World Applied Sciences Journal 5 (3): 311-317, 2008
ISSN 1818-4952

Can Computer Understand and Solve Turkish Arithmetic Problems?

Ünal Çakiroglu

Department of Computer and Instructional Technologies, Karadeniz Technical University,
Fatih Faculty of Education, Sögütlü, Trabzon, Turkey

Abstract: Understanding is one of the most important research areas of Natural Language Processing. Problem solving is a way of realizing machine understanding. This paper presents a problem solver system which can understand and solve arithmetic problems in Turkish. There are three main phases of the system: morphology, syntax and semantic. The difference from other systems that is; data modeled as semantic networks in semantic analysis. This type of modeling realized correct and fast understanding by reducing unnecessary data. Many of school mathematics problems can be solved by this model in Turkish Language. Performance of the system is analyzed by corresponding system's performance and primary school students' through selected problems. The system is enabling generate correct answers and necessary messages in a user friendly interface.

Key words: Natural language processing . understanding . semantic networks . school mathematics

INTRODUCTION

Turkish is an agglutinative language with respect to word structures formed by productive affixations of derivational and inflectional suffixes to root words [1]. The computer understanding concept may be explained as; rewriting entering data by its own words or response the requests about data. If the requests cannot be achieved, the reasons must be explained.

Researches about natural languages can be classified in three groups. The first group consist of syntax and semantic together for sentence analysis in human-computer interaction. In second group; data is stored as semantic networks, frames or scripts. So, the perpormance of the system is proportional to the amount of data processed by the system. The SHRDLU of Winograd and SAM of Cullingford are this kind of systems. Last group consist of human-computer dialogues related to the aims, plans, beliefs or other tendencies [2, 3].

In this paper a semantic analyzer system is designde to solve arithmetical problems written in Turkish. This study can be classified in second group of NLP studies. Before implementing an analyzer, approximately 300 problems in Turkish Primary Schools Mathematics Books are examined. Then a function is generated for understanding and solving these problems. Some examples are below:

"Ahmet'in 15 kalemi vardir. (*Ahmet has 15 pencils*.) Ahmet kaç kalem daha alirsa 20 kalemi olur? (*How many pencils should Ahmet have according to have 20 pencils.*)"

"Yasemin'in onbes cevizi vardir. (*Yasemin has fifteen walnuts.*) Mehmet bes cevize sahiptir. (*Mehmet has five walnuts.*)Yasemin Mehmet'e kaç ceviz verirse Mehmet'in 18 cevizi olur? (*How many walnuts should Yasemin give to Mehmet, so that Mehmet has got 18 walnuts.)"*

In this paper, a semantic analyzer program for solving arithmetical problems in Turkish is presented. Data is stored as semantic networks and the semantic effect of morphological structures are taken in to consideration. After examining primary school 1st, 2nd and 3rd grade school mathematics books, it is seen that most of the problems can be solved by using the basic four arithmetic operations; addition, subtraction, multiplication and division. This study includes generating and applying algorithms for solving these kind of problems. The rest of the paper is about determining the performance of the algorithms.

MATERIALS AND METHODS

The mathematical model for turkish language: It is necessary to identify all regular and irregular structures of natural language for NLP. According to achieve this process, there are some notations about languages that decides if the sentence or word structures matches language rules. Constructing a problem solver is a task of many processes that are shown on Fig. 1. First,

Corresponding Author: Dr. Ünal Çakiroglu, Department of Computer and Instructional Technologies, Karadeniz Technical University, Fatih Faculty of Education, Sögütlü, Trabzon, Turkey

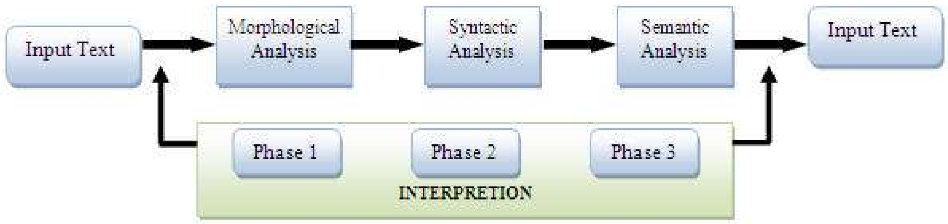

Fig. 1: Problem solver system

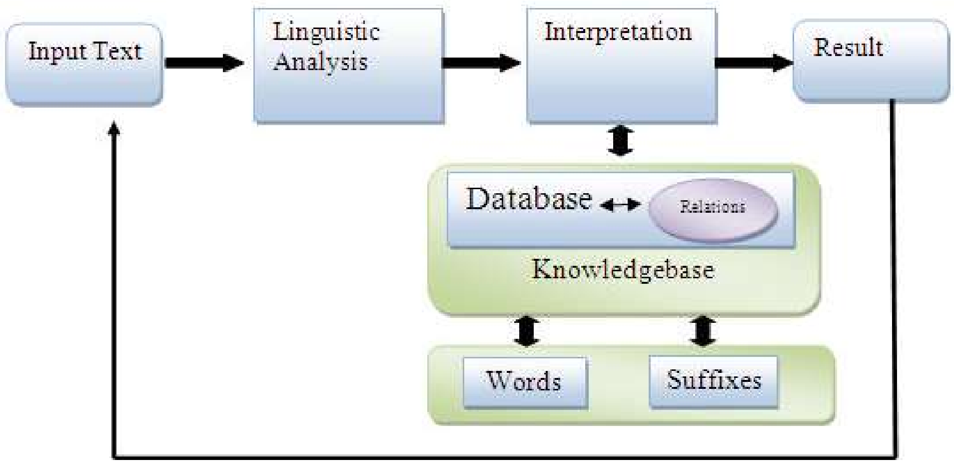

Fig. 2: Morphological analysis level

program has to deal with all steps of natural language analysis: morphology, syntax and semantic analysis [4].

Morphological analysis: The sequence of morphemes, appearing in a word is determined by morphotactics of the language in Turkish [5]. The morphological analyzer recognizes punctuations, possessives, proper names, short forms, words, roots and suffixes. Punctuations are symbols such as comma, full stop, question mark and semi-colon. The roots and words may be nouns, verbs, pronouns, adjectives, adverbs, prepositions, conjuctions, numerals and connectives. Suffixes may be derivational and inflectional [6]. Almost all of must commonly used Turkish words and suffixes are included in the system's database. In NLP studies the main aims of morphological analysis are:

Determining the types of words: In Turkish, the meaning sets of word sets are generally defined by language grammar [7, 8]. By the way, determining the word types can help to obtain some hints about the meaning sets of the words in sentences.

Searching affixes and suffixes of the words: In this process, word is divided into affix morphemes that are designated and accepted by the linguists.

Determining affix types: In Turkish, sometimes different word types can be formed by the same affixes. Ex: In the sentences "Armud*u* yedim." ("-u": (suffix of object)) and "Ali'nin armudu …" ("-u " : determinative suffix). The morphological analysis of the system is simply shown in Fig. 2.

After this phase an example morphological analysis of a problem "Ahmet'te kaç elma oldu?" (*How many apples did Ahmet have?*) text is given below:

"Ahmet'te" → Ahmet: noun(root) ; te :case (locative)
"kaç "" → adjective (root)
"elma" → noun (root)
"oldu" → ol :verb (root); du : tense (past)

Syntactic analysis: Syntactic analysis is hierarchical structure of sentence units. That is comparing the syntactic or morphological units of sentences with the hierarchical syntax rules. The most common methods for implementing syntax are:

1. Presentation of syntactic structure of language as identifiers.
2. Constructing the general structure of language on a certain grammar basics.

Before semantic analysis, the word groups must be tested if a sentence is recognized by the analyzer. If not it is called "non-sentence" (NS). In Turkish there are some difficulties in meaning extracting from NS's for everybody. It is also should be difficult for computer. So NSs did not taken into account. The meaningful

Table 1: Syntactic parsing of a Turkish sentence

Main units of sentence
<sentence>::=<subject>,<case>,<adverb>,<object>,<verb>
<subject> ::=<noun set> | <adjective set> | <noun|
<case>::= <noun set>, <affix (case)>| <adjective set>, <affix (case) >
<adverb>::= < adverbs (verb)> | <noun set>,<other adverb>
<object>::= <noun object>|<adjective object>
<verb>::= <verb>,<affix (tense)>, <affix (plural)| <noun>,<affix (verb)>
Units of Main units
<noun set>::=<determinative>,<determined>| <noun>| <pronoun>
<adjective set>::= <adjective>,<noun set>
<noun object> ::= <noun set>, <affix(object)>
<adjective object>::= <adjective set>, <affix(object)>
<determined>::= <noun>, <affix(determined)>
<determinative>::= <noun>, <affix(determinative)>| <pronoun>, <affix(determinative)>
Example Unit Words
<adverbs (time)>::= "simdi" (now) | "bugün" (today)| "yarin" (tomorrow) | "bu sabah" (this morning)| "geçen yil" (last year) | "dün" (yesterday)
<adjective>::= "beyaz" (white) | "kirmizi" (red) | "küçük" (small) | " güzel " (good)

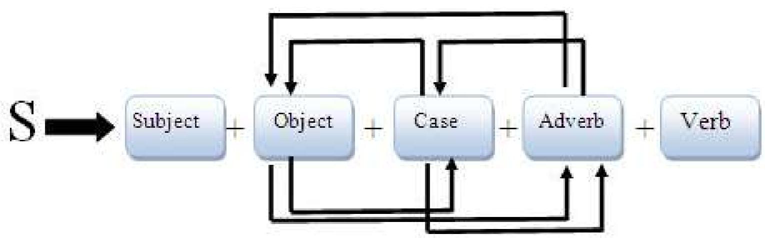

Fig. 4: Syntactic rule for a sentence

"Kirmizi elmalardan bes tane yedim."(*I ate five red apples*.)
"Kirmizi " : adjective, "elma" :noun
"-lar" :affix-plural, "-dan " : affix-case ()
"bes" : adjective, "tane " : noun
" ye" :verb, "-di" :affix-tense (past),
"-m" :affix-possession

Fig. 3: Syntactic analysis of a selected text

sentences are finite units, so they can be identified by Finite State Machines (FSM). These kinds of machines are used to determine the task of the word or word groups in the sentences, that are valuable for semantic analysis. In Turkish semantic, the considered unit of the sentence is located close to the verb of the sentence [9]. Therefore in this study the subject of the sentence is located at the beginning and the verb of the sentence is located at the end of the sentence for getting better performance in semantic analysis. The sentence structure used in this system is organised as Figure 4. In syntactic analysis Backus-Naur Form is used to identify the sentence units [10]. A small part of Turkish text is shown in Table 1. The grammar itself consist of target problem texts is a expressive subset of Turkish shown in Fig. 3. An example text parsed by syntactic analyzer is shown Fig. 3.

Semantic analysis: Although the language is finite, it consist of millions sentence alternatives. It is impossible store all words and all other units of sentences in the system database. So in this study, the necessary data is stored as semantic networks. A program is developed for constructing this network. This program forms a knowledgebase that can be thought as the main part of the system. This knowledgebase is a union of objects (words) and the types of relationships between the objects. This knowledgebase can be updated by the system it self, so the vocabularies are not so large in system database. In this module, the positions of meaningful words and suffixes are determined and the relations between these units are identified. These relations should be used in meaning extraction or solving problems. It is impossible storing all words and all other units of sentences in the system database that the necessary data is stored as semantic networks. This knowledgebase of the system is a union of objects (words) and the types of relationships between the

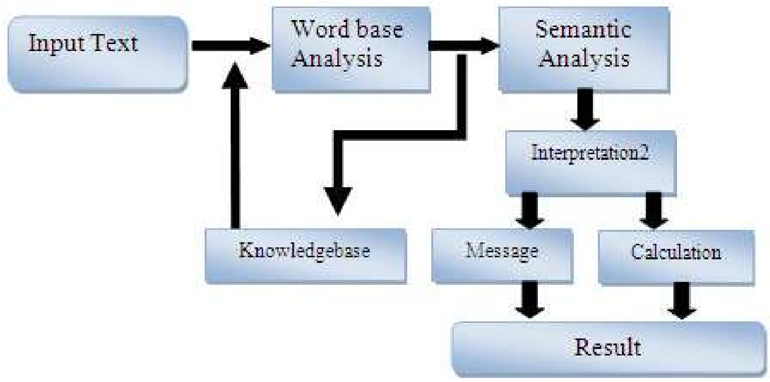

Fig. 5: Semantic analysis of the system

objects. This knowledgebase can be updated by the system, it self. So limited vocabularies are used in system database. The semantic analysis phase of the system is basically presented in Figure 5.

The primary school 1, 2, 3rd grade mathematics books are examined and then the problem texts aregrouped as addition, substraction, division, multiplication, etc. We called these groups as operation groups (OG).

For example, " Züleyha 15 balona sahiptir. Cemal'in 5 balonu vardir. Cemal balonlarindan üçünü Züleyha'ya verirse, Züleyha'nin kaç balonu olur? (*Züleyha has 15 balloons, Cemal has 5 balloons. How many balloons does Züleyha have if Cemal gives 3 of his balloons to Züleyha?*) " This type of problems can be defined as:

X1-Y1=F1, X2-Y2= F2, Here ;
X1:The amount of first object at the beginning
X2:The amount of second object at the beginning
Y1:The decreasing amount of first object
Y2:The increasing amount of second object
F1: The amount of first object at the end
F2: The amount of second object at the end

All types of problems are grupped like above as a subset of Turkish. In all types of problems, many kinds of problems can be obtained by changing the "given" and "wanted" parts of the problems. It is seen in mathematics books that; there are many alternatives of problem texts that can be solved by multiplication and division. One of the disadvantages is; this kind of problems consist different structures and different kinds of word sets. So many of the word sets must be identified to the system for solving multiplicative and division problems. For example, "Mehmet 25 kalemini 5 arkadasina bölüstürdü..." (*Mehmet divided his 25 pencils to his 5 friends...*), "Hasan'in 20 çikolatasi vardir. 4 kardesinin herbirine beser çikolata verdi. " (*Hasan has 20 chocalates. He gave five chocalates to his 4 brothers...*)

Table 2: A piece of knowledge base of the system

Related words	Related units	Relation
"var"		
"sahip olmak"		
"mevcut olmak"		
"elde olmak"...	Subject-object	owner
"almak"		
"eklemek"		
"artmak"..	Subject-object	Add
"vermek"		
"azalmak"		
"yirtmak"		
"eksilmek"...	Subject-object	decrease
"olmak"		
"kalmak"		
"bulunmak"..	Subject-object	Have
1,2,3,…	Object-noun	quantity

So, many word sets must be identified for our system for solving these kinds of sentences. So the number of OGs is proportional to the performance of our system.

Semantic Analysis phases are shown below:

Word base analysis: In this phase the sentence is parsed word by word and the meaning of words are searched by word-suffix matching,

Interpretation1: This is a test module, that tests if the words are in knowledge base or not. If not, there is a feed back module for alternatives.

Knowledgebase: Here, the relationships between words and word duties are identified. Here is a piece of knowledge base presentation shown in Table 2.

Semantic networks: This phase is the main part of the module and shown in Figure 6. The relations of the words are constructed and the types of relations are identified in this module. The sentence ""Bir dükkanda

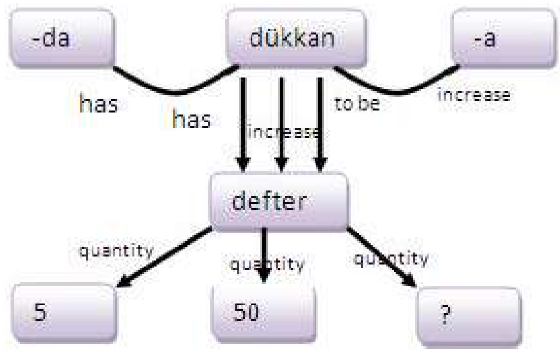

Fig. 6: A semantic network for selected problem

50 defter var. Dükkana 5 defter alindi. Dükkanda kaç defter oldu? (There are 50 notebooks in a shop. If 5 are bought, how many notebooks will be in the shop?) " has a network shown below:

In the problem solving phase, all the wanted and given data classified in the problem types about the selected problem.

The implemented system can identify subjects, objects, cases, adverbs and verbs of the sentences by semantic networks [11]. That is very useful for determining words as "given" or "wanted" data.

Interpratation2: In this module the answer of problem is communicated to the user in an easily comprehensible fashion. Also if there are logical mistakes in the input problem texts, system presents these and addresses the location of mistakes. Example:

Input Text:.Davut'un 15 oyuncagi var. Hasan'in 5 meyvesi vardir. Toplam kaç armut vardir? (*Davut has 15 toys. Hasan has 5 fruits. How many pears are there?*)

Program Message: " 'toys' and 'fruits are not same kind of objects, can not do operation!'"

Program can also gives correct answers to same kinds of object sets like:

Input Text:. Bülent'in 32 *elmasi* var. Hasan'in 5 seftalisi vardir. Toplam kaç *meyve* vardir? (*Bülent has 32 apples. Hasan has 5 peachess. How many fruits are there?*)

Program Message: " There are 37 fruits.'"

Some selected problems given to the system as input problems are;

1. Ali'de 5 elma vardir. Ali 4 elma daha aldi. Ali'nin kaç elmasi oldu? (*Ali has got 5 apple. Ali bought 4 more apples. How many apple has Ali got now?)*
2. Fatma'da 88 findik var. 10 daha alirsa kaç findigi olur? *(Fatma has got 88 nuts. If she takes 10 more, how many nut has she got?*) Here there is no object after "10" in "10 daha …" but by the word "daha" (more) program understands ; "10 daha " means "10 more nuts". There are many kind of missing words (missing objects, missing nouns, etc.) Program can tolerate these kinds of missing.
3. Davut'un 15 oyuncagi var. Hasan'in 5 meyvesi vardir. Toplam kaç armut vardir? (*Davut has 15 toys. Hasan has 5 fruits. How many pears are there?*)
4. Hüseyin'de 504 elma vardir. 412 elmayi atti. Kaç elmasi kaldi? (*Hüseyin has 504 apples. He threw 412 of them. How many apple has he got now*?)
5. Ayse sekiz çilek verirse on bes çilegi kaliyor. Ayse'de kaç elma vardir? (*If Ayse gives 8 strawbery, she will have 15 strawberry. How many apples has Ayse got?.*)
6. Hasan 5 bilye satin alirsa 3 bilyesi olur. Hasan'da kaç bilye vardir? (*If Hasan buys five alleys,he will have three alleys. How many alleys has hasan got?*)
7. Ali'de 8 armut vardir. Hasan'da 10 portakal vardir. Toplam kaç meyve vardir? (*Ahmet has 8 pears. Hasa has 10 oranges. How many fruits are there?*)
8. Bir kirtasiyeci yüz yirmi iki kalem alirsa bes yüz kalemi oluyor. Kaç kalemi vardir? *(If a retailer buys one hundred and twenty two pencils, he will totally have five hundred pencils. How many pencils did the retailer have initially?)*
9. Ahmet'te 10 defter mevcuttur. Kemal Ahmet'e 3 verdi. Ahmet'in kaç defteri oldu? (*Ahmet has 10 pencils. Kemal gave 3 to Ahmet. How many notebooks has Ahmet got?)*Here in "… Ahmet'e 3 verdi…" there is a missing noun "kalem" (pencil) after 3 but program can tolerate it.

In order to testing the program, another part of a study is realized in a Turkish primary school 1st, 2nd and 3rd grade students who are 6,7,8 years old. For this study every kinds of problem texts are tried to be selected for comparing all kinds of texts. Here are some examples about the problems.

"Ali 5 elma daha alirsa 9 elmasi olur. Ali'de kaç elma vardir?" (*If Ali takes 5 apples, the number of his apples become 9. How many apples does Ali have?*)

"Bir ögrenci günde 5 sayfa kitap okuyor. 10 günde kaç sayfa okur?" (*A student reads 5 pages of a book in a day. How many pages does he read in 10days?*)

"Bir çiftlikte 200 tavuk mevcuttur. Otuz bes tavuk ölürse kaç kalir?" (*There are 200 chicken in a farm. If thirtyfive die, what will be the chicken population?*)

"Bir kirtasiyeci yüz yirmi iki kalem alirsa bes yüz kalemi oluyor. Kaç kalemi vardir?" (*If a retailer buys one hundred and twenty two pencils, he will totally have five hundred pencils. How many pencils did the retailer have initially?*)

RESULTS

In this paper an implementation of a problem solver problem is presented. This method describes how to use semantic networks for storing data. And how to obtain relations by semantic and morphological

Table 3: System's answers to the 20 selected questions

1	2	3	4	5	6	7	8	9	10
T	T	F	T	T	F	T	T	T	T
11	12	13	14	15	16	17	18	19	20
T	T	F	F	T	T	T	T	T	T

T: True, F: False

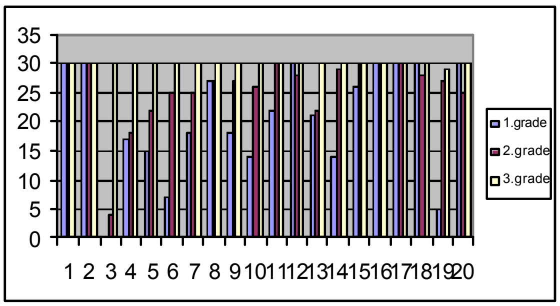

Fig. 7: Students and system comparison about selected problems

structures of the language. Experimental results are indicating that the proposed program has sufficient power for solving problems. According to selected problems, the primary school students and the system performance obtained by calculating the average correct answers, through 20 problems. The performance of the students analyzed through selected problems are shown in Figure 7. All grades consist of 30 students:

The system's answers are listed in Table 3. As shown in Table 3, performance of the system is higher than 1st and 2nd grade students and it is close to the 3rd grades for selected 20 problems. The system performance, in addition and subtraction problems if 98%. These kinds of problems are classified in X+Y=F, X-Y= F and other variances of addition and subtraction groups that the reasons are explained above. The system performance is 61% in multiplication and division problems through selected 400 problems.

DISCUSSION

Nowadays human-computer interactions become an important part of human life and provide useful information for people in daily life. In this paper, problem solving as a natural language problem in school mathematics is taken a base study.

In this study the words and relationships in knowledgebase is limited. In future work implementation of knowledge system would be employed to maximize the words and relationships. The various phases of the program are developped quite independent each other. So they can be adapted to the similar NLP studies. The statistics that are made for the sentence structure and word orders in Turkish will provide important conclusions for a general understanding model. This system can help solving arithmetical problems and also learning Turkish grammatical rules. With no doubt Turkish language have exceptions in addition to its rules. Therefore it would be appropriate to add such exceptions for the newly constructed model. Also the model can be applied on educational web contents [12].

With no doubt Turkish language have exceptions in addition to its rules. Therefore it would be appropriate to add such exceptions for the newly constructed model. The results obtained in this work can be used in research which aims to make reasonings, to offer alternatives by semantic networks.

REFERENCES

1. Güngördü, Z. and K. Oflazer, 1995. Parsing Turkish using the Lexical-Functional Grammar Formalizm. Machine Translation", 10: 4.
2. Bozsahin, C., D. Zeyrek and I. Pembeci, 1998. Computer-aided Learning of Turkish Morphology. Proceedings of Joint Conf. of ACH and ALLC, Debrecen, Hungary.
3. Rich, E., 1991. Artificial Intelligence, McGraw Hill Inc., Second Edition, Newyork.
4. Budy, L. Byrd, G. Luger, C. Mellish, R. Milne and M. Palmer, 1979. MECHO: A program to solve mechanics problems. Working Paper 50, Department of Artificial Intelligence, Edinburgh University.
5. Pakka, M. and C.C. Say, 1994. Qualitative collision detection in the roller world. Proc. Ninth Turkish Symposium Computer and Information Sciences, Antalya, pp: 550-557.
6. Eryigit, G. and E. Adali, 2004. An Affix Stripping Morphogical Analyser For Turkish. Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria,
7. Schank, R.C. and R.P. Abelson, 1977. Scripts, Plans, Goals and Understanding, Lawrence Erlbaum, Hillsdale, NJ, James, Natural Language Understanding(2nd Edn.). The Benjamin/Cummings Publishing Company, Inc., 1995.
8. Arican, K. and T. Yemliha, 1993. Synonyms and antonyms dictionary for Turkish. B.S. Thesis, Dept. of Computer Engineering, Bogaziçi University,
9. Gazdar, G. and C. Mellish, 1989. Natural Language Processing in Prolog: An Introduction to Computational Linguistics, Addision-Wesley, Workingham.

10. Lee, M.C. and R.V. Leong, 1992. NLUS-A Prolog Based Natural Language Understanding System. In Proceeding of the 4th International Conference on Computing and Information, pp: 204-207.
11. Woods, W.A., 1970. Transition network grammars for natural language analysis. Communications of the ACM, 13: 591-606.
12. Keyhanipour, A..H., M. Piroozmand and K. Badie, 2009. A Neural Framework for Web Ranking Using Combination of Content and Context Features. World Applied Sciences Journal, 6 (1): 6-15.