PARSING AGGLUTINATIVE WORD STRUCTURES AND ITS APPLICATION TO SPELLING CHECKING FOR TURKISH

by

AYŞIN SOLAK and KEMAL OFLAZER
Department of Computer Engineering and Information Sciences
Bilkent University
Bilkent, Ankara, 06533 Türkiye

ABSTRACT

Most of the research on parsing natural languages has been concerned with English, or with other languages morphologically similar to English. Parsing agglutinative word structures has attracted relatively little attention most probably because agglutinative languages contain word structures of considerable complexity, and parsing words in such languages requires morphological analysis techniques. In this paper, we present the design and implementation of a morphological root-driven parser for Turkish word structures which has been incorporated into a spelling checking kernel for on-line Turkish text. The agglutinative nature of the language and the resulting complex word formations, various phonetic harmony rules and subtle exceptions present certain difficulties not usually encountered in the spelling checking of languages like English and make this a very challenging problem.

1. Introduction

Morphological classification of natural languages according to their word structures places languages like Turkish, Finnish, and Hungarian to a class called "agglutinative languages". In such languages, words are combination of several morphemes. There is a root and several suffixes are combined to this root in order to modify or extend its meaning. What characterizes agglutinative languages is that stem formation by affixation to previously derived stems is extremely productive. A given stem, even though itself quite complex, can generally serve as basis for even more complex words. Consequently, agglutinative languages contain words of considerable complexity, and parsing such languages necessitates a thorough morphological analysis.

Morphological parsing has attracted relatively little attention in computational linguistics. The reason is that nearly all parsing research has been concerned with English, or with languages morphologically similar to English. Since in such languages words contain only a few number of affixes, or none at all, allmost all of the parsing models for them consider recognizing those affixes as being trivial, and thus do not require a morphological analysis. In agglutinative languages, words contain no direct indication of morpheme boundaries which are in general dependent on the morphological and phonological context. A morphological parser requires a morphophonological component which mediates between the surface form of a morpheme as encountered in the input text and the lexical form in which the morpheme is stored in the morpheme inventory, i.e., a means of recognizing variant forms of morphemes as the same, and a morphotactic component which specifies which combinations of morphemes are permitted [7].

Morphological parsing algorithms may be divided into two classes as *affix stripping* and *root-driven* analysis methods. Both approaches have been used from very early on in the history of morphological parsing. For instance, Packard's parser for ancient Greek [15], and Brodda and Karlsson's for Finnish [3] used affix stripping. Sagvall, on the other hand, devised a root-driven morphological analyzer for Russian [17]. In addition, other root-driven morphological parsers for the agglutinative languages Quechua [9, 10], Finnish [11], and Turkish [6] were developed independently in the early 1980's. All of these three parsers proceed from left to right. Roots are sought in the lexicon that match initial substrings of the word, and the grammatical category of the root determines what class of suffixes may follow. When a suffix in the permitted class is found to match a further substring of the word, grammatical information in the lexical entry for that suffix determines once again what class of suffixes may follow. If the end of the word can be reached by iteration of this process, and if the last suffix analyzed is one which may end a word, the parse is successful [7].

Another left-to-right parsing algorithm for automatic analysis of Turkish words was proposed and applied by Köksal in his Ph.D. thesis [12]. His algorithm called "Identified Maximum Match (IMM) Algorithm", tries to find the maximum length substring, which is present in a root dictionary, from the left of the word. If a solution is obtained, i.e., the root morpheme is identified, the remaining part of the word is considered as the search element. This part is looked for in the suffix morpheme forms dictionary and the morphemes are identified one by one. The process stops when there is no remaining part. However in some cases, although a solution is obtained further consistency analysis proves that this solution is not the correct one. In such cases the previous pseudo-solution is reduced by one character and all the search procedure is initiated once more.

These approaches to morphological parsing of Turkish words have the following shortcoming: They do not consider the fact that in Turkish, words contain tremendous amount of semantic information that has to be taken into account. In these parsers, it is only the grammatical category of the stem that determine the suffixes that may follow. However, most of the suffixes in Turkish, especially the derivational ones, can be attached only to a limited number of roots or stems mostly due to semantic reasons.

Another shortcoming of the previous parsers for Turkish is that they allow the iterative usage of derivational suffixes. Although, Köksal [12], prevents the consecutive usage of the same morpheme twice, he still parses the word GÖZLÜKÇÜLÜKÇÜLÜK correctly, so do Hankamer [7]. It is true that some Turkish suffixes can form an iterative loop, but usually the number of iteration is not too high. The above word can be parsed correctly up to the point GÖZLÜKÇÜLÜK (the occupation of oculists), but the words GÖZLÜKÇÜLÜKÇÜ and GÖZLÜKÇÜLÜKÇÜLÜK are meaningless, and therefore some control mechanisms using semantic information should be included within the parser to avoid parsing such meaningless words as if they were correct.

One of the most important application areas of parsing words in natural languages is checking their spellings. Although many spelling checkers for English and some other languages have been developed, so far no such tool was present for Turkish. The reason for this is probably the complexity of parsing problem for Turkish as explained above. Wrong ordering of morphemes and errors in vowel or consonant harmonies may cause the wrong spelling of Turkish words. Consequently, in order to check the spelling of a Turkish word, it is necessary to make significant phonological and morphological analyses.

This paper describes a morphological root-driven parser developed for Turkish language and its application to spelling checking. A major portion of this work depends on a detailed and careful research on some features of Turkish that make the parsing problem for this language especially hard and interesting. The following section presents an overview of certain morphophonemic and morphological aspects of the Turkish language which are especially relevant to the problem under consideration (for details see [20]).

2. The Turkish Language

Turkish is an agglutinative language that belongs to a group of languages known as Altaic languages. For an agglutinative language, the concept of word is much larger than the set of vocabulary items. Word structures can grow to be relatively long by addition of suffixes and sometimes contain an amount of semantic information equivalent to a complete sentence in another language. A popular example of complex Turkish word formation is ÇEKOSLOVAKYALILAŞTIRAMADIKLARIMIZDANMIŞSINIZ whose equivalent in English is "(it is speculated that) you had been one of those whom we could not convert to a Czechoslovakian." In this example, one word in Turkish corresponds to a full sentence in English. Each suffix has a certain function and modifies the semantic information in the stem preceding it. In our example, the root morpheme ÇEKOSLOVAKYA is the name of the country *Czechoslovakia* and the suffix -LI converts the meaning into *Czechoslovakian*, while the following suffix -LAŞ makes a verb from the previous stem meaning *to become a Czechoslovakian*,[1], and so on.

2.1. Turkish Phonetic Model

Being phonetic, the Turkish language can be adapted to a number of different alphabets. In the past, various alphabets have been used to transcribe Turkish, e.g., Arabic. Since 1928, Latin characters have been used. The Turkish alphabet consists of 29 letters of which 8 (A, E, I, İ, O, Ö, U, Ü) are vowels, and 21 (B, C, Ç, D, F, G, Ğ, H, J, K, L, M, N, P, R, S, Ş, T, V, Y, Z) are consonants.

Turkish word formation uses a number of phonetic harmony rules. Vowels and consonants change in certain ways when a suffix is appended to a root, so that such harmony constraints are not violated.

2.1.1. Vowel Change in Suffixes

Almost all suffixes in Turkish use one of two basic vowels and their allophones. We have denoted these sets of allophones with braces around the main vowels A and I, as {A} and {I}. The allophones of {A} are A and E, where {I} represents I, İ, U, or Ü. The vowels O and Ö are only used in root morphemes (especially in the first syllable) of Turkish words.[2]

The vowel harmony rules require that vowels in a suffix change according to certain rules when they are affixed to a stem. The first vowel in the suffix changes according to the last vowel of the stem. Succeeding vowels in the suffix change according to the vowel preceding it. If we denote the preceding vowel (be it in the stem or in the suffix) by v then {A} is resolved as A if v is A, I, O, or U, otherwise it is resolved as E. On the other hand, {I} is resolved as I if v is A or I, as İ if v is E or İ, as U if v is O or U, and as Ü if v is Ö or Ü. For example the word "YAPMAYACAKTINIZ" can be broken into suffixes as:

YAP/M{A}/[Y][3]{A}C{A}{K}[4]/{D}[5]{I}/N{I}Z

[1] From now on, we will indicate the English meaning of a word in Turkish in parentheses following it.

[2] The progressive tense suffix -{I}YOR is an exception.

[3] [ ] indicates an optional morpheme that must be inserted before a suffix to satisfy certain harmony rules. In this case, [Y] indicates that the consonant Y must be inserted if the last letter of the stem is a vowel, otherwise it is dropped: e.g., OKU (read) → OKUYACAK (s/he will read), but SOR (ask) → SORACAK (s/he will ask).

[4] The two allophones of {K} are K and Ğ.

[5] The two allophones of {D} are D and T.

It can be seen that the vowels in the correct spelling of the word obey the rules above, while a spelling like "YAPMAYACEKTİNİZ" violates the harmony rules because an {A} in the suffix can not resolve to an E as the preceding vowel is an A. It should be mentioned in passing that there are also some suffixes, such as -KEN, whose vowels never change.

2.1.2. Consonant Harmony

Another basic aspect of Turkish phonology is consonant harmony. It is based on the classification of Turkish consonants into two main groups, *voiceless* and *voiced*. The voiceless consonants are Ç, F, T, H, S, K, P, Ş. The remaining consonants are voiced. Interested readers can find the complete list of consonant harmony rules in Koksal [12], and Solak [20]. To give an example, one of the rules says that if a suffix begins with one of the consonants D, C, G, this consonant changes into T, Ç, K respectively, if a voiceless consonant is present as the final phoneme of the previous morpheme, e.g., YOLDA (on road), but UÇAKTA (on plane).

Some morphemes are affixed with the insertion of either N, S, Ş, Y when two vowels happen to follow each other (e.g. BAHÇESİ (his/her garden), BAHÇEYİ (accusative of garden), İKİŞER (two each)), or when there is another morpheme following (e.g. BAHÇESİNDE (in his/her garden), or in context of some pronouns (e.g., BUNA (to this), KENDİNDEN (from yourself)) and the pronomial suffix -Kİ (e.g. SENİNKİNİ (accusative of yours)). In our example above, the future tense suffix [Y]{A}C{A}{K} comes after the stem YAPMA and since the last phoneme is a vowel Y is inserted.

2.1.3. Deformation of Roots

Normally Turkish roots are not flexed. However, there are some cases where some phonemes are changed by assimilation or various other deformations [12]. An exceptional case related to the flexion of roots is observed in personal pronouns BEN (I) and SEN (you) having datives BANA (to me) and SANA (to you) respectively. These are individual cases and can be treated as exceptions.

A more systematic ellipsis occurs when the suffix -{I}YOR comes after the verbal roots and stems ending with the phoneme {A}. In such cases, the wide vowel at the end of the stem is narrowed, e.g., YAP → YAPIYOR (s/he/it is doing [it]), but ARA → ARIYOR (s/he/it is searching).

Another root deformation occurs as a vowel ellipsis. When a suffix beginning with a vowel comes after some nouns, generally designating parts of the human body, which has a vowel {I} in its last syllable, this vowel drops, e.g. BURUN (nose) → BURNUM (my nose). Similarly, when the passiveness suffix {I}L is affixed to some verbs, whose last vowel is {I}, this vowel also drops, e.g. ÇAĞIRMAK (to call) → ÇAĞRILMAK (to be called). Other root deformations and their exceptions can be found in Solak [20].

2.2. Turkish morphology

Turkish roots can be classified into two main classes: *nominal* and *verbal*. The verbal class comprises the verbs, while nominal class comprises nouns, pronouns and adjectives, etc. The suffixes that can be received by either of these groups are different, i.e., a suffix which can be affixed to a nominal root can not be affixed to a verbal root with the same semantic function.

Turkish suffixes can be classified as *derivational* and *conjugational*. Derivational suffixes change the meaning and sometimes the class of the stems they are affixed, while a conjugated verb or noun remains as such after the affixation. Conjugational suffixes can be affixed to all of the roots in the class that they belong. On the other hand, the number of roots that each derivational suffix can be affixed changes. The nominal model

The simplified models for nominal and verbal grammars can be given as follows:[6]

The nominal model:

nominal root + plural suffix + possessive suffix + case suffix + relative suffix

The verbal model:

verbal root + voice suffixes + negation suffix + compound verb suffix + main tense suffix + question suffix + second tense suffix + person suffix

3. Implementation

We have implemented a root-driven morphological analyzer for Turkish and used it as a *spelling checking kernel* that can be integrated to different applications on a variety of platforms.

The program takes a list of Turkish words as input, and then checks them one by one in the order they appear. If the spelling of an input word is incorrect, it is output as misspelled. Each word is analyzed individually with no attention to the semantics or to the context. If a word is spelled correctly but is the wrong word in the context, we have no intention for, and way of flagging it as erroneous. Thus, as in all other spelling programs, the text is examined with respect to words, not with respect to sentences. In addition, we do not yet give any suggestion about the most likely correct words after detecting a misspelled word, i.e., spelling correction is not done. Word analysis is handled in four step as syllabification check, root determination, morphophonemic check, and morphological analysis. During these steps a dictionary of Turkish root words, and a set of rules for Turkish syllable structure, morphophonemics, and morphology are used concurrently. All these steps will be explained in the following sections, after a

[6] Refer Solak [20] for detailed information on each of the suffixes in these models and the exceptional cases about them.

brief information on the dictionary used in this implementation.

3.1. Dictionary

The dictionary is based on the Turkish Writing Guide [23] as the source. Some words in the dictionary have to be marked as having certain semantic and structural properties such as being a verbal root or a nominal root, being a proper noun, not obeying to vowel harmony rules, deforming under certain conditions, and so on. For example, the word BURUN (nose) have to be marked as being a nominal root, and deforming by vowel ellipsis. For this reason, for each word in the dictionary a series of flags representing certain properties of that word are held. Thus, each entry of the dictionary contains a word in Turkish and a series of flags showing certain properties of that word.

Nearly 23,500 words, each having 7 letters on the average, are listed in our current dictionary. 41 flags per word[7] have been used so far, but later it may be necessary to use more. Because of this, two long integers (whose bits represent flags, for a total of 64 flags) are assigned for every word.

3.2. Syllabification Check

Analyzing all the words in Turkish Writing Guide [23] and all the suffixes in Turkish [1, 8], we have constructed a regular expression and a corresponding finite state automaton for validating if a word matches the syllable structure rules of Turkish [18]. This regular expression is used as a heuristic in our spelling checker. The input word is first processed with the regular expression. It is reported as misspelled if its syllable structure can not be matched with this expression, i.e., the phonemes of the word do not form valid sequences according to Turkish syllable structures. On the other hand, if it can be matched, it is further analyzed as it may still be a non-Turkish or a misspelled word.

With the help of the syllabification check, most of the typographical errors can be detected. For example, if the word YAPMAK (to make) were typed as YPMAK or YAPMKA, the word would not be matched by the expression and its spelling would be reported incorrect. On the other hand, if it were written as YAPMEK, where a vowel harmony error is made, it would pass the syllabification check, but would be reported as misspelled during morphophonemic checks.

3.3. Root Determination

Before analyzing the morphophonemic and morphological structures of a Turkish word, the root has to be determined. If the word passes the syllabification check, its root is searched in the dictionary using a maximal match algorithm. In this algorithm, first the whole word is searched in the dictionary. If it is found then the word has no suffixes and therefore its spelling is correct. Otherwise, we remove a letter from the right and search the resulting substring. We continue this by removing letters from the right until we find a root. If no root can be found although the first letter of the word is reached, the word is reported as misspelled.

The maximum length substring of the word that is present in the dictionary is not always its root. If further analyses show that the word is misspelled, a new root is searched in the dictionary, this time removing letters from the end of the previous root. If a new root can be found the same operations are repeated, otherwise the word is reported as misspelled.

Root determination presents some difficulties when the root of the word is deformed. For the root words which have to be deformed during certain agglutinations, a flag indicating that property is set in the dictionary. For example, the root of the word ŞEHRE (to the city) must be found as ŞEHİR (city). In order to determine it correctly, when the substring ŞEHR is not found in the dictionary, considering that it may be a deformed root by vowel ellipsis, the vowel İ is inserted between the consonants H and R, and the word ŞEHİR is searched in the dictionary. When it is found, the flag corresponding to vowel ellipsis is checked. Since it is set for this word, the root of the word ŞEHRE is determined as ŞEHİR, and remaining analyses are continued. If that word were written as ŞEHİRE, we should report it as incorrect although ŞEHİR + dative case suffix form looks correct. For all other root deformations, the real root of the word can be found by making such checks and some necessary changes (see [20]).

For some roots both of the forms above are valid. For example, both METNİ (accusative of text) and METİNİ (accusative of strong) are correct although the root of both words is METİN (text, strong) because this word can be used in two different meanings.

3.4. Morphophonemic Check

Turkish words obey vowel and consonant harmony rules during agglutination (see sections 3.2.1 and 3.2.2). The vowel harmony check may be done just after the root determination, but other morphophonemic checks should be done during morphological analysis.

After the root of the word is found, the rest of the word is considered as its suffixes. The first vowel in the suffixes part must be in harmony with the last vowel of the root, while the succeeding vowels must be in harmony with the vowel preceding them. Since there are some suffixes, such as -KEN, whose vowels never change, when a disharmony is found, we check whether it is the result of such a suffix (e.g., YANARKEN (while it is burning)).

[7] The list of all flags can be found in Solak [20].

Some words of foreign origin do not obey vowel harmony rules during agglutination (e.g., KONTROL (control)). Before the vowel harmony checks are done, the flag corresponding to that property must be checked. If it is set for the root of the word, the vowel harmony check must be applied inversely. Thus, the first vowel in the suffixes part must be in disharmony with the last vowel of the root (e.g., KONTROLLER (controls)). As another interesting case, some roots that may be used in two meanings, i.e., the homonyms, obey vowel harmony rules when they are used with a certain meaning, while they do not obey them when they are used in the other meaning. For example, both SOLA (to the left) and SOLE (to the note sol) pass the vowel harmony check since their root SOL has two meanings as "left" and "musical note."[8]

The suffixes must be determined before the consonant harmony checks are done. Because of this, these checks are done during morphological analysis, after each suffix is isolated.

If a word does not pass any of the morphophonemic checks, considering the possibility that the root may have been determined wrongly, a new root is searched in the dictionary.

3.5. Morphological Analysis

The spelling checker has two separate set of rules for the two main root classes. For the implementation of the lexical analyzers and parsers in which the rules are included, two standard UNIX utilities, *lex* and *yacc*, have been utilized respectively [13]. *Lex* is used to separate the suffixes of a word from left to right, and *yacc* is used to parse those suffixes using morphological rules of Turkish grammar.

The models given in various books on Turkish grammar [1, 2, 4, 5, 14] and previous research on Turkish computational linguistics [12, 16] have been utilized in for generating the rules used in the parsers. Additionally, all of the known exceptional cases have also been considered (see [20]). Although all the conjugational suffixes have been included into the rules, only a small subset of the derivational suffixes have been handled. The reasons for this are that majority of the derivational suffixes may be received by only a small group of roots, and determining such groups is a rather difficult and time-consuming job, and depends on various semantic criteria. The derivational suffixes that may be affixed to all of the roots in a class and those which can be affixed to large percentage, but not all, of the roots in their class are included in the rules. That makes it possible to eliminate a number of words from the dictionary.

The two parsers are alternatively used. First parser to be used is determined according to the class of the root, but as the parsing continues it may be necessary to switch from one parser to another and continue there, or again pass back to the previous one, since the class of a stem can change when it receives certain suffixes. The switches between parsers can sometimes be very complicated. Some suffixes can have two different usages. In such cases both possibilities have to be considered.

If a word has received more than one derivational suffixes then many switches between parsers will be necessary. For example, the root of the word BEYAZLAŞTIRMAYANLARDAN (from those which do not cause to become white) is found as the noun BEYAZ (white) in our dictionary. Then comes the suffix -L{A}Ş, which makes a verb from a noun, therefore a switch to the verb parser has to be made. Parsing continues there until the suffix -M{A} is matched. This suffix can either make a verb a noun or negate it. First considering the possibility that it is used as a derivational suffix, the noun parser is invoked. The remaining part of the word can not be parsed by this parser. So accepting -M{A} as the negation suffix, the verb parser is returned to and parsing continues there. Later comes the suffix -[Y]{A}N, which is a suffix that makes a noun from a verb, so again a switch to the noun parser is made. Continuing in this parser, the word is parsed correctly.

Some Turkish roots can take the suffixes belonging to both nominal or verbal classes. For such roots if parsing is unsuccessful in the first parser chosen, the other one must also be tried. For example, the root of the word AÇLAR (hungry people) is AÇ. This root may either be used as a verb (open) or as a noun (hungry). If parsing is first attempted with the verbal parser it will be unsuccessful. So we backtrack and use the nominal parser. With the nominal parser the word can be parsed successfully.

Figure 1 shows the block diagram of the word analysis. Summarizing, first the syllable structure of the word is checked. If it is wrong the word is added into the output list of misspelled words, otherwise the root is determined. If no root can be found the word is reported as misspelled. If a root is found, first the vowel harmony check is done. Then, according to the class of the root, one of the parsers is activated. In the parsers, as the suffixes are isolated one by one, necessary morphophonemic checks are done. Depending on the suffixes, switches between the parsers are possible. When the end of the word is reached, if no errors can be found then the spelling of the word is correct. If any error is found in any of the parsers or during morphophonemic checks, a new root is searched. If another root is found same operations are done. If no successful parsing can be done although the first letter of the word is reached, the word is added into the output list.

4. Performance Results

This spelling checker has been implemented in UNIX environment, on SUN SPARC workstations,

[8] The word SOL is pronounced slightly different in the latter.

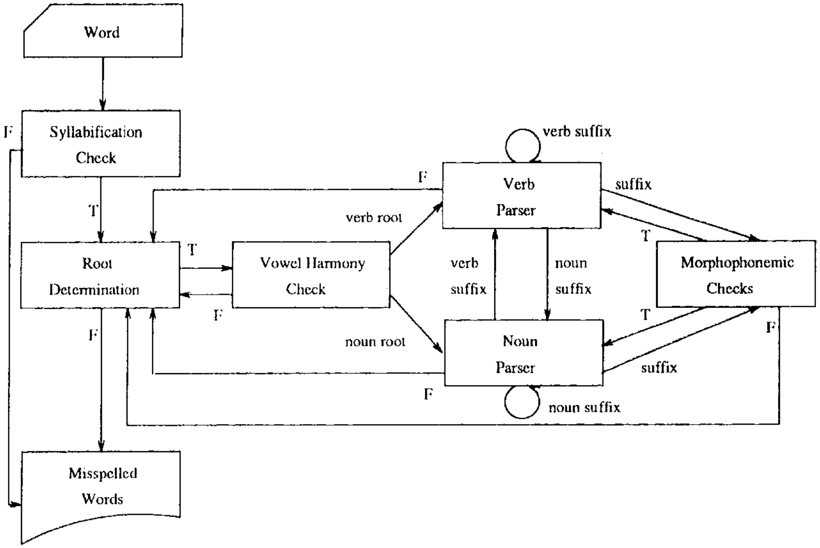

Figure 1: Word analysis

at Bilkent University, using the C programming language. Its current version takes nearly 600 Kbytes including the dictionary.

The checker can be inserted to different word processing applications or can be used separately. We have integrated it to GNU-EMACS text editor for use on LaTeX documents. In this form, the program is available for use within the university and around a number of sites on Internet. It is also possible to obtain some statistical information by running the program with -s option.

Our results indicate that the number of distinct words within a document is relatively small, and more particularly, the percentage of distinct words to total words processed increases as the length of the document decreases. Approximately 40% of the misspelled words are detected by syllabification check and the rest are detected by other checks. The number of distinct words affect the execution time more than the total number of words, as expected, because a word is fully analyzed only once. If it occurs again in the text, the result of the previous check is used. In general, the spelling checker can process at 1000-3000 words (roughly 2-6 pages) per second, depending on the document. The functional performance of the spelling checker can be fine tuned by analyzing the word list and inserting the additional appropriate flags.

5. Conclusions

In this paper, we have presented a morphological parser for an agglutinative language, Turkish, and its application to spelling checking of this language.

Parsing agglutinative word structures necessitates rather nontrivial phonological and morphological analyses which present special difficulties in the development of parsers for such languages, not usually encountered in parsers for other languages. As a result, the number of parsers developed for agglutinative languages, and particularly for Turkish, is quite limited, and they have certain shortcomings. We have solved most of the problems encountered in the previous parsers by making a detailed and careful research on Turkish word formation rules and their exceptions [20]. These results may hopefully be helpful for future researchers on Turkish linguistics. We should note that even though it is claimed that word formation rules in Turkish are well–defined and Turkish is a very regular language, as used today it shows many irregularities that cause the problem of parsing this language to become a very hard and interesting problem.

Many grammar books have been referred to collect Turkish word formation rules. In those books, after each rule is defined, usually it is reminded that there may occur some exceptions to that rule in some conditions, but mostly those conditions can not be "well" defined. For example, in all Turkish grammar books, it is said that "When a Turkish word ending with one of the consonants P, Ç, T, K receives a suffix beginning with a consonant, that final consonant is softened, but there are some such words whose final consonant does not change." However, none of the books says what the common property of those words which do not obey to that rule is, because most probably it is not known yet. In order to include that rule correctly in the parser, all words having the indicated property have been examined, the list of the irregular ones have been obtained, and special checks have been done to catch those irregularities. In order to obtain reliable results from the spelling checker, all of the known rules and their exceptions have been implemented.

The spelling checker sometimes reports correct words as incorrect. One reason of this is the absence of some words in our dictionary. Although the dictionary is reasonably complete, there still remains many technical terms and proper names which are not included. Adding more and more words will obviously increase the functional performance of the checker. Another reason is that, most of the derivational suffixes are not included into the rules. If a stem that is derived by such a suffix is not present in the dictionary, it is reported as misspelled. Additionally, for the derivational suffixes that are included in our rules, the list of the roots that they can be affixed to may not be fully determined. This problem can also be solved by examining the dictionary. As far as execution performance goes, our implementation is very satisfactory giving an almost 1000 words/second word analysis throughput [19].

References

[1] Adalı, O., "Türkiye Türkçesinde biçimbirimler", TDK, Ankara, 1979.

[2] Banguoğlu, T., "Türkçenin grameri", TDK, Ankara, 1986.

[3] Brodda, B., Karlsson, F., "An experiment with morphological analysis of Finnish", Papers from the Institude of Linguistics, University of Stockholm, Publication 40, Stockholm, 1980.

[4] Can, K., "Yabancılar için Türkçe-İngilizce açıklamalı Türkçe dersleri", METU, Ankara, 1987.

[5] Demircan, Ö., "Türkiye Türkçesinde kök-ek bileşmeleri", TDK, Ankara, 1977.

[6] Hankamer, J., "Turkish generative morphology and morphological parsing", a paper presented at Second International Conference on Turkish Linguistics, İstanbul, 1984.

[7] Hankamer, J., "Morphological parsing and the lexicon", edited by William Marslen-Wilson, MIT Press.

[8] Hatiboglu, V., "Türkçenin ekleri", TDK, Ankara, 1981.

[9] Kasper, R., Weber, D., "User's reference manual for the C's Quechua adaptation program", Occasional Publications in Academic Computing, Number 8, Summer Institude of Linguistic, Inc., 1982.

[10] Kasper, R., Weber, D., "Programmer's reference manual for the C's Quechua adaptation program", Occasional Publications in Academic Computing, Number 9, Summer Institude of Linguistic, Inc., 1982.

[11] Koskenniemi, K., "Two-level morphology", University of Helsinki, Department of General Linguistics, Publication No. 11, Helsinki, Finland, 1983.

[12] Köksal, A., "Automatic morphological analysis of Turkish", Ph.D. Thesis, Hacettepe University, Ankara, 1975.

[13] Mason, T., Brown, D., "lex & yacc", edited by Dale Dougherty, O'Reilly & Associates, Inc., USA, May 1990.

[14] Özel, S., "Türkiye Türkçesinde sözcük türetme ve bileştirme", TDK, Ankara, 1977.

[15] Packard, D., "Computer-assisted morphological analysis of Ancient Greek", Computational and Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics, Pisa Leo S. Olschki, Firenze, 343 – 355, 1973.

[16] Sagay, Z., "Sözcük çekimi", Bilişim'78, Ankara, 1978.

[17] Sagvall, A., "A system for automatic inflectional analysis implemented for Russian, Data Linguistica 8, Almquist and Wiksell, Stockholm, 1973.

[18] Solak, A., Oflazer, K., "A finite state machine for Turkish syllable structure analysis", In Proceedings of the Fifth International Symposium on Computer and Information Sciences, Nevşehir, Türkiye, 1990.

[19] Solak, A., Oflazer, K., "Implementation details and performance results of a spelling checker for Turkish", in Proceedings of the Sixth International Symposium on Computer and Information Sciences, Side, Antalya, Türkiye, Oct. 1991.

[20] Solak, A., "Design and implementation of a spelling checker for Turkish", M.S. Thesis, Bilkent University, Ankara, 1991.

[21] Underhill, R., "Turkish", Studies in Turkish Linguistics, edited by Dan Isaac Slobin and Karl Zimmer, 7 – 21, 1986.

[22] "Türkçe sözlük", TDK, Ankara, 1988.

[23] "Yeni yazım kılavuzu", Eleventh Edition, TDK, Ankara, 1981.