# Probabilistic Turkish Word Root Generation

**A. Cüneyd TANTUĞ, Gülşen ERYİĞİT**

**Department of Computer Engineering, Istanbul Technical University**

**Istanbul, Turkey.**

cuneyd@cs.itu.edu.tr, gulsen@cs.itu.edu.tr

## *ABSTRACT*

In this paper, a new methodology that facilitates the generation of Turkish counterparts of newly emerging terms or concepts in foreign languages is introduced. The rules of Turkish word letter sequences are investigated. In the generation process, these rules and the bi-gram letter probabilities of Turkish words are used in cooperation. Only the generation of word roots is considered, the generation with the derivational suffixes is out of scope of this paper. The results are evaluated by making statistical comparisons between the generated words and a word lexicon and by carrying out a survey measuring the likelihoods of the generated words. As a result of this evaluation, it is seen that generation of word roots in a probabilistic and rule-based manner gives promising results and the newly generated roots are likely to be used.

**Keywords:** Turkish, root generation, bi-grams, letter sequences

## I. Introduction

Languages are living things that continuously develop and change together with the development of the society in different fields such as technology and culture. The penetration of the foreign words into the languages is a great problem that all the nations try to avoid. This is because it causes the deformation of the language structure and the incoherence between the intelligentsia and public. As a preventive action, a number of countries like Germany, France and Hungary established language customs that generate and propose new equivalents for the penetrating foreign words [1].

In the languages like English and French, the process of finding a new word to represent a new concept is limited. In order to find counterparts of a foreign word, the Latin roots are scanned or the abbreviations are used or the subparts of different words are combined to find a new word. On the other hand, the richness of the Turkish derivational suffixes yields the productivity of this language.

Turkish has a special place within the natural languages due to its rule-based structure. This feature facilitates the processing of a language by computers and makes it appropriate for natural language processing tasks.

In this work, it is aimed to develop a systematic way of word formation. The characteristics of Turkish, the rules and probabilities of word letter sequences are investigated. A software is designed to generate Turkish word roots by concatenating the letters using both the phonetic rules and probabilities. The resulting set $\mathfrak{R}$ consists of words that all obey Turkish rules and is composed of two subsets; $\boldsymbol{U}$ and $\boldsymbol{P}$.

$$\mathfrak{R} = \boldsymbol{U} \cup \boldsymbol{P} \tag{1}$$

The subset $\boldsymbol{U}$ stands for "Used Words" and implies the known Turkish words that are already in use. The other subset $\boldsymbol{P}$ is relatively large and the words in subset $\boldsymbol{P}$ are not in use. This means they have no current meaning, but they are subject to represent new concepts. This subset is denoted by $\boldsymbol{P}$ which indicates "Potential Words". The size of subset $\boldsymbol{P}$ is really important because it gives the potential of the language in some sense.

The main aim of our work concentrates on helping linguists to find new Turkish words corresponding to foreign terms. In order to achieve our motivation, we have generated words in two ways; rule based and probabilistic rule based methods. In the former, the rules of letter arrangements in Turkish words are investigated and a software is developed to generate word roots by using this rules. In the latter one we have added the probability of the next letter given the preceding letter (letter 2-grams of Turkish).

This paper describes the process in 4 sections. Second section gives the details of the rule based word generation while the following section expresses the probabilistic rule based word generation. In the fourth section, the evaluation of our work has been done. Conclusion and future work are discussed in the last section.

## II. Rule Based Word Generation

To facilitate building a rule based word generation system, one should investigate the letter arrangement rules of Turkish words. The rules of letter

arrangements of Turkish words can be found in the literature [1, 2, 3, 4, 5] as follows:

- Palatal harmony
- Labial harmony
- Consonant harmony
- A consonant can not be followed by the same consonant
- A vowel can not be followed by a vowel
- Three consecutive consonants can not occur
- The letters 'o' and 'ö' can be found in the first syllable only
- Some consonants can not be the first letter of the word
- Some letters can not be last letter of the word
- Two consonants can not be in the beginning of the word
- There are some patterns of consonants pairs that can be occur in the end of the word
- The letter "n" can not be followed by "b"

We have developed a software which employs these rules as finite state automata. A candidate word root is formed by concatenating all possible letters; at each step, a list of possible successor letters is populated, one element of this list is selected and concatenated with the head of the candidate word and this process repeats recursively.

In addition to these rules, the developed software has some other properties. One of them is the limit of the generated word length, i.e. the user can choose the maximum length of the words that will be generated. Moreover, the ability of selecting a letter sequence as the head of the candidate word is involved in the software. This allows users to generate words that begin with the desired sequence of letters.

The rule based word generation algorithm given below generates all the possible combinations with a maximum length.

```
Procedure RBGenerate(word){
   If maximum length reached then stop
   Populate a list of possible successor letter
   Repeat
      Choose one letter from the list in order
      Add the selected letter to the end of the word
      Store the word
      Call RBGenerate(word)
   Until (the list is empty)
}
```

## III. Probabilistic Rule Based Word Generation

The rule based word generation system produces all possible combinations of the letters that obeys Turkish letter arrangement rules. It is clear that, the size of the result set $\mathfrak{R}$ will be huge in some cases, even though the user determines a beginning sequence of letter. We thought of selecting the successor letters in a more elegant way; picking up the most likely letter from the possible list of letters. At this point, the N-Gram technique is used as a measure of likeness to perform the selection of the most likely letter. This model approximates the probability of a letter given all previous letters [6]. For the sake of simplicity, the 2-gram model is implemented in our work. The needed Turkish letter 2-gram probabilities are taken from the Dalkılıç's work [7]. In his work, Dalkılıç gives the probabilities of occurrence of all letter couples (ie: "ae", "al", "ak") that are obtained from various Turkish corpuses.

The selection of the successor letter is based on a proportionate (roulette wheel) selection schema in which all letters have a selection probability linearly proportional to their occurrence probability with the last letter in the word.

The rule based system generates a list of all possible letters that can be added to the current word root. If $W$ is a candidate word root and $n$ is the current length of this word at an arbitrary step, the next successor letter that will be concatenated to $W$ is selected from the list of possible letters $L$:

$$W_{n+1} = x_i \quad \text{or} \quad W = W + x_i \ , \quad x_i \in L \qquad (2)$$

This selection is done by giving more chance to the more likely letters that should follow the last letter of $W$ that is $W_n$. Each letter $x_i$ in $L$ has a chance of selection directly proportional to the probability $P(x_i|W_n)$ :

$$P_s(x_i) = \frac{P(x_i \mid W_n)}{\sum_{j=1}^{k} P(x_j \mid W_n)} \qquad (3)$$

where $k$ is the cardinality of the set $L$ and $P_s(x_i)$ is the selection probability o letter $x_i$ from $L$.

At each step of recursive generation process, a proportionate selection is carried out to find the letter that will be added to the word root. This recursive structure returns when the maximum length $n_{max}$ is reached or the size of possible letters set is zero ($card(L) = 0$).

Because of the probabilistic nature of this generation method, same word roots can be generated. This is avoided by not storing the repetitive words. Also the user should give a maximum number of word roots that will be generated. Unlike the former method, this probabilistic rule based method do not generate all possible combinations of the letters, it simply make selections among an appropriate list of letters. The

algorithm of the probabilistic rule based word generation method is given below. This algorithm generates just one word with a given length.

```
Procedure RBPGenerate(word){
 For 2 to Maxlength
     Populate a list of possible successor letters
     Randomly choose a letter according to the
     probabilities
     Add the selected letter to the end of the word
     If not stored before then Store the word
 End for
}
```

## IV. Results

The results of our work can be divided into 2 parts. First part gives the results of the rule based classification while the second part shows the evaluation of the probabilistic rule based word generation.

### 1. Rule Based Generation Results

Like all other agglutinative languages, Turkish has an important and functional feature: "derivation". Derivational suffixes change most of the time the part of speech or the meaning of the word to which they are affixed. For example in Turkish "göz" means "eye", "gözlük" means "eye-glasses", "gözlükçü" means "the person who sells eye-glasses" and etc. This can be view as a way of generating new words. But in this paper, we aimed to produce new word roots rather than concatenating suffixes. Since the average length of the Turkish words is stated in [8] as 6, we limited our generation with a length of 7. The number of the generated words with respect to length criteria is given in Table 1.

| Length | The Number of Generated Word Root |
|---|---|
| 2 | 240 |
| 3 | 1,416 |
| 4 | 12,512 |
| 5 | 146,416 |
| 6 | 1,058,512 |
| 7 | 10,618,784 |
| **TOTAL** | **11,837,848** |

Table 1 : The number of generated words

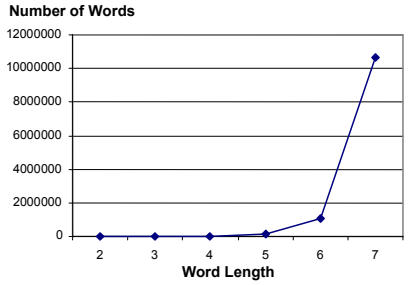


Figure 1: The distribution graph of the generated words

In the evaluation, it must be observed that how many of words currently in use can be generated and how many of these words obey Turkish rules. To achieve these aims, there must be a dictionary of words that are currently in use. Hence, we build a large dictionary by combining words from various sources [9,10]. The word distribution of the dictionary is given below:

| Length | # of Words |
|---|---|
| 2 | 119 |
| 3 | 920 |
| 4 | 2331 |
| 5 | 6021 |
| 6 | 6080 |
| 7 | 7218 |
| 8 | 7286 |
| 9 | 5270 |
| 10 | 4614 |
| 11 | 2979 |

| Length | # of Words |
|---|---|
| 12 | 1419 |
| 13 | 851 |
| 14 | 512 |
| 15 | 231 |
| 16 | 129 |
| 17 | 44 |
| 18 | 24 |
| 19 | 9 |
| 20 | 2 |
| **TOTAL** | **22689** |

Table 2: The distribution of the words in the dictionary

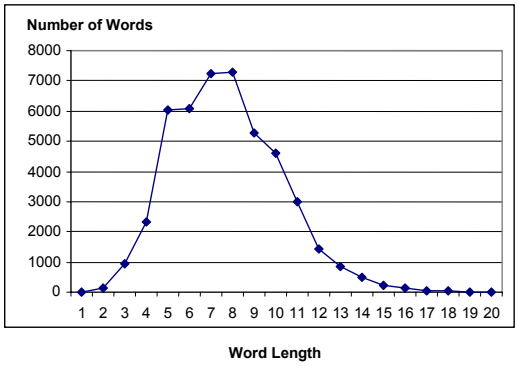


Figure 2: The distribution of the words in the dictionary with respect to the lengths

The result set of rule based word generator ($\mathcal{R}$) must be compared with the dictionary in many ways.

First, we check for the existence of the generated words in the dictionary. The occurrence column in Table 3 represents how many dictionary words found in resulting set $\mathcal{R}$. The ratio column shows the ratio of the number of generated words existing in the dictionary to the number of words in the dictionary. These results are calculated for each word length separately.

| Length | Occurrence | Ratio |
|---|---|---|
| 2 | 89 | %74.8 |
| 3 | 474 | %51.5 |
| 4 | 1080 | %46.3 |
| 5 | 2537 | %42.1 |
| 6 | 2081 | %34,23 |
| 7 | 2576 | %35.7 |
| | **TOTAL: 8837** | **AVERAGE : %38,95** |

Table 3 – Existence of the generated words

As seen in the Table 3, the existence ratio of the generated words in the dictionary decreases as the length increases. Our software can not generate all of the words in the dictionary, the main reason for that is the considerable number of foreign words that do not obey Turkish rules.

These results give the size of the used words subset ***U***. Also it is the main purpose of this work that finding new word roots for new concepts so we should investigate the size of the other subset ***P*** called potential words subset. The Table 4 shows the usage ratios of the generated words.

| Length | Usage Ratio | Unused Ratio |
|---|---|---|
| 2 | %42,79 | %57,21 |
| 3 | %33,47 | %76,53 |
| 4 | %8,63 | %91,37 |
| 5 | %1,73 | %98,27 |
| 6 | %0,20 | %99,80 |
| 7 | %0,02 | %99,98 |

Table 4: The usage ratio of the generated word

The unused ratio gives us the flexibility of Turkish in some manner. The higher unused ratio means the more unassigned potential words. So linguists can choose a potential word to represent a new concept by simply assigning a new meaning to an unused word. Consequently, it is better for a language to have a large potential word set ***P***. These results prove that Turkish has a large set of ***P*** that makes Turkish more productive than most of the other languages.

## 2. Probabilistic Rule Based Generation Results

It is relatively hard to evaluate the results of the probabilistic rule based generation software. Because of the generated word set $\mathfrak{R}$ is probabilistic, the elements of this set can not be evaluated by an automatic process. So it should be shown that a word which is composed of letter pairs with higher probabilities is more human acceptable than the one with a lower probability. The former one is generated by selecting the most likely sequence of letters to appear, this means the resulting word is very similar to Turkish words in use whereas the latter one has very low similarity with Turkish words and probably sounds jarring to people.

This likelihood of a generated word is defined by the formula:

$$L(W) = \frac{1}{l}\sum_{i=1}^{l-1} f(W_i)P(W_i \mid W_{i+1}) \quad (4)$$

where $W$ is the word, $W_i$ is i.th letter of the word $W$, $f(W_i)$ is the occurrence frequency of the letter $W_i$, and $l$ is the length of the word $W$.

We prepared a survey in which there are words that have both high and low likelihood measures, and 30 person are directed to give marks ranging from 1 (the most jarring) to 5 (the most similar) to these words. It is expected that, people accepts the words with high likelihood measures more and classifies words with low likelihood measures as jarring or useless.

The average mark for each word is calculated by processing the survey answers and compared with the likelihood measure of the word computed by the formula (4). The resulting table is given in Table 5.

| Word | Survey Result | Likelihood Measure |
|---|---|---|
| alark | 3,09 | 5,02 |
| alcıp | 1,96 | 0,95 |
| alşaç | 2,13 | 1,10 |
| aldak | 3,87 | 2,49 |
| alısan | 3,35 | 2,91 |
| kastala | 3,65 | 3,78 |
| kasıman | 3,00 | 3,55 |
| kasak | 3,48 | 3,08 |
| kaspav | 2,70 | 1,73 |
| kashırk | 1,39 | 1,62 |
| bilerk | 2,83 | 3,76 |
| bilmer | 4,00 | 3,28 |
| bilineç | 2,91 | 2,57 |
| biliz | 3,30 | 2,30 |
| bilcef | 2,35 | 1,31 |
| kalmar | 4,30 | 4,58 |
| kalatır | 3,43 | 3,76 |
| kalışmay | 2,65 | 2,87 |
| kalcıç | 1,13 | 1,85 |
| kalğav | 1,17 | 1,79 |
| kapyız | 2,17 | 1,35 |
| kapılk | 2,22 | 1,50 |
| kapranaz | 3,70 | 2,63 |
| kapıkla | 3,13 | 2,92 |
| kapala | 4,48 | 3,67 |
| içleter | 2,83 | 3,17 |
| içinele | 3,09 | 3,12 |
| içpez | 2,52 | 0,39 |
| içkil | 4,00 | 1,64 |
| içigne | 1,65 | 0,82 |

Table 5 – Survey Results

The calculated likelihood measure is scaled by a factor in order to set the range of this measure between 0 and 5 as the same range that the survey results have.

The graphical representation of the survey results can be seen in Figure 3.

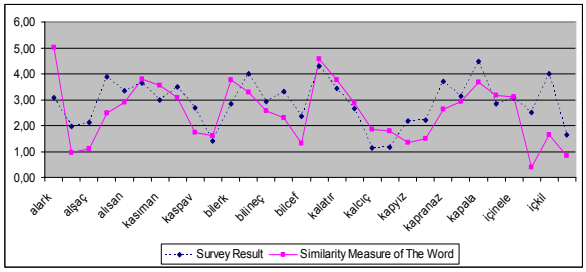


Figure 3 – The graph of the average survey marks and scaled similarity measure for each measure

Before making the survey, the expectation was that the higher similarity measure of a word will get higher ranks from people. Figure 3 illustrates that the

two norms of the results are very close to each other. This means the expectation is proved. The interpretation of that in a more informal way is "people choose the words that have higher similarity measures as more usable or familiar".

## V. Conclusion And Future Work

The main purpose of this work was developing a computer software that can generate Turkish words root that obey the whole Turkish letter sequence rules. This will have a great effect in finding new Turkish counterparts of concepts or terms that are newly arising in foreign languages. By this way, the impact of the deformation of Turkish by non-Turkish words can be eliminated.

The generation of Turkish words is implemented by using two approaches; in one approach a rule based system that combine letters by using Turkish rules generates all possible Turkish words. The generated words are then compared with Turkish words in use. A dictionary is built to achieve this. It is shown that most of the words in the dictionary are in "the generated words set". But the major consequence of this approach is the large number of unassigned (means not in use), legal (means obeys all Turkish rules) possible Turkish word roots. In some manner, this large number bring outs the ability or flexibility of Turkish in finding new words representing new concepts.

A probabilistic rule based word root generation is implemented as the second approach. This can be view as a combination of both rule based system and the probabilities of letter pairs in Turkish words. This method populates a list of letters that obeys the rules and uses a selection mechanism that randomly picks a letter from this list considering the 2-gram probabilities of letters. By this way, this implementation generates more similar word roots to the words that are already in use. Also, by a survey, it is proved that people really think the words with higher likelihood measure, more usable or acceptable.

This study can be extended, especially the probabilistic rule based generation method, by selecting letters more appropriately, for example by using 2-grams, 3-grams and 4-grams incorporated. This will yield a better approximation of generated words to the words that are already in use.

**References**


[1] Hengirmen, M., "Türkçe Dilbilgisi", *Engin Yayıncılık*, Ankara, 2002.

[2] Banguoğlu, T., "Türkçenin Grameri", *Türk Dil Kurumu Yayınları : 528*, Ankara, 2000.

[3] Keçeci, H. F., "Bir Robot Koluna Kumanda Eden Doğal Dil Anlama Sistemi", *İstanbul Teknik Üniversitesi Yüksek Lisans Tezi*, İstanbul, 1996.

[4] Cebiroğlu, G. Adalı E., "Sözlüksüz Köke Ulaşma Yöntemi", *TBD 19. Bilişim Kurultayı*, İstanbul, 2002.

[5] Cebiroglu G., Tantug C., Adali E., Erenler Y., 2003, "Sentetik Türkçe Sözcük Kökleri Üretimi", Tainn 2003 Çanakkale, 2003

[6] Jurafsky, D., Martin, J.H., "Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", *Prentice Hall,* 2000.

[7] Dalkıllıç M.E., Dalkılıç G., "On The Cryptographic Patterns and Frequencies in Turkish Language", *ADVIS 2002, LNCS 2457*, 2002

[8] Dalkıllıç M.E., Dalkılıç G., "Basılı Türkçe'nin Önemli Bazı İstatistiksel Özellikleri", *İstatistik Araştırma Dergisi*, 2002

[9] Oflazer, K., "PC-KIMMO Rule Specification For Turkish Morphology", *Bilkent University*, Ankara, 1994.

[10] Türk Dil Kurumu, "İmla Kılavuzu", Türk Dil Kurumu Yayınları, Ankara, 2000.