A Finite State Pronunciation Lexicon for Turkish

Kemal Oflazer
Faculty of Engineering and Natural Sciences
Sabancı University,
Istanbul, Turkey
oflazer@sabanciuniv.edu

Sharon Inkelas
Department of Linguistics
University of California,
Berkeley, CA, USA
inkelas@socrates.berkeley.edu

Abstract

This paper describes the implementation of a full-scale pronunciation lexicon for Turkish using finite state technology. The system produces at its output, a parallel representation of the pronunciation and the morphological analysis of the word form so that morphological disambiguation can be used to disambiguate pronunciation. The pronunciation representation is based on the SAMPA standard and also encodes the position of the primary stress. The computation of the position of the primary stress depends on an interplay of any exceptional stress in root words and stress properties of certain morphemes, and requires that a full morphological analysis be done. The system has been implemented using XRCE Finite State Toolkit.

1 Introduction

Pronunciation lexicons are computational devices that map the graphemic representation of words to a representation of their pronunciation. They are a valuable resource in annotating speech data used in training automatic speech recognizers, and in generating accurate speech in text-to-speech systems. In this paper, we present the design and implementation of a full scale finite state pronunciation lexicon of Turkish, an agglutinating language with extremely productive inflectional and derivational morphological and an essentially infinite lexicon. The agglutinating nature of the language implies that any corpus based compilation of a word list for use in speech applications will be rather inadequate (Hakkani-Tür et al., 2002). Noun roots typically have a few hundred forms, and Hankamer estimates a few million forms for each verbal root (Hankamer, 1989). This necessitates that one employ a generative model that is able to recognize all possible words in the language and base the pronunciation lexicon on this to avoid a significant out-of-vocabulary word problem. The lexicon implemented as a finite state transducer takes as input a word form and produces all possible pronunciations of the word, paired with the corresponding morphological analyses. The pronunciations are encoded using the SAMPA encoding, and also include the marking of the position of the primary stress.[1] The dependence of the stress computation on the proper identification of morphemes and their morphotactical function requires that the pronunciation lexicon be built on top of a morphological analyzer. Although Turkish superficially seems to have an almost one-to-one mapping between graphemics and pronunciation, there are quite a number of subtle phenomena in loan words, and in morphophonology such as suffixation induced exceptional vowel lengthening and palatalizations in the surface realizations of certain suffixes depending on vowel harmony resolution. Such phenomena are not distinguished in orthography but have to be handled while representing pronunciation. There are also a number of inter-word phe-

---

[1] See http://www.phon.ucl.ac.uk/home/sampa/turkish.htm. We use the SAMPA notation to show pronunciations in the text, where necessary.

nomena such as word-initial devoicing or word-final voicing that have to be handled at the sentence level, as orthography does not reflect those interactions either.

Finite state transducers are commonly used in building pronunciation lexicons for mapping between orthographic strings and phonological strings, either as an efficient encoding of direct mapping, or a mapping involving some kind of morphological processing (Jurafsky and Martin, 2000). Recently, Gibbon et al. (2000) have described a finite state transducer to act as a pronunciation resource for an inflected languages like German. Their approach explicitly models morphologically out-of-vocabulary words by using a morphological parser that can segment and identify suffixes without an explicit root lexicon. Instead, it uses a small finite state transducer that can capture the contextual grapheme-to-phoneme mapping rules for German along with prediction of affix and root boundaries.

Some approaches also model dialectal pronunciation variations using finite state machinery. Hazen et al. (2002) describe a finite state transducer that uses a phonemic baseform dictionary. Rewrite rules implemented as weighted transducers derive phonological variations. The main goal of the transducer is to generate variations such as contractions, reductions, part-of-speech pronunciation variants, variations that depend on stress and syllable positions, etc.

This paper makes extensive use of finite state methods and two-level morphology to build a full-scale pronunciation lexicon for a language with essentially an infinite vocabulary. Its contributions lie in its use of the morphological structure to compute the position of the lexical stress and phonological changes that are not marked in graphemic representations of words, in addition to being the first such resource for Turkish.

2   Turkish

Turkish is an Ural-Altaic language, having agglutinative word structures with productive inflectional and derivational processes. Turkish word forms consist of morphemes concatenated to a root morpheme or to other morphemes, much like "beads on a string." Except for a very few excep-

tional cases, the surface realizations of the morphemes are conditioned by various regular morphophonemic processes such as vowel harmony, consonant assimilation and elisions. The morphotactics of word forms can be quite complex when multiple derivations are involved.

2.1   Aspects of Turkish Phonology

Overviews of Turkish phonology can be found in Clements and Sezer (1982), van der Hulst and van de Weijer (1991) and Kornfilt (1997). Turkish has a 8-vowel inventory which is symmetrical around the axes of backness, roundness, and height: /i, y, e, 2, a, o, 1, u/ which correspond to *i, ü, e, ö, a, o, ı,* and *u* in Turkish orthography. Suffix vowels typically harmonize in backness, and (if high) in roundness to the preceding stem vowel (compare e.g., *ev+ler* /evler/ 'house-plural' to *at+lar* /atlar/ 'horse-plural'), although there are several suffixes whose vowels do not harmonize.[2] Many roots are internally harmonic but many others are not; these include loanwords (e.g., *kitap* /kitap/ 'book', from Arabic) as well as some native words (e.g., *anne* /anne/ 'mother'). Further, vowel harmony does not apply between the two components of compounds.

Turkish has 26 consonants: /p, t, tS, k, c, b, d, dZ, g, gj, f, s, S, v, w, z, Z, m, n, N, l, 5, r, j, h, G/. /G/ represents the velar fricative or glide corresponding to the historical voiced velar fricative that was lost in Standard Turkish; but we treat it as a consonant for the purposes of this work. On the other hand, orthography uses only 21 letters for consonants: /g/ and its palatal counterpart /gj/ are written as *g*, while /k/ and its palatal counterpart /c/ are written as *k*, /5/ and /l/ are written as *l*, /v, w/ are written as *v* and /n/ and its nasal counterpart /N/ are written as *n*. Palatalized segments (/gj, c, l/) contrast with their nonpalatalized counterparts only in the vicinity of back vowels (thus *sol* is pronounced /so5/ when used to mean 'left' vs. /sol/ when used to mean 'note in scale'). In the neighborhood of front vowels, palatality is predictable (*lig* /ligj/ 'league').

Plosives (/b, d, g/) typically devoice syllable-finally (thus *kitab+a* /ci-ta-ba/ 'book-dative' but *kitap* /ci-tap/ 'book', *kitap+lar* /ci-tap-lar/ 'book-plural'). Suffix-initial plosives assimilate in voice

---

[2]We use − to denote syllable boundaries and + to denote morpheme boundaries wherever appropriate.

to the preceding segment (thus *kitap+ta* /ci-tap-ta/ 'book-locative' but *araba+da* /a-ra-ba-da/ 'car-locative').

Velar consonants (/g/ and /k/) reduce to /G/ at most root-suffix boundaries; thus *sokak* /sokak/ 'street', *sokak+ta* /so-kak-ta/ 'street-locative' but *sokağa* /so-ka-Ga/ 'street-dative'. In certain dialects a syllable-final /G/ may manifest itself as the lengthening of the preceding vowel.

Turkish syllable structure allows open and closed syllables but no onset clusters.[3] Only a subset of consonant clusters are permitted in coda position. Vowel length is phonemic, and long vowels are in complementary distribution with coda consonants; long lexical vowels will shorten when forced into a closed syllable (thus /za-ma:-na/ 'time-dative' but /za-man-da/ 'time-locative').

2.2 Stress in Turkish Words

Turkish has lexical stress: each word has exactly one primary-stressed syllable.[4] Some roots are lexically stressed[5] and certain suffixes are pre-stressing, meaning if not overridden, they will stress the preceding *syllable*. A word composed of only unstressed free and bound morphemes exhibits the default stress pattern where the last syllable is stressed. In a word with stressed root and/or prestressing suffixes, the stress of the leftmost such morpheme will prevail (see e.g., Inkelas (1999) for an overview.)

In place names and foreign names used in Turkish, a different default pattern is used, termed here the "Sezer" stress pattern, in tribute to its description in Sezer (1981). For such words the antepenultimate syllable is stressed when it is heavy (meaning containing a long vowel or ending in a consonant) and the penultimate is light (meaning ending in a short vowel), e.g., /ˈan-ka-ra/; otherwise stress is penultimate (e.g., /is-ˈtan-bul/, /a-ˈda-na/. The Sezer pattern can be imposed on any word if used as a place name (thus *kandil+li* /kan-dil-ˈli/ 'oil lamp-with', but *Kandilli* /kan-ˈdil-li/ (same word used as place name).

3 Computational Considerations

The problem of grapheme-to-phoneme mapping for Turkish is simpler than for languages such as English or French. Orthography more or less maps one-to-one to pronunciation yet there are quite a number of cases where orthography is ambiguous. These cases usually stem from the fact that a loan word (usually from Arabic, Persian or French) is a homograph of another Turkish word. The once used accent marks to mark the distinctions are not longer consistently used, if at all, and one is left to rely on the context for inferring the correct pronunciation. The other major source of pronunciation ambiguity is the location of primary stress in the word. As we saw, stress is determined by an interplay of any Sezer/exceptional stress in root words and the stress marking properties of morphemes. Certain morphemes which are homographs (but marking different morphosyntactic features) may appear in the same inflection paradigm in morphotactics but have different stress marking properties. For instance, the +*ma*/+*me* suffix in the verb paradigm will (depending its position in the suffix sequence) either mark negative polarity or an infinitive derivation. So, a word form like *okuma* would either mean the imperative 'don't read' or the infinitive 'to read/reading'.[6] In the imperative reading, the stress will be on the *syllable* preceding the +*ma* suffix, while in the other reading the suffix is neutral and stress is on the last syllable. Thus, morphological analysis is necessary to determine the morpheme structure which along with any stress markers in the root morpheme, then determines the location of the stress. On the other hand, in an application context such as text-to-speech, the appropriate pronunciation of the word has to determined by a morphological disambiguation and/or word sense disambiguation process. For instance, morphological disambiguation of the readings of the word *okuma* would be necessary to select the appropriate pronunciation in a context, while a process akin to word sense disambiguation would be necessary to disambiguate the appropriate pro-

---

[3]Except in a few loanwords such as *angstrom* /angs-trom/.
[4]The existence of secondary stress is controversial.
[5]We call this *exceptional stress*.

[6]In addition to a nominal reading *ok+um+a*, 'arrow-1sg possessive-dative' meaning 'to my arrow' which has the same pronunciation as the infinitive reading.

nunciation of the word *sol* in Section 2.1.[7] Application level disambiguation requires the availability of the morphosyntactic features so that morphological interpretations and hence the appropriate pronunciation can be selected using contextual information with statistical and/or symbolic means (Hakkani-Tür et al., 2002).

It is however necessary to generate the representations of pronunciation and morphological analysis in a paired and parallel fashion. Generating them separately and independently would not be of much use. It would not be possible to associate a given pronunciation with an analysis as this in general an *n*-to-*m* mapping as shown in Figure 1.
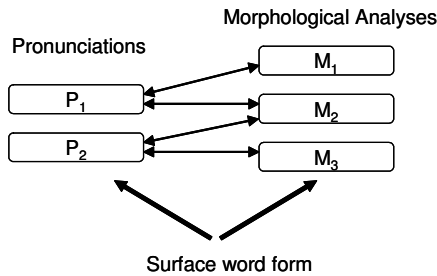
Figure 1: General relationship between pronunciations and morphological analyses

These considerations have prompted us to build the pronunciation lexicon on the scaffolding provided by the wide coverage morphological analyzer for Turkish (Oflazer, 1994) that we have built using XRCE finite state tools (Karttunen and Beesley, 1992; Karttunen, 1993; Karttunen et al., 1996)

4 The architecture of the pronunciation lexicon

4.1 The Word Pronunciation Transducer

The word pronunciation lexicon transducer is the composition of a series of transducers that transform a surface form into all possible and ambiguous parallel representations of its pronunciation and morphological features. The overall internal structure of the transducer is shown in Figure 2. All the boxes in this figure are finite state transducers, and in implementation, they are all com-

posed at compile-time to give one (very large) transducer. Let us now describe the function each of these transducers in detail:
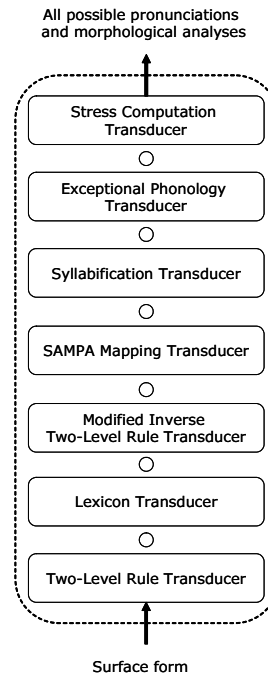
Figure 2: The structure of the word pronunciation lexicon transducer

The *Two-level Rule Transducer* at the bottom implements the morphographemic mapping described by a set of parallel two-level rules (Koskenniemi, 1983). It is the intersection of the transducers for about 35 morphographemic rules that capture the morphographemics of Turkish (Oflazer, 1994). It maps the surface representation of a word into possible lexical representations.[8] For example the surface form `karın` would map to five lexical forms:[9]

Lexical Form	Gloss
kar+Hn	your snow/profit
kar+[n]Hn	of the snow/profit
kar+[y]Hn	mix (it)!
karı+[H]n	your wife
karın	belly

The next transducer is the *Lexicon Transducer* comprising the root and the affix lexicons. In addi-

---

[7]Though the two are not senses of a word in the lexicographic sense.

[8]Though such lexical representations do not necessarily make distinctions not required by morphographemic processes.

[9]H represents a lexical meta-phone that denotes a high vowel unresolved for frontness and roundness. `[..]` denotes segments of morphemes that are deleted on the surface.

tion to the standard ordering of the suffix lexicons in the inflectional and derivational paradigms, it also comprises a couple hundred finite state constraints motivated by morphosyntactic and semantic concerns. These constraints impose fine grained tuning on word structures and significantly tame the overgeneration of the paradigm-based morpheme lexicon ordering. In the context of the pronunciation lexicon the *Lexicon Transducer* has two main functions:

1. it produces all possible morphosyntactic feature representations of the free and bound morphemes,

2. it replicates the lexical form at its output possibly augmented with any stress markers induced by *specific prestressing bound morphemes*.

Continuing the example earlier, the five lexical forms above would map to the following at the output of this transducer.[10]

| Input Lexical Form |
| Output of the Lexicon Transducer |

```
kar+Hn
  (kar)kar+Noun+A3sg(+Hn)+P2sg+Nom
kar+nHn
  (kar)kar+Noun+A3sg+Pnon(+nHn)+Gen
kar+yHn
  (kar)kar+Verb+Pos(+#p#yHn)+Imp+A2sg
karɪ+Hn
  (karɪ)karɪ+Noun+A3sg(+Hn)+P2sg+Nom
karɪn
  (karɪn)karɪn+Noun+A3sg+Pnon+Nom
```

A couple of remarks are in order here:

1. The morphological analysis process is essentially complete. We have an interleaved representation of lexical morphemes (between `(...)`) and the morphosyntactic features they map to. The concatenation of the contents in pairs of parentheses comprise the original lexical form with possible addition of certain

stress markers, while the rest when concatenated gives the morphological analysis.

2. The morpheme `+yHn` in morphotactics is a prestressing morpheme (hence the marker `#p#`), that is, it *may* eventually cause the primary stress to appear in the *syllable* before this marker.

From this point upwards in the structure, we carry the morphological features around, manipulating the lexical representation sandwiched between the `(...)` to generate the representation of the corresponding pronunciation.

The *Inverse Two-level Transducer* is essentially (but not exactly) the inverse of the *Two-level Transducer*. We have the same set of rules and a slightly different set of (inverse) feasible pairs (different for a variety of technical reasons.) The only difference in the rules is that the context regular expressions of the two-level rules are modified to ignore the `(,)` symbols, the stress markers, and the feature symbols outside the parentheses. The function of this transducer is to map the lexical form back to the *surface morphemes*, the concatenation of which will give the original surface form (plus any stress markers.) So, for the five outputs above we will get the following:

| Output of the Inverse Two-level Rule Transducer |

```
(kar)kar+Noun+A3sg(ɪn)+P2sg+Nom
(kar)kar+Noun+A3sg+Pnon(ɪn)+Gen
(kar)kar+Verb+Pos(#p#ɪn)+Imp+A2sg
(karɪ)karɪ+Noun+A3sg(n)+P2sg+Nom
(karɪn)karɪn+Noun+A3sg+Pnon+Nom
```

Now that we have the original surface form of the word, the surface morphemes can now be mapped to the representation of their pronunciations by the *SAMPA Mapping Transducer*. This transducer employs a separate lexicon which maps from the root words and a subset of their morphosyntactic features into a representation of their pronunciation in the SAMPA standard of phonetic encoding. The root words explicitly encoded here are those whose pronunciation (including stress) can not be handled by a default mapping. All other root words and all bound morphemes are handled by a grapheme level default mapping lexicon. The mapping of the roots involves a number of issues:

1. Any Sezer/exceptional stress in root words are generated during this mapping. For example

---

[10]Note that some default feature are produced with zero morphemes. The morphological features used in the paper, other than the obvious POSs are: `Imp`: imperative mood, `+P2sg`: 2nd person possessive agreement, `+A1sg`: 1st person singular agreement, `+A2sg`: 2nd person singular agreement, `+A3sg`: 3rd person singular agreement, `+Pnon`: No possessive agreement, `+Nom`: Nominative case, `+Gen`: Genitive case, `+Pos`: Positive Polarity, `+Neg`: Negative Polarity, `+Become`: Become verb, `+Caus`: Causative verb, `+Prog1`: Progressive aspect, `+Past`: Past tense `^DB` denotes a derivation boundary.

the monomorphemic root word `Avrupa` (Europe) would map to `awr#S#upa` (/aw-'ru-pa/) which has the Sezer stress on the second syllable.

2. Homograph roots with different pronunciations and/or different stress locations have to be handled by this transducer. For instance, the word `kar` when used to mean *profit* would be pronounced as /car/ but when used to mean *snow* (or when used as a verb), would be pronounced as /kar/.

3. Certain root (such as compounds) which go through morphographemic changes during affixation have to be listed here by explicitly modeling any pronunciation changes if they can not be handled by the default mapping.

Thus the outputs above now become:

Output of the SAMPA Mapping Transducer
Gloss

```
(car)kar+Noun+A3sg(1n)+P2sg+Nom
  your profit
(kar)kar+Noun+A3sg(1n)+P2sg+Nom
  your snow
(car)kar+Noun+A3sg+Pnon(1n)+Gen
  of the profit
(kar)kar+Noun+A3sg+Pnon(1n)+Gen
  of the snow
(kar)kar+Verb+Pos(#p#1n)+Imp+A2sg
  mix (it)
(karı)karı+Noun+A3sg(n)+P2sg+Nom
  your wife
(karın)karın+Noun+A3sg+Pnon+Nom
  belly
```

The concatenation of the segments within (...) comprise the preliminary SAMPA encoding of the word's encoding. A number of minor phenomena have to be fixed in the SAMPA representation. These are handled by the exceptional phonology transducer later.

The *Syllabification Transducer* performs the syllabification of the complete SAMPA encoding. The syllabification essentially involves breaking up (by inserting a –) the consonant cluster between any two vowels into the coda of the left syllable and the onset of the right syllable. The implementation of this essentially consists of the composition of a series of finite state transducers implementing replace rules for inserting the syllable boundary marker in the right place among the consonants between two vowels. The outputs of

the *Syllabification Transducer* for the cases listed earlier would be follows.

Output of the Syllabification Transducer

```
(ca-r)kar+Noun+A3sg(1n)+P2sg+Nom
(ka-r)kar+Noun+A3sg(1n)+P2sg+Nom
(ca-r)kar+Noun+A3sg+Pnon(1n)+Gen
(ka-r)kar+Noun+A3sg+Pnon(1n)+Gen
(ka-r)kar+Verb+Pos(+#p#1n)+Imp+A2sg
(ka-rı)karı+Noun+A3sg(n)+P2sg+Nom
(ka-rın)karı+Noun+A3sg+Pnon+Nom
```

The *Exceptional Phonology Transducer* handles a set of phenomena for certain exceptional roots and morphemes. The most important of these is the handling of the lexically long vowel in selected roots. For such roots, the last vowel in the root will be coded as a long vowel when it is part of an open syllable and as a short vowel when it is part of a closed syllable. It turns out that *kar* (/car/) is one of those root words (but not *kar* (/kar/)), and in the examples above, the root is followed by the surface morpheme /1n/ which forces the last vowel in the root to an open syllable /ca-/. So /a/ is lengthened and for those we get:[11]

```
(ca:r)kar+Noun+A3sg(1n)+P2sg+Nom
(ca:r)kar+Noun+A3sg+Pnon(1n)+Gen
```

A second common phenomenon handled by this transducer is the palatalization of /5/ to /l/ and of /k/ to /c/ in certain suffixes when they are followed or preceded by front vowels /e, i, y/.[12] Since this is not a morphographemic process, the underlying morphological analyzer is oblivious to this in earlier stages.

The output of this transducer has the correct SAMPA encoding except that the stress location is not yet determined.

The *Stress Computation Transducer* computes the location of the primary stress by examining any Sezer/exceptional stress markings in the roots and prestressing markers in the morphemes. The stress is determined in a series of steps:

1. All prestressing markers (`#p#`) interspersed in the SAMPA representation, that have a prestressing marker or a Sezer/exceptional stress marker somewhere on their left, are removed,

---

[11]Since the original morphological analyzer did not make vowel length distinctions (as they were not needed while handling the morphographemic process), the process has to be handled here as a vowel length adjustment. The end result from a computational point of view does not change.

[12]/2/ (*ö* in orthography) does not appear in any suffixes.

that is, the left-most marker "wins." For example, the representation at this point of the surface word *taşlaştıramıyorduk* ('we were not being able to petrify (them)') would be

```
(taS)taş+Noun+A3sg+Pnon+Nom
 (-laS)^DB+Verb+Become
 (-t1-r)^DB+Verb+Caus
 (a-#p#m)^DB+Verb+Able+Neg
 (1-j#p#or)+Prog1
 (-#p#du)+Past
 (k)+A1pl
```

where three morphemes have prestressing markers. This step deletes all such prestressing marker except the first one (in the surface morpheme `(a-#p#m)`).

2. If there is a Sezer/exceptional root stress marker left then that is also the location of the final stress. So for example the word *pencerede* (on the window) would have the output `(pen-dZ^e-re)pencere+Noun+A3sg+Pnon (-de)+Loc`.[13]

3. If there is a prestressing marker (which should be the only marker left at this point), then the vowel of the preceding syllable receives the stress. Thus the word above in item 1 would have the final representation of its pronunciation `taS-laS-t1-r^a-m1-jor-duk`.

4. If there are no stress markers at this point, then the final stress mark is inserted just before the vowel of the last syllable (all except the second from the last in the examples below).

For representational purposes the final stress mark is then moved to the preceding syllable boundary. For the examples that we have been tracing all along, the final outputs will be:

Output of the Stress Computation Transducer

```
(ca:-"r)kar+Noun+A3sg(1n)+P2sg+Nom
(ka-"r)kar+Noun+A3sg(1n)+P2sg+Nom
(ca:-"r)kar+Noun+A3sg+Pnon(1n)+Gen
(ka-"r)kar+Noun+A3sg+Pnon(1n)+Gen
("ka-r)kar+Verb+Pos(1n)+Imp+A2sg
(ka-"r1)karı+Noun+A3sg(n)+P2sg+Nom
(ka-"r1n)karın +Noun+A3sg+Pnon+Nom
```

The only exceptions to these rules are forms of the question clitic and the emphasis clitic. Both clitics are bound morphemes. But, the question clitic along with any following morphemes and

---

[13]Monosyllabic roots with Sezer/exceptional stress have a different behavior. Any prestressing morphemes will override the root stress in such words.

the emphasis clitic are written as a separate token, though their initial vowel will harmonize with the last vowel of the previous token. They are technically prestressing morphemes but with the rules above they can not make any change to the stress of the previous token (which can either override the clitics' stress or have a word final stress which would be the "previous syllable" that the clitic morphemes would stress). Thus these clitics do not themselves carry any stress marker.

4.2 Inter-word Interactions

There are also a couple of pronunciation phenomena that occur due to various inter-word interactions within a sentence. These phenomena are not marked in orthography.

1. If the last consonant of a word with at least two syllables is one of /p, tS, t/ and the next word starts with a vowel, the last consonant is voiced and pronounced as /b, dZ, d/ respectively (except for a limited set exceptional lexical forms). For example *şarap içiyorum* (I am drinking wine) would actually be pronounced as /Sarab/ /itSijorum/. This is the analogue of the consonant voicing process that is reflected to orthography when it is on a stem-morpheme boundary within a word.

2. If the token preceding an emphasis clitic *de/da* ends in a voiceless consonant, then the initial d is pronounced as /t/. For example, *kitap da* would be pronounced as /ci-tap/ /ta/, while *masa da* would be pronounced as /masa/ /da/.

Although these phenomena can also be implemented by finite state means on top of the word-level pronunciation transducer described above, by extending it so that it can handle a sequence of tokens, there are a number of engineering issues that prevent doing this in a pure finite state approach. Any unknown word would make the whole process fail even if all the other words in the sequence are known. We have decided thus to implement the inter-word interactions by a non-finite state external process.

4.3 Some implementation details

The complete sentence pronunciation model is implemented using the XRCE finite state tools, *xfst*,

lexc, and *twolc*. Apart from various lexicons making the morphological analyzer, the complete system is described by about 500 regular expressions. The resulting transducer has about 6.5 million states and about 9 million transitions. A complete build will require about 30 minutes on 1.7 Ghz Pentium 4 running Windows 2000.

The underlying morphological analyzer is very high coverage and implements all word formation processes of Turkish. It has a noun root lexicon of about 25K entries, a verb root lexicon of about 5K entries and a proper noun lexicon of about 40K entries. The current version of the pronunciation lexicon does not attempt to vocalize various tokens such as abbreviations and tokens containing non-alphabetical characters such as digits and punctuation.

5 Conclusions

We have presented the design and implementation of a finite state pronunciation lexicon for Turkish and agglutinating language with an infinite lexicon. The lexicon produces a representation that encodes in parallel, the SAMPA representation of the all possible pronunciations of the word along with the corresponding morphological analyses. The correct computation of the pronunciation and the location of the stress requires a full morphological analysis be done so the pronunciation has been built on top of a two-level morphological analyzer. The system has been implemented using the XRCE finite state tools.

6 Acknowledgements

References

George N. Clements and Engin Sezer. 1982. Vowel and consonant disharmony in Turkish. In Harry van der Hulst and Norval Smith, editors, *The structure of phonological representations, part II*, pages 213–255. Foris, Dordrecht.

Dafydd Gibbon, Ana Paula Quirino Simoes, and Martin Matthiesen. 2000. An optimised FS pronunciation resource generator for highly inflecting languages. In *Proceedings of LREC 2000*, Athens, Greece.

Dilek Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4).

Jorge Hankamer. 1989. Morphological parsing and the lexicon. In W. Marslen-Wilson, editor, *Lexical Representation and Process*. MIT Press.

Timothy J. Hazen, I. Lee Hetherington, Han Shu, and Karen Livescu. 2002. Pronunciation modeling using a finite-state transducer representation. In *Proceedings of the ISCA Workshop on Pronunciation Modeling and Lexicon Adaptation*, Estes Park, Colorado, September.

Sharon Inkelas. 1999. Exceptional stress-attracting suffixes in Turkish: Representations vs. the grammar. In *The Prosody-Morphology Interface*. Cambridge University Press, Cambridge.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, New Jersey.

Lauri Karttunen and Kenneth. R. Beesley. 1992. Two-level rule compiler. Technical Report, XEROX Palo Alto Research Center.

Lauri Karttunen, Jean-Pierre Chanod, Gregory Grefenstette, and Anne Schiller. 1996. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328.

Lauri Karttunen. 1993. Finite-state lexicon compiler. XEROX, Palo Alto Research Center– Technical Report.

Jaklin Kornfilt. 1997. *Turkish*. Routledge, London.

Kimmo Koskenniemi. 1983. Two-level morphology: A general computational model for word form recognition and production. Publication No: 11, Department of General Linguistics, University of Helsinki.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2).

Engin Sezer. 1981. On non-final stress in Turkish. *Journal of Turkish Studies*, 5:61–69.

Harry van der Hulst and Jeroen van de Weijer. 1991. Topics in Turkish phonology. In Hendrik Boeschoten and Ludo Verhoeven, editors, *Turkish Linguistics Today*. E. J. Brill.