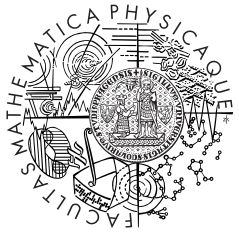
Processing of Turkic Languages



Sibel Ciddi

Faculty of Mathematics and Physics

Charles University in Prague

Thesis Supervisor:

RNDr. Daniel Zeman, Ph.D.

Co-Supervisors:

Prof. Dr. Hans Uszkoreit

Dr. Yi Zhang

Specialization:

Mathematical Linguistics

A thesis submitted for the degree of

European Masters in Language and Communication Technologies

2012 - 2013

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Acknowledgements

I am thankful to everyone who has supported me during my journey sharing both my happiness and difficulties. I give my special gratitude to my local coordinators and supervisor in Prague, and the LCT program for giving me the opportunity to study in Europe. Finally, I am thankful to my sister, İdil, without whose support I could not have made it through this far.

Sibel Ciddi
Prague, 2013

Abstract

Title: Processing of Turkic Languages
Author: Sibel Ciddi
Department: Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic
Supervisor: RNDr. Daniel Zeman, Ph.D.
Abstract: This thesis aims to present several combined methods for the morphological processing of Turkic languages, such as Turkish, which pose a specific set of challenges for computational processing, and also aims to make larger data sets publicly available. Because of the highly productive, agglutinative morphology in Turkish, data sparsity—besides the lack of the publicly available large data sets—impose difficulties in natural language processing, especially with regards to relying on purely statistical methods. Therefore, we evaluate a publicly available rule-based morphological analyzer, TRmorph, based on finite state transducers. In order to enhance the efficiency of this analyzer, and to expand its lexicon; we combine statistical and heuristics-based methods for the named entity processing (and construction of gazetteers), morphological disambiguation task and the multiword expression processing. Experiment results obtained so far point out that the use of heuristic-methods provides promising coverage increase for the text being processed by TRmorph, while the statistical approach is used as a back-up for more fine-grained tasks that may not be captured by pattern-based heuristics approach. This way, our proposed combined approach enhances the efficiency of a morphological analyzer based purely on FST constructions.
Keywords: Morphological analysis, disambiguation, FSTs, NE processing, and detection of multiword expressions.

Abstract

Název práce: Processing of Turkic Languages
Autor: Sibel Ciddi
Katedra: Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic
Vedoucí diplomove práce: RNDr. Daniel Zeman, Ph.D.
Abstrakt: Tato práce se zabývá několika kombinovanými metodami morfologického zpracování turkických jazyků, zejména turečtiny. Součástí našich snah bylo i obstarání větších zdrojů jazykových dat, než jaké jsou v současnosti k dispozici, a jejich zpřístupnění veřejnosti. Počítačové zpracování turečtiny zahrnuje specifickou sadu problémů spojených zejména s vysoce produktivní, aglutinační morfologií. Rozsah veřejně dostupných dat je s ohledem na čistě statistické metody nedostatečný a pro účely strojového učení jsou tato data příliš řídká. Z tohoto důvodu vyhodnocujeme veřejně dostupný morfologický analyzátor TRmorph, založený na konečných převodnících, tedy na pravidlech. Snažíme se rozšířit záběr a slovník tohoto analyzátoru; kombinujeme statistické metody s heuristikami pro rozpoznávání pojmenovaných entit (a konstrukci zeměpisných slovníků), zjednoznačnění morfologické analýzy a zpracování víceslovných výrazů. Výsledky dosavadních experimentů s heuristickými přístupy ukazují slibné rozšíření pokrytí textu TRmorphem. Statistické metody používáme jako záložní řešení pro jemnější úlohy, které nelze snadno zachytit heuristickými pravidly. Tímto způsobem náš hybridní systém rozšiřuje uplatnění morfologického analyzátoru, jenž je sám postaven čistě na pravidlech.
Klíčová slova: morfologická analýza, zjednoznačnění, konečný převodník, zpracování pojmenovaných entit, rozpoznávání víceslovných výrazů.

Contents

Chapter 1

Introduction

There have been many research studies, and special interests groups around morphologically rich languages with efforts to solve natural language processing (NLP) problems, concerning their complex morpho-syntactical and morpho-phonological structures. Combined with the lack of data resources problem (aka. low resource), in dealing with morphologically rich languages, morphological analysis of Turkic languages–*which is one of the main agglutinative languages, besides the other Altaic-Uralic language family, such as Finnish, Hungarian*–still continues to provide a lot of unresolved questions for researchers in the NLP field.

The existing research methodologies and various processing tools do attempt to take advantage of recent developments in NLP. However, due to the complex structure of agglutinative languages and the lack of resources, the state-of-the-art implementations of various NLP tools, such as parsers, part-of-speech taggers morphological analysers, and Named Entity Recognition (NER) systems for these languages still cannot be effectively compared with other NLP tools developed for less complex languages that have access to a lot more resources.

In order to alleviate some of these current issues, this thesis aims to assess currently available NLP tools for the morphological analysis of Turkish language, and propose new extensions and methods with proven effectiveness in order to complement the missing components, and improve the accuracy and the efficiency of the existing tools. Finally, it aims to provide the researchers and academicians with publicly available data resources that can be used for further research in similar issues, or for the extension of the methods that will be described in this thesis.

With these goals in mind, the following chapter 2 discusses morphological processing and analysis in a general context and describes the main features of Turkish morphology that sets up the ground for the motivation in pursuing this

research. Chapter 3 continues to discuss the previous works and theories in morphological analysis of Turkish, with a greater focus on one of the open-source morphological analyzer tools, TRmorph[1], [2], and discusses the evaluation of TRMorph on various data sets that were available at the time of this research.

The following chapters and sections, thereafter discuss the three main, interrelated issues concerning the morphological analysis of Turkish, thus the TRmorph tool; and describe our proposed methods for improving those issues. Namely, these target issues are Named Entity (NE) labelling *(and proper noun recognition)*, morphological disambiguation, Multiword Expression (MWE) detection and processing.

More precisely, chapter 4 briefly discusses the previous works in NE labelling and the recognition of proper nouns in Turkish, and then describes the proposed methods used for the recognition of such lexical units and for improving the lexicon. Chapter 5 continues with the discussion of *context-based* morphological disambiguation and related previous works, and goes on to describe the proposed method for improving morphological ambiguity in Turkish lexicon. And finally, Chapter 6 follows up with a discussion of the previous works in the detection and processing of MWEs, and finishes up by describing the proposed method for the morphological analysis of MWEs in the lexicon.

As last, the final chapter 7 discusses the conclusions gathered from this research, and describes some of the limitations that remain to be challenging. Finally, it provides more questions that are yet to be resolved for further research.

[1]http://www.let.rug.nl/ coltekin/trmorph/index
[2]https://github.com/coltekin/TRmorph

Chapter 2

Motivation

2.1 Morphological Processing

In written text processing, morphological analysis is one of the most important tasks that most NLP tools need as a supplement to their final toolkit because it serves as the base for the development of more comprehensive natural language processing tasks. For example, any kind of NLP application that provides a part-of-speech tagger or a parser, or other tools that are used in the development of machine translation systems, need to have a morphological analyzer as a pre-requisite for their own application to work, because these applications often rely on annotated *(labelled)* text. Therefore, if we think of such natural language processing applications having an hierarchical workflow order; morphological analyzers would be placed in one of the initial steps of the workflow that enable the larger NLP tools and systems to proceed to the next steps required for their own set of tasks. If we examine the steps that make up the morphological processing task, then we can divide those steps into their own sub-tasks. In that case, those steps that lead to morphological processing as a whole can be described as *morphological analysis*, and *morphological generation*.

Morphological Analysis. Most commonly, morphological analyzers are implemented as a finite-state transducer. Their main task is to map a given word form to its all possible morphological tags. For example, the word form *'drinks'* may be mapped to its morphological tags and return as its output: `drink+V+3p+Sg` which would denote that the word form being analyzed is a verb, in third-person, singular. Another potential analysis could also show that the word form *drinks* may be mapped to the morphological tags, showing: `drink+N+3p+Pl`, denoting that *drinks* could also be a noun, in third-person, plural form.

As discussed in a greater detail in Beesley and Karttunen (2003), the morphological analysis task–*which is often called as 'lookup' task as well*–returns as successful only when the word form that leads to such an analysis has previously been described *(often via a set of various rules)* for that language. This process requires matching of the symbols from the input words to verify them against the pre-defined symbols *(and rules)* for that language. Such a process implies that a morphological analysis of a word form becomes successful if and only if the required pre-steps have already been done. Otherwise, the analyzer returns nothing. Therefore, in the implementation of a morphological analyzer, how the analysis output is generated becomes trivial, when considering the most important thing is the pre-definition of symbols *(and rules)*, which would result in the analyzed output of the word form given as input.

Morphological Generation. Considering how the morphological analysis task is done with the finite-state-transducers, we can describe similar set of processes and steps for the task of morphological generation. This becomes possible once a finite-state-transducer is built; then it can be considered as a bidirectional processor, which can infer both the potential set of all morphological tag sequences for a given word form, and also the word form given a set of morphological tag sequences.

In other words, as we can see from the description of the task of a morphological analyzer as providing *all* potential analyses of a word form–*which was shown by the ambiguous word form example 'drinks'*–then it is assumed that the task of morphological generation is to map the morphological tag sequences to their associated word forms. This implies that the morphological generation task is required to perform the opposite of the tasks done by the morphological analyzer–in a backwards direction–given a set morphological tag sequences, it is expected to match the set of tag sequences to the word form they are associated with. In this case, if a morphological processor is given the set of tags as `drink+N+3p+Pl`, or `drink+V+3p+Sg`; in either case, the expected word form would be *'drinks'*.

As it is the case with morphological analysis task; in the same way, the morphological generation task also returns a successful matching output, *if and only if* the given set of specific morphological tag sequences has already been defined for that language. In other words, for the generator to return a matching output, via a set of specific rules, and definitions, the finite-state-transducer needs to be taught that the English morpheme *-s* can lead both to a plural form of a *regular* noun, and also the third-person, singular, present tense form of a *regular* English verb.

Given these descriptions of morphological analysis and morphological generation tasks, we see that in natural language processing applications and tools, the role of morphological processing cannot be undermined. This is not exactly because morphological processing is *not* avoidable, due to being used as an initial step in a natural language processing tool; but because the types of output that can be obtained from a morphological processor can also become useful in the further stages of an NLP task. Because morphological processors have the bidirectional capacity to provide both the word forms *(given tag sequences)*, and also the set of morphological tag sequences *(given the word forms)*; the output produced by a processor can be used at different stages of the development of a tool.

However, it is also important to point out that—as we can from the examples of *regular* English verb forms, and *regular* English nouns—the rule-based nature of morphological processors may also make them susceptible to some of the challenging problems that are commonly observed in morphological processing in general. Such problems may become even more complicated and harder to define depending on language specifications. For example, in Turkish, considering the word *'sular'* (water: v., n.), looking at its morphological analysis and generation, we should be able to see the following parts of speech and different derivations based on the same surface form:
For the *all* analyses of *'sular'*:

```
apply up> sular
sula<v><t_aor><3p>
sula<v><t_aor><3s>
su<n><pl>
su<n><pl><3p>
su<n><pl><3s>
```

For the generation to be obtained from verb form, $sula < v >< t\_aor >< 3s >$:

```
apply down> sula<v><t_aor><3s>
sular
Sular
SULAR
```

For the generation to be obtained from noun form, $su < n >< pl >$:

```
apply down> su<n><pl>
sular
Sular
SULAR
```

However, even if this example may look like a straightforward case of a morphological analysis and generation; we cannot always assume that it is possible to generate *any* word form given its tag-sequence. Because pattern-based rules may not always be general enough to make language specific assumptions, we may not always be able to generate the corresponding word form of a tag-sequence, if that tag-sequence is based on previous observations, and the knowledge of previous rule generations. This issue can be demonstrated with an example, in Turkish, by looking at the instrumental noun case suffix: *-le, -la*
If we analyze the word, *'şarapla'* (wine+inst. as in *with wine*), we get the following derivations:

```
apply up> şarapla
şarap<n><ins>
şarap<n><ins><3p>
şarap<n><ins><3s>
```

If we put this analysis in a mini experiment, relying on these derivations obtained from the analysis of *'şarapla'*, we assume that the word stem is *'şarap'*, and the suffix *-la* is the instrumental case. In this experiment, we make a generalization and derive a word form from *'water'* → *'su'* to make the *hypothetical* instrumental case form *'with water'* → *'sula'*. According to our previous assumptions and observations from the word *'şarapla'*, then when we analyze the word form *'sula'*; we should be able to get the same derivations that we observed for *şarapla*:

```
apply up> sula
sula<v><t_imp><2s>
```

Wrong assumption. The morphological analysis of the word form *sula* contradicts our previous assumption—which was based on the derivations of the word form *şarapla*. If we want to see the word form of *su-* in instrumental case, then we generate it by $su < n >< ins >$:

```
apply down> su<n><ins>
suyla
Suyla
SUYLA
```

Notice the */y/* between *su-* and *-la*, which was not in our initial assumption[1] based on the previous example of the word form *şarapla*. Therefore, this mini

[1]The -y infix is used only with certain suffix forms when the word stem ends with a vowel as in *su*, and it is followed by such suffixes starting with a consonant, as in *-le, -la* instrumental case. For such nouns, the exception rules must be made in advance.

experiment shows that not all tag sequences that are rule-based patterns can be directly applicable to make general assumptions; and that there might be a variety of other factors that might give the final word form of words in a language lexicon. This shows us that it is important to identify language-specific, certain points that may pose challenging problems in morphological processing. As described in Beesley and Karttunen (2003), the two biggest challenges in morphology come out as the morphosyntax (*aka. morpho-tactics*), and morphophonology of a given language.

These problems can be described as 'word formation'–which is made up from morphemes, smaller parts of meaning split within a word–and 'phonological and orthographical alternation'–which is the spelling or sound of a morpheme occurring in a specific context. (Karttunen, 1991) argues that word formation process comes out as a result of *principles* that impose constraints on the combinations of stems, affixes, and other types of morphemes. On the other hand, the issue of morphological alternations stem from the fact that a single morpheme can appear in different phonological environments and different word formations without losing its original meaning.

In other words, *word formation* dictates the specific constraints on the order and combination of morphemes within a word. For example, English derivations follow a certain order. We can derive 'playfulness' from the stem 'play-'+full+ness, but not *play-ness-full. As for *alternation*, we observe that certain changes appear as context-dependent sound and spelling alternations, as in the English *-s* suffix to denote plurality of nouns, except for nouns ending with *-s, -ch, -sh, -z*, where the plural form is altered as *-es*, instead of the regular, single *-s* plural suffix. According to these irregular forms in English, for example, 'glass' becomes 'glasses', and 'church' becomes 'churches'.

In order to further examine the issue of word formation and context-dependent phonological & orthographical alternations, the following section describes and discusses how these kinds of potentially problematic morphological issues reflect on Turkish morphology, while giving a general overview on the background and the main aspects of Turkish morphology.

2.2 Turkish Morphology

Dilbilimcileştiremeyebileceklerimizden miydiniz?

As it can be seen from '*Dilbilimcileştiremeyebileceklerimizden_miydiniz*'[1], [2], the productive inflectional and derivational morphology of Turkish makes lengthy word formations–that can even be used as a whole sentence. Therefore, we can deduce that more than the order of words in a sentence, it is this complex structure of morphology that gives a sentence both the syntactical and the semantical context.

Turkish is an agglutinative language that derives words and new word forms from the existing roots via suffixation. In (Kerslake and Göksel, 2005), this word formation process is described as *'the formation of a new word by attaching an affix to the right of a root.'*

In Turkish, suffixation is done by means of derivational and inflectional suffixes. Except for borrowed or foreign words, the use of prefixes is not a part of Turkish morphology. Since the order of suffixation can follow both the derivational and the inflectional suffixes[3] *(as well as certain infixes[4])*, it becomes possible to create these kinds of long word forms. While the example used here only serves to make the point about the productive morphology–and it should be noted that the given example is not a commonly used, frequent word–we can safely assume that this agglutinating nature of the morphology in Turkish makes the word forms less common than the other languages' word forms, because the majority of word forms in Turkish appear more unique as a result of unique suffixation. This problem alone causes one of the biggest difficulties for the field of natural language processing, especially for morphological processing with statistical methods as the problem of data sparsity is unavoidable.

2.2.1 Word Formation in Turkish

Among the Turkish language linguists and grammarians, there is a general consensus that part-of-speech labels of words may not be clearly defined without a

[1] It can be translated as *'Were you one of those whom we would not be able to transform into a linguist?'*, and segmented as "Dilbilim-ci-leş-tir-e-me-yebil-ecek-ler-i-miz-den-mi-ydi-niz" leads to 36 possible analyses.For more, see:`http://en.wikipedia.org/wiki/Longest_word_in_Turkish`

[2] For more details and the derivational analysis, see Appendix B.

[3] Note that in most cases, in a complex word, derivational suffix(es) come before the inflectional suffix classes.

[4] The negation suffix -me, -ma is used as an infix.

given context because most words are derived from either nominal or verbal root forms initially, and their final surface form may be a different part-of-speech. Therefore, Hengirmen (2005) suggests that it is important to consider the meaning and the context of the word in that given sentence before determining its part-of-speech label.

As one of the main identifiers of the part-of-speech of a word, the types of suffixes determine the types of words in Turkish. In other words, we see that the types of words are distinguished according to the types of suffixes they take– whether they can take derivational suffix, or inflectional, or both. Therefore, Turkish *word formation* can be categorized as the following groups of words:

1. **Simple words**: Group of words whose stems are derived from a single part-of-speech. They take only inflectional suffixes, such as the following examples:
   'masa-**lar**' → *table**s***
   'kedi-**cik**'→ *cat**-dimun.***
   'köpek-**ler-im**' → *dog**-s-my***

2. **Complex words**: Group of words whose stems are derived from a different part-of-speech, or still the same part-of-speech, where the meaning of the stem changes [1] due to derivational suffixation *(for example, verb to verb, noun to noun, or verb to noun, or verb to adjective, etc.)*. Following a derivational suffix, they can take inflectional suffixes as well, as it is shown by the following examples:
   'uyku-**lu**' → *sleep**-y***: from noun to adjective.
   'dal**gın**' → *(to) drift**-der.*** >>'absent-minded': from verb to adjective.
   'kitap-lık-**lar**' → *book**-der. + -plr.*** >> 'bookshelves': from noun to noun.
   'sev**gi**' → *love**-der.*** >> 'love-*ing*': from verb to noun.

3. **Compound words**: Group of words that are formed as a combination of two words written together. Sometimes one of the words lose its original meaning, sometimes both lose its original meaning, and sometimes both preserve their original meaning. In all cases, they can take both derivational and inflectional suffixes, such as:
   *'dil+bilim+**ci**'* → 'dilbilim**ci**', >> language+science+**der.**: linguist

[1] Note that these are the types of words that can be ambiguous if there is an overlap between certain derivational and inflectional suffixes, e.g.: *-ecek, -acak* suffixes can be used both as tense inflection when added onto the verbs, and also it can make verbs into adjectives when the word occurs in a certain context. For example, in *'Şiiri sonuna kadar okuy-acak'* (s/he **will read** the poem till its end) where 'okuy-acak' is a verb, and in *'Şiiri okuy-acak çocuk geldi'* (The child who **will read** the poem has arrived), where 'okuy-acak' is an adjectival.

2.2.2 Morphophonology *(sound alternations)* in Turkish

As for the morphophonology of Turkish, as it was briefly shown with the examples in the previous chapter, we see that the following phonological phenomena–which mainly has to do with vowel harmony and consonant changes–give their emphasis to the word stems and cause certain alternations (and certain irregularities) occurring in specific contexts. As it is described in (Kerslake and Göksel, 2005), the forms of suffixes are conditioned by the vowels and consonants that precede them. In that case, vowels in the following environments go through alternation:

Initial consonants and vowels in suffixes are conditioned by the consonants and vowels in the preceding syllable. How these constraints are conditioned can be shown by the following vowel chart in 2.1, which shows the categorization of the vowel sounds according to their position in terms of frontness, backness, roundness and unroundness, and highness and lowness:

	FRONT	**BACK**
	Rounded & Unrounded	Rounded & Unrounded
HIGH	i, ü	ı, u
LOW	e, ö	a, o

Table 2.1: Chart of Turkish Vowels

The vowel alternations are conditioned by two types of *vowel harmony*[1] rules:

1. **Two-way Vowel Harmony:** This type of vowel harmony, which is occasionally called *'front-back vowel harmony'* dictates the type of vowels within a word according to their *'frontness'* or *'backness'*. According to this vowel harmony, a word cannot have have both frontal and back vowels. If we look at the plural suffix *-ler, -lar*, we see that two-way vowel harmony, where the vowel alternates between *-e* and *-a*, determines which one of these plural suffixes a noun can take. The following nouns show some of the examples:

   - *'köp**e**k'* → *'köpek-l**e**r'* (dog, dogs)
   - *'ik**i**z'* → *'ikiz-l**e**r'* (twin, twins)
   - *'k**ö**r'* → *'kör-l**e**r'* (blind, blinds)
   - *'üz**ü**m'* → *'üzüm-l**e**r'* (grape, grapes)

[1]Note: Vowel harmony rules apply to words with Turkish origin. Foreign and borrowed words may be exceptions to these rules.

- *'çoc**u**k'* → *'çocuk-l**a**r'* (child, children)
- *'dokt**o**r'* → *'doktor-l**a**r'* (doctor, doctors)
- *'om**u**z'* → *'omuz-l**a**r'* (shoulder, shoulders)
- *'ay**a**k'* → *'ayak-l**a**r'* (foot, feet)

2. **Four-way Vowel Harmony:** Secondary type of vowel harmony, also occasionally called *'rounding'* harmony imposes matching between *high* and *low* vowels. It consists only of *high-front* and *high-back* vowels: *'i, ü, ı, u'*. According to this type of vowel harmony, words ending with *low-front* vowels are followed by their *high-front* counterparts in their suffix. Namely, after a word ending with *'low-unrounded-front'* vowel, its counterpart *'high-unrounded-front'* vowel follows in the initial suffix. As a result of this, the following mapping of rounded / unrounded vowels comes out:
-e, -i → -i
-ö, -ü → -ü
-a, -ı → -ı
-o, -u → -u

   Therefore, with different suffixes, four-way vowel harmony might be applied. For example, the question-making clitic suffix *-mi* is constrained by preceding syllable's vowels and it is applied four-way vowel harmony–vowel alternates between *-mi, -mu, -mü, -mı*[1]. Some examples:

   - *'güz**e**l m**i**?'* (is it 'nice'?)
   - *'mas**a** m**ı**?'* (is it 'table'?[2])
   - *'b**u** m**u**?'* (is it 'this'?)
   - *'s**ü**t m**ü**?'* (is it 'milk'?)

Also depended on the preceding vowel and consonant in the word stem, the consonant alternation occurs in certain environments. Consonant final syllables in the stem of the word, and the suffix following the stem go through assimilation in order to have similar sounds. This type of consonant assimilation is conditioned depending on whether the final syllable of the stem ends with a voiced or voiceless consonant, and the following suffix initial sound starts with a voiced or voiceless consonant *(or a vowel)*.

[1] high-frontal-rounded: ü, high-frontal-unrounded: i, high-back-rounded: u, high-back-unrounded: ı

[2] Note that the lack of article here is intentional since Turkish does not have an overt article marker.

The chart 2.2 below shows the voiced and voiceless consonants in Turkish:

Voiced:	b, c, d, g, ğ, j, l, m, n, r, v, y, z
Voiceless:	ç, f, h, k, s, ş, t, p

Table 2.2: Chart of Voiced & Voiceless Consonants

According to this chart, two types of assimilation can be described:

1. **Voiced Consonant Alternation:** Words ending with *voiced* consonants are followed by suffixes starting with one of the following *voiced* consonants: *'c, d, g'*. For example:
   - Locative suffix *'-de, -da, -te, -ta': 'Ev'* → *'Ev-**d**e'* (Home, home-at)
   - Locative suffix *'-de, -da, -te, -ta': 'Ofis'* → *'Ofis-**t**e'* (Home, home-at)

2. **Voiceless Consonant Alternation:** Words ending with a *voiceless* consonant are followed by suffixes starting with one of the following *voiceless* consonants: *'ç, t, k'*.

   If the initial suffix starts with a vowel *(as in the dative noun case '-e, -a')*, then the final voiceless consonant in the stem of the word is alternated with its *voiced* counterpart. In that case, the final-consonant alternations in the stem occur in the following way: *p* →*b, t* →*d, k* →*g/ğ, ç* →*c*

   The examples below illustrate these alternations:
   - Locative suffix *'-de, -da, -te, -ta': 'Ağaç'* → *'Ağaç-**t**a'* (Tree, tree-on)
   - Dative suffix *'-e, -a': 'Ağaç'* → *'Ağa**c**-a'* (Tree, tree-to)
   - Locative suffix *'-de, -da, -te, -ta': 'Yatak'* → *'Yatak-**t**a'* (Bed, bed-on)
   - Dative suffix *'-e, -a': 'Yatak'* → *'Yata**ğ**-a'* (Bed, bed-to)

Besides these regular morphophonological alternations, the certain *irregular* sound changes also occur in the following contexts:

1. **Sound Derivation:** When the word stem ends with a consonant and either the immediate suffix that follows starts with a vowel, or the stem is followed by an auxiliary word that starts with a vowel[1]; the final consonant of the stem is doubled. For example: *'his + etmek'* → *'hi**ss**etmek'* (to feel)

   Additionally, the sound derivation is observed with vowels as well if a single syllable word stem takes ones of the *'-cik, -cık, -cuk, -cük'* dimunitive suffixes. For example: *'bir'* → *'bir-**i**-cik'* (one, little one)

[1]Derivation of vowels and consonants is also common with foreign or borrowed words.

2. **Vowel Drop:** In general, when a two-syllable word ending with a consonant[1] takes a suffix starting with a vowel–if the vowel in the second syllable of the stem is a high vowel *(-i, -ü, -ı, -u)*–this vowel is dropped from the stem. For example:
*'bur**u**n'* → *'bur**u**n-**u**m'* → *'burn-**u**m'* (nose, nose-my)

   Additionally, this kind of vowel drop is also observed with *complex words (as is described in the previous section)*, for verbs that are originally derived from nouns ending with a vowel, and for nouns that are originally derived from verbs ending with a vowel. For example:
*'duy**u**'*: hearing *(noun)* → *'duy**u**-mak'* → *'duy-mak'*: to hear *(noun+der.:verb)*
*'-uy**u**'*: -sleep *(verb root)* → *'uy**u**-ku'* → *'uy-ku'*: sleep *(verb+der.:noun)*

3. **Consonant Drop:** When the words ending with the consonant *-k* take one of the diminutive suffixes *'-cik, -cık, -cuk, -cük'*, the consonant-final from the stem is dropped. For example:
*'küçü**k**'*: small → *'küçü**k**-cük'*: small+dimun. → *'küçü-cük'*
*'mini**k**'*: mini → *'mini**k**-cik'*: mini+dimun. → *'mini-cik'*

   Additionally, in order to avoid double consonants in adjacent syllables, similar type of consonant drop is observed when a noun stem ending with a consonant takes a suffix that starts with the same consonant. In those cases, one of the consonants is dropped. For example:
*'A**d**'*: name → *'A**d-d**aş'*: namesake *'A**d**aş'*

4. **Vowel raising:** The final *low*-vowel *'-a, -e'* in verb stems is alternated with one of the high vowels *(-ı, -i, -u, -ü)* if the verb takes verbal aspect inflection *-yor*. In most cases, the alternation happens as the final vowel *'-a'* → *'-ı, -u'*, and *'-e'* → *'-i, -ü'*. For example:
*'kokl**a**-'* → *'kokl**u**-yor'* (*(to)* smell → *(s/he)* is smelling)
*'kayn**a**-'* → *'kayn**ı**-yor'* (*(to)* boil → *(it)* is boiling)
*'ekl**e**-'* → *'ekl**i**-yor'* (*(to)* add → *(s/he)* is adding)
*'özl**e**-'* → *'özl**ü**-yor'* (*(to)* miss → *(s/he)* is missing)

[1] This kind of vowel drop is most common with the words related to *organs*, that are made of two-syllables in the form of: 'V-CVC' or 'CV-CVC'

Chapter 3

Background & Previous Work

Up to today, among the various natural language processing tools and applications, for the morphological processing task, several morphological analyzers have been implemented. These analyzers that are being used today differ from one another according to the methods and approaches they were built upon. The two main approaches that the morphological processors are generally based on consists of rule-based methods—which make an extensive use of finite-state transducers—and statistical methods, which also differ from one another according to their learning setting, whether they use supervised, semi-supervised, or unsupervised methods. For the morphological processing of Turkish text, an additional method has also been tried by researchers in NLP field, where they have applied *hybrid* approaches—which made use of a combination of approaches based on both rule-based and statistical methods. This section gives a brief introduction for the background that has been done in the general morphological processing in NLP and also discusses the previous works that have been done for the morphological processing of Turkish.

Rule-Based Methods in Morphological Processing.

Statistical Methods in Morphological Processing.

1. **Supervised Methods in Morphological Processing**
2. **Semi-Supervised Methods in Morphological Processing**
3. **Unsupervised Methods in Morphological Processing**

A Hybrid Method for Morphological Processing of Turkish

3.1 TRmorph: A Turkish morphological analyzer

TRmorph, by Cöltekin (2010), is morphological analyzer that uses finite-state transducers in its implementations. It has been developed for the morphological analysis of Turkish primarily, by using a lexicon based on the Turkish spell-checker Zemberek[1]; however, the flexibility in its implementation also makes it adaptable to other Turkic languages. TRmorph tool was initially implemented as a two-level morphological processor that was using SFST[2] technology, with ability also to adapt to HFST[3] technology. However, it has recently been converted to using the Foma[4] compiler that maintains a *C* programming language library, and makes use of the Xerox[5] regular-expressions for the grammar rule writing syntax for the construction of finite-state automata and finite-state-transducers used in morphological analyzers.

Foma vs. SFST and HFST Toolkits.

SFST, aka. Stuttgart Finite-State Transducer, is also similar to Foma, a toolkit that uses finite-state transducers for the implementation of morphological analysers. Unlike the Foma compiler, SFST is distributed with a $C++$ transducer library. HFST, aka. Helsinki Finite-State Transducer Technology, also shares common features of the other finite-state transducer implementations. HFST version is mostly targeted at the morphological analyzers based on *weighted and unweighted* FST constructions, mainly for morphologically rich languages such as Finnish.

[1] `https://code.google.com/p/zemberek/`
[2] `http://www.cis.uni-muenchen.de/~schmid/tools/SFST/`
[3] `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/`
[4] `https://code.google.com/p/foma/`
[5] `http://www.fsmbook.com`

3.1.1 Coverage Assesment of TRmorph

METU–CoNLL TreeBank

Unannotated Texts: Wikipedia Data

Unannotated Texts: Milliyet Newspaper Data

3.2 TRmorph Evaluation

3.2.1 Evaluation of the Morpho Challenge Shared Task Data

Morpho Challenge vs. TRmorph Segmentation

3.2.2 Evaluation of the Averaged Perceptron Disambiguation Model Data

Chapter 4

Expanding & Improving Lexicon

4.1 Previous Work in Named Entity Labelling of Turkish Text

4.2 Experiment in Expanding Existing Lexicons

4.2.1 Abbreviations & Acronyms

4.2.2 Digits & Numbers

4.2.3 Borrowed Words & Foreign Origin Words

4.3 Experiment for Guesser Implementation

4.4 Experiment in Extending Lexicons with NE labels

4.4.1 Named Entity Features & Specifications

4.5 *TBD: Experiment in Statistical Methods with Classification of NEs*

4.6 Results & Evaluation for Named Entity Tagging

Chapter 5

Context-Based Morphological Disambiguation

5.1 Previous Work in Context-Based Morphological Disambiguation

5.2 Verbal Sub-Categorization

Valency Lexicon for Verbs as Extra Features

5.3 Different Learning Methods for Morphological Sequence Modelling

5.3.1 Experiment in Morphological Sequence Modelling

5.3.2 Experiment in Unsupervised Morphological Learning Model

5.4 Results & Evaluation

Chapter 6

Processing of Multiword Expressions

6.1 Previous Work in Multiword Expression Detection & Processing

6.2 Processing of Multiword Expressions with TRmorph

Chapter 7

Conclusion

7.1 Limitations

7.1.1 Lack of Data Resources

7.2 Future Work

Appendix B

`bilimcileştiremeyebileceklerimizdenmiydiniz`: *The structure of this word became famous as one of the most popular, longest words of Turkish in the linguistics and Turkish grammar circles, 'Çekoslovakyalılaştıramadıklarımızdan_mısınız?', which could be translated as 'Are you one of those whom we could not turn into a Czechoslovak?'. This word lost its popularity due to differing opinions on the discussion of whether the question clitic -mi should be considered as a separate unit or not. Later on, after the separation of the two countries, this famous longest word lost its popularity all together, and new longest words were made by the same way of suffixation.* For illustration, we show the morphological analysis of this word below:

```
bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>
bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>
bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>
bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>
```

bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><vn_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><vn_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><part_acak><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><part_acak><ncomp><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><pl><p1p><abl><q><cpl_di><2p>

bilim<n><ncomp><D_CI><adj><JN_0><n><D_lAS><v><caus><abil><neg><abil><t_fut><VJ_0><adj><JN_0><n><ncomp><pl><p1p><abl><q><cpl_di><2p>

List of Tables

References

Kenneth R. Beesley and Lauri Karttunen. *Finite state morphology*. CSLI Publications, Stanford, Calif., 2003. ISBN 1-57586-433-9. 4, 7

Cagri Cöltekin. A freely available morphological analyzer for turkish. In *LREC*, 2010. 15

Mehmet Hengirmen. *Türkçe Dilbilgisi*. Engin Yayınevi / Eğitim Dizisi, 2005. 9

Lauri Karttunen. Finite-state constraints. In *Proceedings International Conference on Current Issues in Computational Linguistics*, Universiti Sains Malaysia, Penang, 1991. 7

Celia Kerslake and Asli Göksel. *Turkish: A Comprehensive Grammar*. Comprehensive Grammars. Routledge (Taylor and Francis), New York, 2005. 8, 10