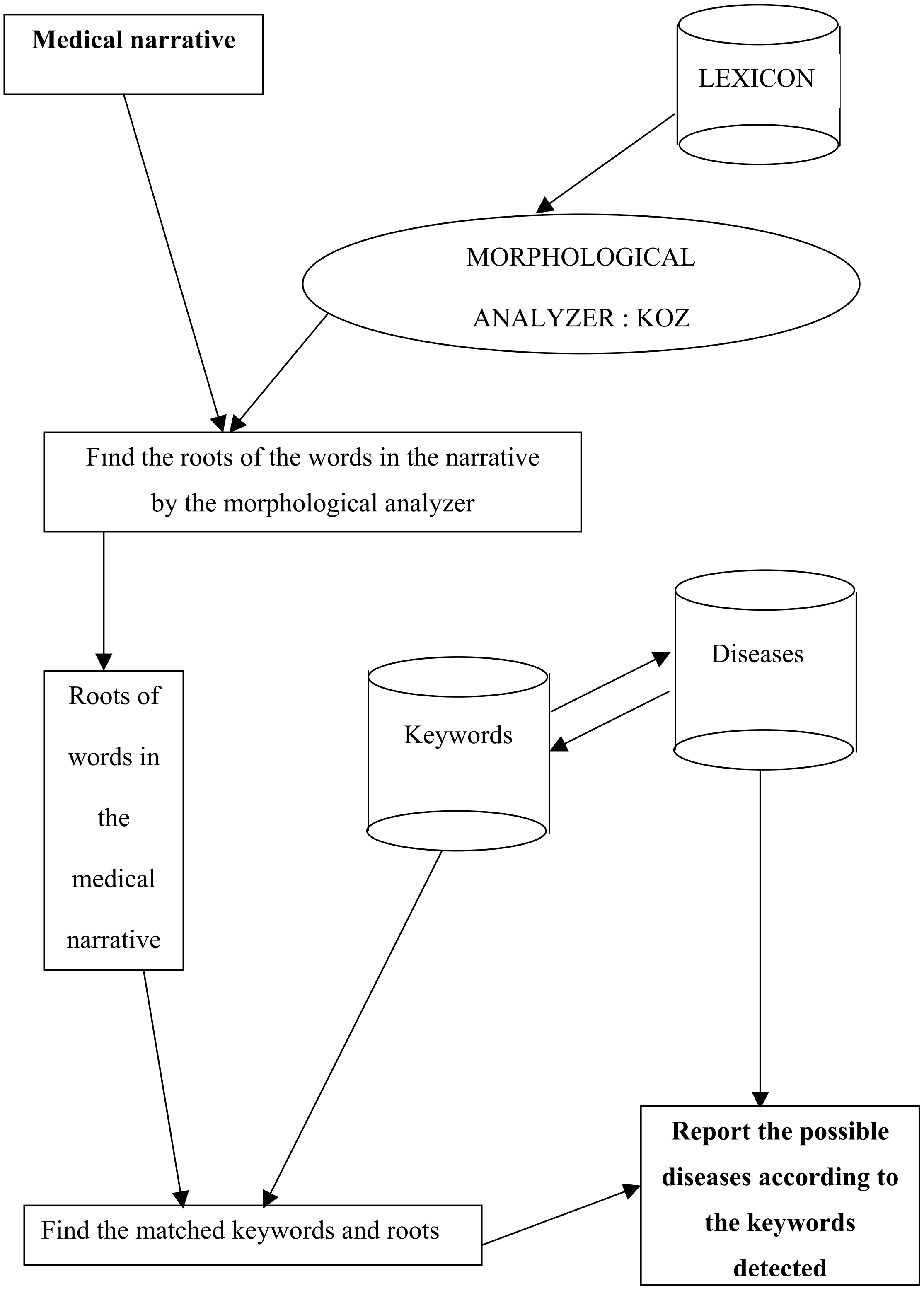

Figure 4. Overall system view of application

7. CONCLUSION

Within this thesis a new morphological analysis methodology is developed for the agglutinative languages and an implementation of it on Turkish is given. Based on this analyzer an intelligent diagnostic system from Turkish clinical narratives is build with the keyword analysis technique. The system is tested on 69 patient records explaining the complaints of patients. This data is taken from another expert system application, which works according to the yes/no trees [24]. Table 5 shows the results of the system on these medical narrative data.

Table 5. NLP based medical diagnosis system evaluation

	Heart Diseases	Vessel Diseases	Blood Diseases	Respiratory Diseases	TOTAL
Number of patients	19	12	18	20	69
Number of miss classified patients	0	0	3	1	4
Number of correct, but ambiguous results	4	0	4	2	10
Error Ratio	0 %	0 %	16 %	5 %	6 %
Ambiguity Ratio	21 %	0 %	22 %	10 %	14 %

The ambiguity of a decision is defined as one or more diseases have been ranked as the most possible disease with the correct disease. For example, if the system classifies a medical narrative, that belongs to a patient with blood problems, to both blood and heart diseases, then that result is a correct but an ambiguous answer. It can be observed from Table 21 that ambiguity rates of the system are not so low. That is because one keyword especially exists in more than one disease. To overcome that problem a more descriptive way of disease definitions may be done by assigning weights to the keywords and with an interactive application the system may ask distinctive questions to the user when an ambiguity occurs. With that approach a weighted average may lower the ambiguity rates. Within this study only morphologic analysis stage of NLP has been covered. If semantic and syntactic representations are applied both to the system and to the diseases it will also increase the performance of the system.

The overall performance evaluation of the system is in an acceptable range for a NLP application. Most of the errors reported are due to the short narratives of patients which are not so descriptive and do not contain enough number of keywords. The morphological analyzer deployed has been observed to work fast and correct. Although morphological analysis is a heavy CPU task, with the architecture and methodology of KOZ it has been done immediately as soon as the evaluation starts.

The usage of the morphological analyzer developed in this study is not restricted with the diagnostic application. In medical area, where huge amounts of textual data present, it has a wide range of use. It may be embedded in a search engine to find any topic in a database and also be integrated to Internet for similar purposes. It may also be used as a desktop clerk in a hospital to serve the patients to the clinic related with their complaints more efficiently. Outside the scope of the medical area, it will be programmed to any NLP system.

REFERENCES

1. Heyer&Haugeneder, “Language Engineering”, Vieweg, 1995
2. Annie Gal, “Prolog for Natural Language Processing”, John Willey, Chichester-England, 1991
3. Coiera E., “Artificial Intelligence in Medicine” in Guide to Medical Informatics, the Internet and Telemedicine, Chapter 19, 1997
4. Coiera E., “The Role of Knowledge Based Systems in Clinical Practise”, Knowledge and Decision in Health Telematics-The next Decade, IOS Press, Amsterdam, pp 199-203, 1994
5. Coiera E., “Question the Assumptions”, Knowledge and Decision in Health Telematics-The next Decade, IOS Press, Amsterdam, pp 61-66, 1994
6. Tange H.,”How to Approach the Structuring of Medical Record Towards a Model for Flexible Access to Free-Text Medical Data” ,International Journal of Biomedical Computing Vol 42, Iss 1-2, pp 27-34, 1996
7. Tange , Hasman, Robbe, Schouten, “Medical Narratives in Electronic Medical Records”, International Journal of Medical Informatics, Vol 46, Iss 1, pp 7-29, 1997
8. Wagner, Lovis, Baud, Scherrer, “Medical Concept Systems, Lexicons and Natural Language Generation”, Lecture Notes in Artificial Intelligence Vol 1211, pp 398-401, 1997
9. Zweigenbaum, “Menelas- An Access System for Medical Records Using Natural-Language”, Computer Methods and Programs in Bio-Medicine, Vol 45, Iss 1-2, pp 117-120, 1994
10. Rassinoux, Michel, Juge, Baud, Scherrer, “Natural Language Processing of Medical Texts Within the Helios Environment”, Computer Methods and Programs in Bio-Medicine, Vol 45, Iss S, pp S79-S96, 1994
11. Pietrzyk, “Free Text Analysis”, International Journal of Biomedical Computing Vol 39, Iss 1, pp 139-144, 1995
12. Bassoe Carl-F., “Automated Diagnoses From Clinical Narratives: A Medical System Based on Computerised Medical Records, Natural Language Processing,

and Neural Network Technology", Neural Networks, Vol 8, No 2, pp313-319, 1995

13. Covington, M.A., "Natural Language Processing for Prolog Programmers", Prentice Hall, New Jersey, 1994
14. Güngör T., "Computer Processing of Turkish Morphological and Lexical Investigation", Ph.D. Dissertation, p. 136 Boğaziçi University, İstanbul, 1993
15. Özkan M., Külekci M.O., "Morphological Analysis Of Agglutinative Languages With A New Methodology and An Implementation Of It On Turkish", Submitted to TÜBİTAK Elektrik Journal, 2000
16. Külekci M.O., Özkan M., "KOZ: A New Morphological Analysis Method For Agglutinative Languages And An Implementation Of It On Turkish", Submitted to Association of Computational Linguists ACL'2000 Conference
17. Hankamer, J., "Finite State Morphology and Left to Right Phonology," Proceedings of the West Coast Conference on Formal Linguistics, Vol. 5, Stanford University, 1986.b
18. Hankamer, J., "Turkish Generative Morphology and Morphological Parsing," Unpublished paper Presented at Second International Conference on Turkish Linguistics, İstanbul, 1984.
19. Güngör, T. and Kuru, S., "Representation of Turkish Morphology in ATN", Proceedings of the Second Symposium on Artificial Intelligence and Artificial Neural Networks, _İstanbul, pp. 94-104, June 1993.
20. Koskenniemi, K., "Two-level Morphology: A General Computational Model for Word Form Recognition and Production", Publication No: 11, Department of General Linguistics, University of Helsinki, 1983.
21. Oflazer, K., "Two-level Description of Turkish Morphology," Proceedings of the Second Turkish Symposium on Artificial Intelligence and Neural Networks, pp. 86-93, İstanbul, 1993.
22. Türk Dil Kurumu İmla Kılavuzu
23. Pınar T., Semptoma Yönelik Tanı El Kitabı, Hacettepe Doktorlar Yayınevi, Ankara, 1994
24. Ergin, M., Türk Dilbilgisi, 2nd ed., İstanbul University Literature Faculty Publications Number :785, İstanbul, 1962.