DIRETRA, A CUSTOMIZABLE DIRECT TRANSLATION SYSTEM: FIRST SKETCHES

Alëna Aksënova, Lomonosov Moscow State University
Marina Ermolaeva, Lomonosov Moscow State University

1. Introduction

DIRETRA is a direct translation system designed for Turkic languages. Turkic languages are agglutinative and have rich and complex morphology; therefore, the primary goal is to provide a word-for-word translation of a given text, reflecting the morphological phenomena of the source language (SL) as precisely as possible.

DIRETRA includes three modules: the parser, which outputs gloss sequences for the source language; the mapper, which transforms gloss sequences of the source language into those of the target language (TL); and the generator, which creates a representation in the target language. The system has been designed for Turkish; the next step is to implement other Turkic languages as well. The structure of the system is shown using the example of nominal inflection.

2. Morphological parsing

The first module of the system converts raw sequences of SL into gloss lines. The words are processed right-to-left: first all possible suffixes are found, then the remaining part is compared to the stem dictionary, cf. (Eryiğit&Adalı 2004). The parser can analyze morphology even if the stem is absent in the dictionary.

The simplest method of parsing is to store a list of all possible morphological word variants (Segalovich 2003). However, for agglutinative languages the number of possible forms is theoretically infinite (Jurafsky&Martin 2000). The approach often applied to them involves designing complicated finite-state machines where each transition corresponds to a single suffix. In such cases the implementing of new languages requires a considerable redesigning of the whole system.

The “hybrid” approach used in the DIRETRA system involves combining sequences of categories which have strictly fixed order into slots. The resulting slot system for Turkish includes two stem slots (for processing nominal compounds)[1], noun inflection

[1] Complex stems, following the "adjective + noun" or "noun + noun" pattern, can sometimes be written in one word, e.g. *sarıhumma* "yellow fever" from *sarı* "yellow" and *humma* "fever". Since this type of compounding is productive in Turkish, the whole set of complex stems cannot be stored in the stem dictionary.

(number, possession and case), noun loop (the recursive suffix -*ki*[2]), and nominal verb suffixes (copulas and adverbial markers).

The number and order of categories within each slot can be changed without modifying the system itself, which simplifies adding new languages, cf. (Akin&Akin 2007). For each slot, a list of possible affix sequences is obtained. All checks of morphotactic and phonological compatibility of the suffixes within a slot are performed at this step; thus, the time for applying these rules at runtime is reduced. The lists for each slot are then represented as tries. The analysis is performed via a finite-state machine with multiple initial transitions, where each transition corresponds to a slot instead of a single suffix.

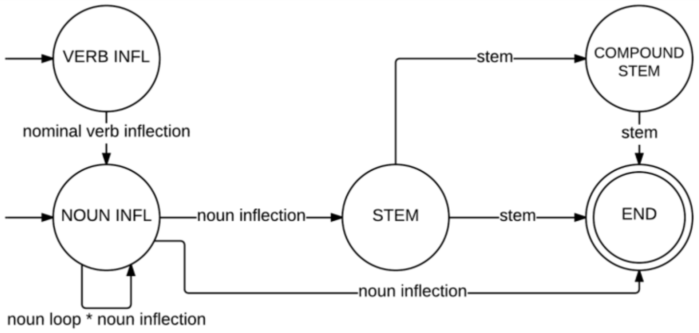

Figure 1. The FSM of the parser module

Since the first DIRETRA module can produce a considerable number of parses, the following hierarchy of outputs is applied:

known single stem > known compound stem > unknown stem

Currently, the system does not perform disambiguation. The parser yields all analyses from the highest available group in the hierarchy. Analyses in lower positions are discarded, unless there are no better options. A sample full output is shown in (1):

(1) Input: *adamdı*
Segmentation: a. adam-∅-∅-∅-dı-∅ b. ada-∅-m-∅-dı-∅
Parser output: man-SG-NPS-NOM-COP.PST-3 island-SG-P1SG-NOM-COP.PST-3

For more details on the parser see (Ermolaeva 2014).

[2] The relative -*ki* attaches to nominals in genitive or locative case. Forms already containing -*ki* can also receive case markers, leading to a loop in morphotactics (Göksel&Kerslake 2005).

3. Correspondence mapping

The next goal is to create a TL gloss line for each parser output. With Turkish as SL and English as TL, we face the well-known problem: what is syntax in some languages is morphology in the others. Indeed, Turkic languages have rich morphology, whereas English morphology is much less complicated. The complex is transformed into the simple via the following steps.

First, the affixes constituting the SL gloss sequence are divided into two groups. The first one contains affixes that correspond to morphemes in TL (morphological glosses). The second group includes all remaining glosses that can only be represented with lexical items in TL (lexical glosses). Next, the glosses are rearranged in the order required for TL. One of the linguistic generalizations that can help to deal with it is the Mirror Principle, formulated by Mark Baker:

> The Mirror Principle:
> Morphological derivations must directly reflect syntactic derivations (and vice versa).
> (Baker 1985, 375)

In compliance with this generalization, the SL gloss line is inverted[3]. Later, this “mirrored” sequence will be transformed into syntactic units in TL. Affixes that are represented with TL morphemes stay in place. TL morphological glosses (like number for English) are assigned to appropriate stems in both simple and more complex cases; this process is performed recursively if recursive suffixes like *-ki* are involved:

(2) Input: *çocuklarınkiler*
Segmentation: çocuk-lar-Ø-ın-ki-ler-Ø-Ø
Parser output[4]: child-PL-NPS-GEN-REL2-PL-NPS-NOM
CorMap output: NOM-NPS-PL-REL2-GEN-NPS-child-PL
Synthesis output: ones.owned.by.children

[3] Within a syntactic phrase, the dependents occupy (linear) positions to the right of the head in head-initial languages (e.g. English) and to the left of the head in head-final languages (e.g. Turkish). Thus, in order to capture the correct sequence of the analogous items in SL and TL, inversion is used whenever SL and TL have different values of the head-directionality parameter.

[4] Here and afterwards, only one of the possible parses for the given input is shown for the sake of space.

(3) Input: *evdekilerinki*
Segmentation: ev-∅-∅-de-ki-ler-∅-in-ki-∅-∅-∅
Parser output: house-SG-NPS-LOC-REL1-PL-NPS-GEN-REL2-SG-NPS-NOM
CorMap output: NOM-NPS-SG-REL2-GEN-NPS-PL-REL1-LOC-NPS-house-SG
Synthesis output: ones.owned.by.ones.located.in.house

The correspondence mapper also takes care of certain syntactic phenomena, e.g. the auxiliary movement in general questions. In the TL gloss sequence the presence of a question marker triggers the movement of copulas to the leftmost position (4):

(4) Input: *çocuklarmıyız*
Segmentation: çocuk-lar-∅-∅-mı-∅-yız
Parser output: child-PL-NPS-NOM-Q-COP.PRS-1PL
CorMap output: COP.PRS-1PL-NOM-NPS-child-PL-Q
Synthesis output: are.we.children.?

The structure of the correspondence mapper is illustrated by Figure 2, where lower-case letters (a, b, c) refer to morphological glosses, and upper-case letters (A, B, C) denote lexical glosses.

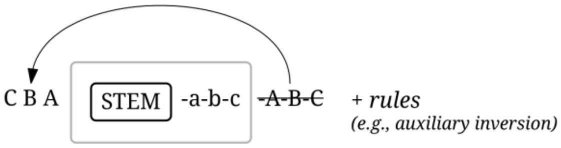

Figure 2. The design of the correspondence mapper

4. Synthesis

In the final module, a TL representation is generated. There are five types of rules: simple replacement, morphology-driven replacement, phonology-driven replacement, application of irregular forms, and statistics-based replacement.

The gloss of Dative case, replaced with the preposition *to* (5a), as well as possessors and subjects of copular predicates (5b), can serve as an example of a simple replacement rule application:

(5)	Input:	a. *adama*	b. *elmam*
	Segmentation:	adam-Ø-Ø-a	elma-Ø-m-Ø
	Parser output:	man-SG-NPS-DAT	apple-SG-P1SG-NOM
	CorMap output:	DAT-NPS-man-SG	NOM-P1SG-apple-SG
	Synthesis output:	to.man	my.apple

Morphology-driven replacement is used in the implementation of agreement of the copula and its subject; see (4). Accusative case processing serves as another example. In Turkish the overt marker of accusative is closely connected with referentiality (Differential Object Marking). Thus, if a word form bears the overt accusative marker, the definite article is inserted by the synthesis module (6a); however, no article is inserted in the presence of a possessive (6b):

(6)	Input:	a. *arkadaşı*	b. *arkadaşımı*
	Segmentation:	arkadaş-Ø-Ø-ı	arkadaş-Ø-ım-ı
	Parser output:	friend-SG-NPS-ACC	friend-SG-P1SG-ACC
	CorMap output:	ACC-NPS-friend-SG	ACC-P1SG-friend-SG
	Synthesis output:	the.friend	my.friend

Sequences containing the stem and morphological glosses are compared to the list of irregular TL word forms. With English as TL, this is primarily applicable to irregular plural forms. If the stem is on the list, the corresponding stored form is used (7a); otherwise phonology-driven rules are applied (7b):

(7)	Input:	a. *eksenler*	b. *karılar*
	Segmentation:	eksen-ler-Ø-Ø	karı-lar-Ø-Ø
	Parser output:	axis-PL-NPS-NOM	wife-PL-NPS-NOM
	CorMap output:	NOM-NPS-axis-PL	NOM-NPS-wife-PL
	Synthesis output:	axes	wives

Statistics-based replacement deals with cases where no language rules could be applied – namely, where a single SL item corresponds to multiple TL items. For example, English lacks a morphological locative case. However, for each English noun there is a locative preposition (*in*, *on* or *at*) that is used with it most often. The system employs frequencies calculated from a corpus to determine the best candidate for the replacement of the locative gloss (8).

(8)				
	Input:	a. *evde*		b. *okulda*
	Segmentation:	ev-Ø-Ø-de		okul-Ø-Ø-da
	Parser output:	house-SG-NPS-LOC		school-SG-NPS-LOC
	CorMap output:	LOC-NPS-house-SG		LOC-NPS-school-SG
	Synthesis output:	in.house		at.school

5. Future work

The most important goal set for the future development is to simplify the implementation of new languages. Presently, the parser module is flexible enough to handle various suffixal languages; prefixes and other affix types are to be added. Deriving the slot system and morphotactics automatically (from a relatively small corpus of glossed texts) instead of using hand-written rules is another promising possibility that could dramatically reduce the amount of manual work needed for each new language.

Introducing Deep Structure (DS) into the system will provide yet another step towards universality. The notion of Deep Structure as a generalized deep level of representation, introduced by Chomsky (1957), was first applied to computational linguistics in (Nida 1964). In Nida's model, the SL input is analyzed and transferred to the DS; a translation is synthesized from the DS representation through the restructuring process.

Currently, DIRETRA's correspondence mapper necessarily includes language pair-specific rules of the type "Language$_1$ → Language$_2$". Transforming it into a full-fledged DS module will replace them with rules of the model "Language$_1$ → DS" and "DS → Language$_2$", resulting in more available SL/TL combinations.

6. Abbreviations

1PL – first person plural verbal agreement affix; ACC – accusative; COP.PRS – present-tense copula; DAT – dative; GEN – genitive; REL1 – *-ki* after locative; REL2 – *-ki* after genitive; LOC – locative; NOM – nominative; NPS – non-possession; P1SG – possessive affix of the first person singular; PL – plural; Q – question marker; SG – singular.

7. References

Akin, A.A./Akin, M.D. (2007): Zemberek, an open source NLP framework for Turkic Languages. In: Structure 10.

Baker, M. (1985): The Mirror Principle and Morphosyntactic Explanation. In: Linguistic Inquiry 16, 373-416.

Chomsky, N. (1957): Syntactic Structures. The Hague: Mouton.

Ermolaeva, M. (2014): An adaptable morphological parser for agglutinative languages. In: R. Basili/A. Lenci/B. Magnini (eds.): Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014. Vol. I. Pisa University Press, 164-168.

Eryiğit, G./Adalı, E. (2004): An Affix Stripping Morphological Analyzer for Turkish. In: Proceedings of the IASTED International Multi-Conference on Artificial Intelligence and Applications. Innsbruck, 299-304.

Göksel, A./Kerslake, C. (2005): Turkish: A Comprehensive Grammar. London, Routledge.

Jurafsky, D./Martin, J.H. (2000): Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, N.J.: Prentice Hall.

Nida, E. (1964): Towards a Science of Translating. Leiden: E.J. Brill.

Segalovich, I. (2003): A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In: MLMTA, CSREA Press, 273-280.