# Morphology and lexicon-based machine translation of Ottoman Turkish to Modern Turkish*

Joomy Korkut
Princeton University
joomy@cs.princeton.edu

## Abstract

We present a rule-based system to translate Ottoman Turkish to Modern Turkish. This system depends on partial morphological parsing of Ottoman words with the help of dictionaries to verify the roots, and it generates the Modern Turkish translation from the translated root by conjugating and declining the parsed morphology into its modern spelling and pronunciation.

We use two dictionaries, one for loanwords from Arabic and Persian, and one for Modern Turkish words. Searching in the loanword dictionary is straightforward, but for the modern dictionary we derive a pattern from the old spelling and search using that pattern.

## 1 Introduction

Ottoman Turkish is a variant of the Turkish language that was used in the Ottoman Empire. This term usually refers to the prestige dialect used by the educated upper class. This dialect heavily used loanwords from Arabic and Persian, going as far as borrowing grammatical structures [7, 12, 16]. Today Ottoman Turkish can refer to the script used before 1928 to write the aforementioned prestige dialect and also the simple Turkish that was used by others. Both were written in Perso-Arabic script, with a few adjustments to fit the Turkish language. However, despite being used as the default script from 13$^{th}$ to 20$^{th}$ century, there were no standard spelling rules for Ottoman Turkish.

In this project, we will prioritize the spelling that was common in the late 19$^{th}$ and early 20$^{th}$ century.

When we talk about machine translation from Ottoman Turkish to Modern Turkish, we mean a transcription system that follows the modern pronunciation of words instead of the historical one. For example, consider the Ottoman word ايدوب (Eng.: "having done"). Following the old pronunciation, this word would be transcribed "*idüb*", while in Modern Turkish it should be "*edip*".

In this paper, we present a rule-based algorithm to achieve translation from Ottoman Turkish to Modern Turkish. We will show that partial morphological parsing of Ottoman Turkish is easier because of its syntax rules that do not reflect pronunciation. Ottoman Turkish suffixes are often not conjugated or declined, which makes it easier for a computer to parse. Once we strip a word of its suffixes, we search for the root in our dictionaries, one for loanwords from Arabic and Persian, and one for Modern Turkish words. The loanword dictionary is a mapping from the old script to the new script, while the modern dictionary is only a set of words. In order to search a word the modern dictionary, we generate a pattern from the old spelling and see what words fit that pattern. The pattern is based on what sounds that an Ottoman letter can correspond to in that context, and where it can take extra vowels.

### 1.1 Motivation

Ottoman Turkish is an extinct language, but all college-level Turkish literature and history students have to study it in Turkish universities. Transcription and transliteration from Ottoman Turkish to Modern Turkish is still a task that Turkish studies scholars and historians have to do, or hire other people to do. It is an expensive task that takes a lot of labor-hours. Our system aims to solve this problem by automating it. We envision that our system will be used alongside optical character recognition (OCR) systems for Perso-Arabic script, which can take a scanned image of text and turn it into a digital format. Then our system can take the digital Ottoman text and translate into Modern Turkish.

Our system can also be a learning tool for people studying Ottoman Turkish. Especially for non-native speakers of Turkish, the Ottoman dialect and script is a tough nut to crack. Students have to interact with a teacher who confirms their transcriptions, but teachers are not available to everyone all the time. Their other option is to use an already existing transcription, which also might not be available. Having a system that can verify the work of learners will be quite helpful in that sense.

While scholars and historians should still learn Ottoman

*Version compiled on May 14, 2019 at 9:34am.

Turkish, there will also be a group of users who do not know Ottoman Turkish, but still want to understand a given text. If our system is developed enough, it will make primary sources available to more researchers.[1]

We preferred a rule based approach due to the lack of data in Ottoman Turkish. The amount of digitized texts alongside their translation or transcription is very limited in the first place. Resources in Ottoman Turkish are often either in image form, or untranslated.[2] Furthermore, we would need enough data to account for all the grammatical irregularities. Unfortunately, this is not possible in the current state of Ottoman text digitization.

# 2 Challenges

Before we present our system, it is necessary to explain the challenges of reading and translating Ottoman Turkish.

## 2.1 Orthographical ambiguity

Ottoman Turkish used Perso-Arabic script with a few extra letters and diacritics, we will call this the Ottoman script for brevity. Modern Turkish script, which is the Latin alphabet with extra letters, does not correspond to the Ottoman script in a straightforward way.

There are letters in Ottoman script that correspond to many different letters in the new script. For example, و corresponds to *v*, *o*, *ö*, *u* or *ü* depending on the context. Another example is ك, which can correspond to *k*, *g*, *ğ*, *y* or *n* depending on the context. These are the most extreme cases, and while there are other such examples, we will not go through all of them here. Given that our end goal is to generate Modern Turkish text, this is the kind of ambiguity we should worry about.

Similarly, some letters in the new script can be written with multiple letters in the Ottoman script. For example, the letter *s* should be written with س, ص or ث, depending on the word. Since we are not generating any Ottoman Turkish text, we are not concerned about this kind of ambiguity.

## 2.2 Missing vowels

Many of the vowels in words are omitted in writing and inferred by the reader. For example, the Ottoman word ترلك (Eng.: "slipper") should be translated to "*terlik*". However, the Ottoman spelling consists of only four letters, which are the consonants *t*, *r*, *l* and *k*. The vowels *e* and *i* are inferred by the reader, with help from their handle on the vocabulary and their experience with the language.

[1] Given that they are available in digital format. OCR systems only work well on printed (*matbu*) material; other writing styles and handwriting require a different kind of expertise.

[2] The OpenOttoman project (https://openottoman.org/) and Ottoman WikiSource (https://wikisource.org/wiki/Osmanlıca_VikiKaynak) are good examples of both.

## 2.3 Legacy spellings of loanwords

Ottoman Turkish preserves the original spellings of words borrowed from Arabic and Persian, regardless of how those words are pronounced in Turkish.

Consider the Ottoman loanword from Arabic جدا (Eng: "seriously"). In modern Turkish this would be written "*cidden*". However, notice that the Ottoman spelling consists of three letters: ج, which corresponds to *c* here with no issues, then د, which inexplicably is doubled in pronunciation, and then ا, which would normally correspond to *a* at the end of the word, but instead is pronounced *en*. For someone who cannot recognize this word, it is easy to miss the translation above and go for the Persian loanword "*cüda*" (Eng: "separated"), which is spelled the same way.

The reason for the mess here is that diacritics are almost always entirely omitted in Ottoman Turkish. If we were to write the same word with all the diacritics, we would write جِدًّا, which would disambiguate the translation.

For an example of a Persian loanword, consider پاپوش (Eng: "shoe"). A direct transcription would be "*papuş*", but in Modern Turkish this would be written "*pabuç*".

Later eras of Ottoman Turkish spell the same word as پابوج, which directly transcribes to "*pabuc*". Notice that the *c* at the end becomes *ç* in Modern Turkish. Similarly *d*'s at the end of words often change to *t*, and *b*'s at the end change to *p*.[3] This is a common change in Turkish grammar called "fortitive assimilation"; loanwords in Modern Turkish reflect this in the spelling as well, but loanwords in Ottoman Turkish keep the original spellings.

## 2.4 Ambiguous word boundaries

The Ottoman script is written from right to left, by attaching letters to each other. However, not every letter attach to one another. The letters ا, د, ذ, ر, ز, ژ and و only attach to the letter before (on the right), and not the letter after (on the left). This is already handled perfectly by digital systems. When we are inspecting a list of characters, we do not have to worry about what letters attach to another, we can simply inspect them one at a time.

However, in Ottoman Turkish specifically, the letter ه has a conditional attachment rule. This letter can stand for *h*, *a* or *e* in pronunciation. It only attaches to the left if the letter ه is read *h* in that context, and does not attach to the left otherwise.

Consider the loanword from Arabic الهه (Eng: "goddess"), "*ilahe*" in Modern Turkish. The last two letters of this word are both ه. The former is in its middle form ـهـ, since it stands

[3] Such last letters can change back if if the word takes a suffix that starts with a vowel. The word *pabuç* with the dative suffix *-a* would be "*pabuca*". This is an example of "consonant lenition".

for *h*. It attaches to the latter, is in its final form ـه since it stands for *e*. If we were to add a suffix to this word, such as یه, which would be read *-(y)e*, for the dative case, we would see that it does not attach to the ه. The final word would then be الهەیه (Eng: "to the goddess"), "*ilaheye*" in Modern Turkish.

In order to separate that suffix from the word so that it doesn't attach, we had to use a Unicode character called "zero width non-joiner". This character is invisible, but it prevents the characters before and after it from attaching. It is also common to use a space, which is visible, to achieve the same effect. This is a problem for our system, because Ottoman Turkish spelling normally has spaces as word boundaries. However, this usage means not every space is a word boundary, which means we have to take this into account when we translate a word. What we think is a word might not actually be the full word.

# 3 Related work

## 3.1 Morphological parsing

Parsing Turkish morphology has long been an area of interest in computational linguistics, since the agglutinative structure of the Turkish language is an challenging research question. Therefore there is a long line of work in this area [4, 5, 8–11, 13–15], among which it is possible to find both finite-state-based approaches and statistical approaches. Grammar of the Turkish language is unusually regular, which might have encouraged researchers to insist on finite-state methods.

Our approach in this paper is inspired from Eryiğit and Adalı [5], which describes a finite-state machine to parse words in reverse, by stripping affixes and reaching a root. Their approach does not use a lexicon, which is different from our approach.

## 3.2 Machine translation of Ottoman Turkish

Our project is not the first to attempt machine translation from Ottoman Turkish to Modern Turkish. We have found at least two such projects.

One of them was developed by Rahmi Tura, whose web site is not reachable anymore. He had a demo video online, which showcased features like translation from Ottoman Turkish to Modern Turkish with or without the diacritics and vice versa. However, the program was never made available or sold online. In our personal communication[4], he stated that his system has a database that consists of approximately one million words, in their conjugated and declined forms. He argued that matching words in their whole form increases the accuracy, even though it is at the expense of developing a comprehensive database.

[4]On April 29th, 2019, via e-mail.

The other such project is called Dervaze, developed by Güngör and Şahin [6]. Their project is very similar to ours, and they have a working demo system online.[5] We have tested their implementation, and we encountered some problems that we claim stem for their pipeline design:

- They choose to pick only one translation out of many possible ones. Our project will give the full range of possibilities to the user, which reduces readability of the result but increases accuracy.
- They skip words that fail to translate. We will always report to the user if we fail on a word.
- They fail on translating simple Turkish words. This is because they use a Ottoman to Turkish dictionary which might not include simpler Turkish words.

# 4 Method and algorithms

## 4.1 Overview

Despite the challenges mentioned above, our algorithm is based on a simple principle: Turkish is an agglutinative language where roots of words are not inflected when new suffixes are attached, and the new suffixes take forms according to the last syllable of the word that are attached to. We will take a word and perform a partial morphological parsing by break it apart to its roots and suffixes, then we will find the translation of the root and reconstruct the suffixes according to the grammar rules of Modern Turkish.

Take the plural suffix for nouns, written as *-ler* or *-lar* in Modern Turkish. If we attach this suffix to the word "*göz*" (Eng: "eye"), then we get "*gözler*", because the last syllable[6] before the suffix contains one of the front vowels *e*, *i*, *ö*, *ü*. But if we attach the same suffix to the word "*kuş*" (Eng: "bird"), then we get "*kuşlar*", because the previous syllable has one of the back vowels *a*, *ı*, *ö*, *ü*.

The plural suffix in Ottoman Turkish, however, does not change its form. It is written as لر- for all cases, only containing the letters for the *l* and *r*. The word "*gözler*" would be written as گوزلر and "*kuşlar*" would be written as قوشلر. The missing vowels problem we described in subsection 2.2 work in our advantage here; the computer can just look for the لر- at the end, strip it off, and use the rest as an hypothetical root.

An hypothetical root would be looked up in the dictionaries, and if there is not entry for that, we would continue to look for more suffixes. Even if there is an entry, it is possible that a word has multiple translations in Modern Turkish, so we should still look for more suffixes.

There are a few corner cases in the لر- (*-ler*, *-lar*) example. One of them is that the ending with this letter sequence

[5]http://dervaze.com/translate-ott/

[6]Syllabification rules of Turkish dictate that there is only one vowel per syllable.

does not necessarily mean it is a plural suffix. It is possible to have words that end with the same letters that have no suffixes. Consider the loanword from French پوپولر (Eng: "popular"), which translates to Modern Turkish "*popüler*". We can try to parse the ending as a suffix and search for the beginning in the dictionary, which will fail. However, our program can find a word in one our dictionaries that matches either directly or through a pattern, which lets the program recover from the previous state.

The other corner case is that لر- (*-ler*, *-lar*) is not just the plural suffix for nouns. The 3rd person plural suffix in verb conjugations is also of the same form. For example, the word گلدی (Eng: "he/she/it came", Tur: "*geldi*") can take the same suffix and become گلدیلر (Eng: "they came", Tur: "*geldiler*"). Here the fact that our morphological parsing is partial saves us. When we are parsing the suffixes our goal is not to produce a full morphological structure like *göz<n><pl>* for "*gözler*" or *gel<v><past><3p>* for "*geldiler*". Our goal is merely to identify what suffixes we have so that we can reconstruct them, we are not interested in what function they have in the grammar.

Most suffixes in Ottoman Turkish generally have only one form, even though their Modern Turkish equivalents might have many more.[7] For example, the suffix of the direct past tense دی- in Ottoman Turkish has 8 different forms in Modern Turkish: *-dı*, *-di*, *-du*, *-dü*, *-tı*, *-ti*, *-tu* and *-tü*. Notwithstanding, there are exceptions to these cases. Rarely it is possible to see دی- as دو-, تی- or تو-, which correspond to the forms in Modern Turkish. This is due to the lack of standardization in Ottoman Turkish; a writer's spelling depends on the era they live in and also their "idiography", i.e. an individual's unique way of spelling.

### 4.2 Dictionary lookup and pattern generation

As we described earlier, our program will use two kinds of dictionaries to look up words. The first one is a dictionary that maps Ottoman Turkish spellings to Modern Turkish spellings[8], it is basically a table with two columns. It will have a mapping from الهه to "*ilâhe*", from جدا to "*cidden*" and so on. The second dictionary is a table with only one column, it is simply a list of words in Modern Turkish borrowed from the Zemberek project [1].

The first dictionary mostly consists of loanwords from Arabic and Persian, since they are spelled in Ottoman Turkish the same way they are spelled in the original language, even when their pronunciation is changed in Turkish. Therefore, unless there is a dictionary the program can look up from, it will get those word wrong. Consider the word چهارشنبه (Eng: "wednesday"). It is spelled "*çarşamba*" in Modern Turkish, but a direct transcription would be "*çeharşenbe*".

For word that are originally Turkish, there is a much higher chance that we can get the right translation from the spelling, despite the ambiguities. We will generate a regular expression from the Ottoman spelling and match it against the entries in our second dictionary.

Consider the word گوز (Eng: "eye", Tur: "*göz*") that we mentioned above. There are three letters in this word:

1. گ only corresponds to one letter in Modern Turkish, which is *g*. Hence our pattern for this letter only consists of `/g/`.
2. و can correspond to five letters in Modern Turkish, *v*, *o*, *ö*, *u* or *ü*. The last four of these are vowels, and if و here stands for a vowel, then it does not imply the possible existence of another vowel before or after it. However, if و stands for *v* here, it may or may not be followed or preceded by a vowel, and we have to account for that vowel in our pattern.
   Hence our pattern for this letter would be `/(((a|e|i|ı|o|ö|u|ü)?v(a|e|i|ı|o|ö|u|ü)?)|o|ö|u|ü)/`. We have optional vowel patterns before and after the *v*, or this letter might just be standing for *o*, *ö*, *u* or *ü*
3. ز only corresponds to one letter in Modern Turkish, which is *z*. Hence our pattern for this letter only consists of `/z/`.

Therefore, our final pattern for this word will be `/^g(((a|e|i|ı|o|ö|u|ü)?v(a|e|i|ı|o|ö|u|ü)?)|o|ö|u|ü)z$/`.

Searching with this pattern in the modern dictionary, we will find two matching words: "*göz*" and "*güz*". This ambiguity is inherent to the Ottoman script, and there is not much we can do about that. Our program will take this root and decline the suffixes if there are any, and display both translations to the end user.

## 5 Experiments and results

There is a prototype implementation[9] of our system in the Haskell programming language. This implementation is currently missing many of the suffixes in the Turkish language, but it works well on simple inputs. Here is an input sentence we entered to our program: "یاپراقلر باخچەلردن دوشمش." (Eng: "Leaves fell from the gardens.") For this input, we got the following output:

```
yapraklar
bahçelerden
düşmüş, duşmuş, dövüşmüş, döşmüş
.
(2.61 secs, 15,223,379,696 bytes)
```

[7] This is because the Modern Turkish spelling closely reflects the pronunciation.

[8] http://ekitapgunlugu.blogspot.com/2013/03/osmanlca-sozluk-veritaban.html. It is a combination of ten different dictionaries compiled by an unnamed blogger.

[9] https://github.com/joom/dilacar

For the first two words, our system succeeds in finding the right match. For the last word the correct translation is "*düşmüş*" (Eng: "fell"). "*duşmuş*" (Eng: "was the shower") and "*döşmüş*" (Eng: "was the bosom") are technically correct yet nonsensical translations. The correct spelling "*dövüşmüş*" has explicit vowels and consonants for the *ö*, *v* and *ü* in the middle. The aggressive possible vowel insertion in our pattern generation assumes that these words can be spelled with omitted vowels. This can create incorrect translations but also if we encounter an exceptional spelling our system will capture it.

# 6 Future work

As we see above, a three word sentence can take 2.61 seconds to translate, which is very high. Most of this time is spent on dictionary lookups. Our implementation currently goes through the entire dictionary linearly and tries to match the entries. The Ottoman Turkish to Modern Turkish dictionary can be made more efficient through different methods, such as using a trie as the data structure. However, this would require some setup time for our program to populate the data structure with the dictionary content. The Turkish dictionary is slightly different, since our system uses regular expressions to see which words match the pattern, yet the paper of Baeza-Yates and Gonnet [2] show that a special form of tries called suffix trees can help make that more efficient. A more established option for both dictionaries would be to compile the them into a finite-state transducer or machine, similar to the work of Çöltekin [4].

Most important enhancement for our system's overall accuracy would be better dictionaries. Current dictionaries we have are limited: the Ottoman dictionary currently has 14.000 entries, while the modern one has slightly less than 29.000. Compared to the one million words of Tura's project, even if his dictionary contains conjugations and declensions as well, our figures are low. Furthermore, the entries we have in our dictionaries are not in their ideal forms. In the Ottoman dictionary, translations of Arabic and Persian words use too many *â* and *î*, while the more modern spelling replaces them with *a* and *i* unless it creates ambiguity. In the modern dictionary, verbs are in the infinitive form instead of the root form, which currently causes problems. These problems can be solved if more time is spent on data cleaning, however we currently lack the time and manpower to do so.

# 7 Conclusion

In this paper, we have presented a rule-based system to translate Ottoman Turkish to Modern Turkish, along with a prototype implementation that demonstrates the feasibility of our algorithm.

The main feature of our system is partial morphological parsing from the Ottoman spelling of words, taking advantage of the often more stable spellings of suffixes in Ottoman Turkish. Our program tries to parse suffixes and to search in dictionaries with what is left.

This search involves two dictionaries, a more limited dictionary from Ottoman Turkish to Modern Turkish in which we search directly, and a more extensive Modern Turkish word list in which we search using a regular expression we generate from the Ottoman spelling of a presumptive root, using the orthographical rules in Ottoman Turkish.

Our implementation is a testament to our claim that a rule-based solution is the right approach to translation between Ottoman Turkish and Modern Turkish, since the two languages are basically the same except the spelling rules and Modern Turkish spelling has very few exceptions.

# Acknowledgments

We would like to thank Nilüfer Hatemi and Duygu Coşkuntuna from the Department of Near Eastern Studies at Princeton for their comments on the draft of this paper.

# References

[1] Ahmet Afşın Akın and Mehmet Dündar Akın. Zemberek, an open source NLP framework for Turkic languages. *Structure*, 10:1–5, 2007.

[2] Ricardo A. Baeza-Yates and Gaston H. Gonnet. Fast text searching for regular expressions or automaton searching on tries. *Journal of the ACM (JACM)*, 43(6): 915–936, 1996.

[3] Korkut Buğday. *The Routledge Introduction to Literary Ottoman*. Routledge, 2009.

[4] Çağrı Çöltekin. A freely available morphological analyzer for Turkish. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 820–827, 2010.

[5] Gülşen Eryiğit and Eşref Adalı. An affix stripping morphological analyzer for Turkish. In *Proceedings of the IASTED international conference artificial intelligence and applications*, 2004.

[6] Ayşegül Güngör and İ. Emre Şahin. Dervaze: A spelling dictionary for digital translation of Ottoman documents. *International Journal of Languages' Education and Teaching*, 5(3):78–84, 2017. doi: 10.18298/ijlet.1824.

[7] V. Hovhannes Hagopian. *Ottoman-Turkish conversation-grammar: a practical method of learning the Ottoman-Turkish language*. J. Groos, 1907.

[8] Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410, 2002.

[9] Jorge Hankamer. Finite state morphology and left to right phonology. In *Proceedings of the West Coast Conference on Formal Linguistics*, volume 5, pages 41–52, 1986.

[10] Kemal Oflazer. Two-level description of Turkish morphology. *Literary and linguistic computing*, 9(2): 137–148, 1994.

[11] Kemal Oflazer, Elvan Göçmen, and Cem Bozşahin. An outline of Turkish morphology. *Report to NATO Science Division SfS III (TU-LANGUAGE), Brussels*, 1994.

[12] James William Redhouse. *A simplified grammar of the Ottoman-Turkish language*, volume 9. Trübner & Company, 1884.

[13] Haşim Sak, Tunga Güngör, and Murat Saraçlar. Morphological disambiguation of Turkish text with perceptron algorithm. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 107–118. Springer, 2007.

[14] Haşim Sak, Tunga Güngör, and Murat Saraçlar. A stochastic finite-state morphological parser for turkish. In *Proceedings of the ACL-IJCNLP 2009 Conference short papers*, pages 273–276. Association for Computational Linguistics, 2009.

[15] Ayşın Solak and Kemal Oflazer. Parsing agglutinative word structures and its application to spelling checking for Turkish. In *Proceedings of the 14th conference on Computational Linguistics-Volume 1*, pages 39–45. Association for Computational Linguistics, 1992.

[16] Johann Strauss. Linguistic diversity and everyday life in the Ottoman cities of the Eastern Mediterranean and the Balkans (late 19th–early 20th century). *The History of the Family*, 16(2):126–141, 2011.

# A Appendix

Below you can find the Ottoman Turkish script, with each letter in its isolated, final, medial and initial forms, along with the letter names and the letter(s) they correspond to in Modern Turkish [3].

| Iso | Fin | Med | Init | Name | Tur |
|---|---|---|---|---|---|
| ا | ـا | – | | elif | a, e |
| ب | ـب | ـبـ | بـ | be | b (p) |
| پ | ـپ | ـپـ | پـ | pe | p |
| ت | ـت | ـتـ | تـ | te | t |
| ث | ـث | ـثـ | ثـ | se | s |
| ج | ـج | ـجـ | جـ | cim | c |
| چ | ـچ | ـچـ | چـ | çim | ç |
| ح | ـح | ـحـ | حـ | ha | h |
| خ | ـخ | ـخـ | خـ | hı | h |
| د | ـد | – | | dal | d |
| ذ | ـذ | – | | zel | z |
| ر | ـر | – | | re | r |
| ز | ـز | – | | ze | z |
| ژ | ـژ | – | | je | j |
| س | ـس | ـسـ | سـ | sin | s |
| ش | ـش | ـشـ | شـ | şın | ş |
| ص | ـص | ـصـ | صـ | sad | s |
| ض | ـض | ـضـ | ضـ | dad | d, z |
| ط | ـط | ـطـ | طـ | tı | t |
| ظ | ـظ | ـظـ | ظـ | zı | z |
| ع | ـع | ـعـ | عـ | ayn | ʼ, – |
| غ | ـغ | ـغـ | غـ | gayn | g, ğ, v |
| ف | ـف | ـفـ | فـ | fe | f |
| ق | ـق | ـقـ | قـ | kaf | k |
| ك | ـك | ـكـ | كـ | kef | k |
| گ | ـگ | ـگـ | گـ | gef | g, ğ |
| ڭ | ـڭ | ـڭـ | ڭـ | nef | n |
| ل | ـل | ـلـ | لـ | lam | l |
| م | ـم | ـمـ | مـ | mim | m |
| ن | ـن | ـنـ | نـ | nun | n |
| و | ـو | – | | vav | v, o, ö, u, ü |
| ه | ـه | ـهـ | هـ | he | h, e, a |
| ى | ـى | ـيـ | يـ | ye | y, i, ı |

The letters گ (gef) and ڭ (nef) are often written as ك (kef), since they are derived from it. The reader has to infer which one is meant. Also ه (he) does not have medial or initial forms if it stands for *e* or *a*.

In order to disambiguate the pronunciation, there are some available diacritics as well:

| Sign | Name | Function |
|---|---|---|
| ء | hemze | indicates an initial vowel |
| ـَ | üstün | a, e |
| ـِ | esre | ı, i |
| ـُ | ötre | o, ö, u, ü |
| ـّ | şedde | doubled consonant |
| ـْ | sükun | absence of vowel |
| ـٓ | medde | long a |
| ـً | tenvin | -an, -en |
| ـٍ | | -in |
| ـٌ | | -un, -ün |

The ones other than ء (hemze) and ـٓ (medde) are usually omitted.