A HYBRID MACHINE TRANSLATION SYSTEM FROM TURKISH TO ENGLISH

by
Ferhan Türe 2008

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
August 2008

A HYBRID MACHINE TRANSLATION SYSTEM FROM TURKISH TO ENGLISH

APPROVED BY:

Prof. Dr. Kemal Oflazer.....

(Thesis Supervisor)

Asst. Prof. Dr. Esra Erdem.....

Asst. Prof. Dr. Hakan Erdoğan.....

Asst. Prof. Dr. Yücel Saygın.....

Asst. Prof. Dr. Hüsnü Yenigün.....

DATE OF APPROVAL.....

to my wife Elif
&
my family

Acknowledgements

First I would like to express my gratitude to my advisor Kemal Oflazer, for his help throughout my thesis. I would also like to thank Esra Erdem, Hakan Erdoğan, Yücel Saygın, and Hüsnü Yenigün, for their valuable comments and suggestions. I am indebted to TÜBİTAK for its financial support during my studies.

I would like to thank my colleagues and friends, who have made life easier for me. I am very grateful to my parents and family, for their continuous love and support. Finally, I am very lucky to have my wife Elif with me throughout this tough period, and would like to thank for her endless love, support, and patience.

A HYBRID MACHINE TRANSLATION SYSTEM FROM TURKISH TO ENGLISH

Ferhan Türe

M.S. Thesis, 2008

Thesis Supervisor: Prof. Dr. Kemal Oflazer

Keywords: Machine Translation, Turkish

ABSTRACT

Machine Translation (MT) is the process of automatically transforming a text in one natural language into an equivalent text in another natural language, so that the meaning is preserved. Even though it is one of the first applications of computers, state-of-the-art systems are far from being an alternative to human translators. Nevertheless, the demand for translation is increasing and the supply of human translators is not enough to satisfy this demand. International corporations, organizations, universities, and many others need to deal with different languages in everyday life, which creates a need for translation. Therefore, MT systems are needed to reduce the effort and cost of translation, either by doing some of the translations, or by assisting human translators in some ways.

In this work, we introduce a hybrid machine translation system from Turkish to English, by combining two different approaches to MT. Transfer-based approaches have been successful at expressing the structural differences between the source and target languages, while statistical approaches have been useful at extracting relevant probabilistic models from huge amounts of parallel text that would explain the translation process. The hybrid approach transfers a Turkish sentence to all of its possible English translations, using a set of manually written transfer rules. Then, it uses a probabilistic language model to pick the most probable translation out of this set. We have evaluated our system on a test set of Turkish sentences, and compared the results to reference translations.

TÜRKÇE'DEN İNGİLİZCE'YE MELEZ BİR BİLGİSAYARLA ÇEVİRİ SİSTEMİ

Ferhan Türe

M.S. Tezi, 2008

Tez Danışmanı: Prof. Dr. Kemal Oflazer

Anahtar kelimeler: Bilgisayarla Çeviri, Türkçe

ÖZET

Bilgisayarla dil çevirisi bir doğal dildeki yazının başka bir doğal dile, anlamını kaybetmeyecek şekilde çevrilmesi işlemidir. İlk bilgisayar uygulamalarından biri olmasına karşın, şu anki en iyi sistemler bile çevirmenlere alternatif olamamaktadır. Yine de, çeviriye olan talep artmakta ve bunu karşılayacak çevirmen arzı yetersiz kalmaktadır. Uluslararası şirketler, organizasyonlar, üniversiteler, ve birçok diğer kurum günlük hayatta birçok değişik dille baş etmek durumunda, bu nedenle çeviriye ihtiyaç duymaktadır. Bu nedenle, bilgisayarla çeviri yapan sistemler çevirinin maliyetini ve emeğini, çeviri yaparak veya çevirmenlere yardımcı olarak, hafifletmek için gereklidir.

Bu çalışmada, iki değişik yaklaşımı birleştirerek Türkçe'den İngilizce'ye çeviri yapan bir melez çeviri sistemini tanıtıyoruz. Transfere dayalı sistemler iki dil arasındaki yapısal farklılıkları açıklamada başarılı iken, istatistiksel metodlar da paralel veri kullanarak çeviri sürecini açıklayıcı olasılıksal modeller oluşturabilmektedir. Melez yaklaşımda bir Türkçe cümlenin bütün olası İngilizce karşılıkları elle yazılmış transfer kurallarına dayanarak bulunuyor. Sonra, olasılıksal dil modeli bu çevirilerden en olası olanını seçiyor. Sistemimizi bir Türkçe cümle kümesinde test ettik, ve sonuçları referans çevirilerle karşılaştırdık.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

Machine Translation (MT) is a term used to describe any system using an electronic computer to transform a text in one natural language into some kind of text in another natural language, so that the original meaning of the source text is preserved and expressed in the target text ([Hutchins, 1986]). There are many reasons why scientists are interested in studying machine translation systems, but the general aim in MT research is to increase the quality and efficiency of translation, while lowering the cost.

There are approximately 7000 different spoken languages in the world. More than a hundred of these languages have 5 million or more native speakers. As technological developments occur and the world globalizes, the demand for language translation increases. International corporations, organizations, universities, and many others need to deal with different languages in everyday life, which creates need for translation. There is not enough supply of human translators to satisfy this demand, which is one reason to start developing MT systems.

Each year, billions of dollars are spent on human translation industry, mostly the translation of technical documents on international markets to a number of different languages. The European Union (EU) needs to have each document translated to a number of languages, which makes them use 13% of the EU budget for translation purposes ([Europa,]). Automating the process of translation would save much money and effort, which is another motivation to MT research.

Information available via Internet is growing rapidly, however access to a document is limited to only people that understand the language it is written in. It is impossible for human translators to cope with the increasing volume of material, whereas it is essential to make the documents accessible to most of the world. Around 50% of World Wide Web (WWW) content is written in English ([Bowen,]), and this cannot reach to most of the people due to linguistic problems. Creating a reliable MT system to translate web pages automatically would let information spread much faster and easier to all around the world.

Machine Translation was one of the first applications of computers. However, computer scientists have not been able to produce promising results as they expected. On the other hand, statistical approaches have recently proven to be very successful with large amounts of data available through the Internet, which has attracted many researchers to the field. Another reason to study MT is the scientific curiousity of finding the limits to abilities of computers and also exploring challenges in linguistics ([Hutchins, 1986]).

Although the long term goal would be producing fully automated translation with high quality and efficiency ([Hutchins and Somers, 1992]), researchers have mostly considered using MT as an improvement in translations. MT systems where human intervention helps computer processes (or vice versa) have been popular in the field. Human intervention may take place before the translation, during the process, or after translation occurs. Computers can also aid human translation by intervening in some part of the translation process, also referred as Computer-aided Translation ([Hutchins and Somers, 1992]).

1.2 Thesis Statement

Turkish is a language spoken by 75-100 million people worldwide. It is a member of the Altaic language family, being the most commonly spoken language in the subgroup of Turkic languages. This thesis describes a hybrid MT system from Turkish to English, based on the transfer system created by Avenue Project ([Peterson, 2002]). We call the

method “hybrid” in the sense that it combines two different approaches successfully.

1.3 Outline of the Thesis

The organization of this thesis is as follows: In Chapter 2, we give an overview of MT by discussing the historical development of MT systems and various approaches to MT. In Chapter 3, we describe a hybrid MT system from Turkish to English, explaining the procedure step by step and giving detailed examples. Chapter 4 presents the evaluation of the system. Finally, Chapter 5 concludes with final remarks and future work.

Chapter 2

MACHINE TRANSLATION

2.1 Overview of MT

A formal definition of machine translation is as follows: Given a sentence s in some natural language F, the goal is to find the sentence(s) in another natural language E that best explains s. We call F the source language (SL), and E the target language (TL). Consider an example translation from English to Spanish, and the gloss of each word in the Spanish translation:

English:	*Mary didn't slap the green witch.*
Spanish:	*Maria no dio una bofetada a la bruja verde.*
Gloss:	Mary not gave a slap to the witch green

In this example, English is the source language and Spanish is the target language. Another example is shown below, where the source language is English and target language is German.

English:	*The green witch is at home this week.*
German:	*Diese Woche ist die grüne Hexe zu Hause.*
Gloss:	this week is the green witch at house

A translation from English to French is shown in the following example:

English: *I know he just bought a book.*
French: *Je sais quil vient dacheter un livre.*
Gloss: I know he just bought a book

In all of these examples, the two sentences have almost equivalent meanings. The difference is mainly due to the different vocabulary, morphological properties and grammatical structure of these languages. Vocabulary is the set of words used in a language; the grammatical structure determines how words form a sentence; and morphology determines the internal structure and formation of words. Since these components are relatively similar in the languages English, French, German, and Spanish, the sentences may look similar (They are all from the Indo-European language family). Now, let us consider the following translation from Turkish to English.

Turkish: *Avrupalılaştıramadıklarımızdanmışsınız.*
Gloss: European become cause not able to we ones among you were
English: *You were among the ones who we were not able to cause to become European.*

Observe that a single-word sentence in Turkish is translated into English by using 15 words, each word corresponding to some part of the Turkish word. This is an extreme case when translating from an agglutinative language to a non-agglutinative language; but it demonstrates how different a text can be expressed in two distinct languages.

2.1.1 Challenges in MT

In order to translate from one language to another, the vocabulary, morphological properties, and grammatical structure of the source and target languages should be taken into account separately. Moreover, the morphological, syntactic and semantic differences due to these components should be handled carefully. Many challenges arise in machine translation, and some of these are explained below.

Different morphological properties is one of the greatest challenges in machine translation. In agglutinative languages, words may have many morphemes separated clearly by boundaries. On the other hand, in inflectional languages such as Russian, one morpheme may correspond to more than one morphological feature, which creates

ambiguity. In isolating languages such as Viatnamese, each word corresponds to one morpheme, while in polysynthetic languages (like Yupik) each word contains many morphemes and corresponds to a sentence in languages like English ([Jurafsky and H.Martin, 2006]).

In addition to morphological differences, another challenge in MT is syntactic differences, of which the most common is word order. Most of the major languages like English, Spanish, German, French, Italian and Mandarin have a SVO (Subject Verb Object) word order, which means that the verb of a sentence most likely comes right after the subject. Contrarily, some languages like Japanese and Turkish have SOV word order, and languages such as Arabic, Hebrew and Irish have VSO order. Word order is an important determinant of the syntactic structure of a language ([Jurafsky and H.Martin, 2006]).

English:	*He adores listening to music*
Turkish:	*O müzik dinlemeye bayılıyor*
Gloss:	he music listening to adores

Turkish and Spanish have two different versions of past tense (one for definite, the other for indefinite situations), while this distinction is not made in English. Choosing the correct past tense is a potential problem when translating from English to one of these languages. For instance, in Turkish *Ali yap+mış* and *Ali yap+tı* both mean *Ali did it*, but the former one implies that the person has not seen Ali doing it. Therefore, it is called the narrative past tense.

Furthermore, in these two languages, pronouns can be determined from an inflection of the verb, and the pronouns *he*, *she* and *it* are indicated by the same inflection. Therefore, an ambiguity occurs when translating into English for such cases. In Spanish, the sentence *Habla Turco* means either *He speaks Turkish* or *She speaks Turkish.*

Another issue is the order of adjective and noun in a noun phrase. In French and Spanish, adjectives come after nouns, while in English and Turkish, they precede nouns.

Besides syntactic differences, semantic issues may also make machine translation a challenging problem. First of all, word sense ambiguity may cause many different

English: *green witch*
Spanish: *bruja verde*
Gloss: witch green

meanings (and subsequently many different translations) of a sentence. The word *bank* may have two different meanings in English: it may mean an establishment for the custody, loan, exchange, or issue of money (as in *I put money in the bank*) or it may mean the rising ground bordering a lake (as in *We saw the river bank*).

Idiomatic phrases specific to a language should also be handled carefully. For instance, in Turkish, *kafa atmak* literally means throwing (someone) heads, but it actually is an idiom for hitting (somebody) with the head. Furthermore, some languages such as Chinese and Turkish have different words for *elder brother* and *younger brother* (*ağabey* and *kardeş* in Turkish, respectively), while others do not distinguish the two. Handling these kind of issues is challenging, and requires a significant amount of time and effort.

2.1.2 History of MT

The idea to use computers in translation began around 1945, which gave start to the first attempts to research in machine translation. In the 1950s, the US government's aim was to translate Russian text into English automatically, in order to decode Russian messages during the Cold War between the US and USSR. Several projects were funded until the mid-1960s, which turned out to be a great disappointment. Scientists and the government were expecting a working translation system to finish shortly, however research showed that the challenges in language and translation made this task more difficult than expected ([Hutchins, 1986]). In 1966, the Automatic Language Processing Advisory Committee (ALPAC) published a report stating that automatic translation systems were slower and more expensive than human translators. The ALPAC report concluded that there was no need for further MT research and systems were only helpful when assisting translators. As a result of this ALPAC report, most of the financial supports for MT research were withdrawn ([Hutchins and Somers, 1992]).

Starting with the 1970s, research gained pace at different countries, with differ-

ent motives. In Canada, systems were developed to handle difficulties arising due to the multilingual structure. An English-French system called Meteo that translated weather reports in Montreal was demonstrated in 1976 ([Buchmann et al., 1984]). In Europe, the Commission of European Communities completed an English-to-French MT system based on the previous Systran project. Later, this project was extended to complete systems for other language pairs, such as English-Italian and English-German ([Hutchins and Somers, 1992]). Another project, aiming to develop a multilingual system between all European languages was installed in the late 1970s ([Varile and Lau, 1988]). In Japan, after solving the difficulty of handling Chinese characters in 1980, many scientists started research in MT: The translation system TITRAN, the MU project at Kyoto University ([Nagao and ichi Tsujii, 1986]) and another project at the University of Osaka Prefecture are some examples of these Japanese systems ([Hutchins and Somers, 1992]).

In the early 1990s, through the growth of Internet, large bilingual corpora became publicly accessible. A bilingual corpus (plural: "corpora") is a set of aligned sentences, such that each sentence in SL is aligned with a sentence in TL. This motivated researchers to apply statistical methods to bilingual corpora, in order to automatically create a model of the translation process. In statistical machine translation (SMT) from source language F to target language E, the problem is to find the most probable translation of a sentence f in F. The idea is to build a language model for the target language, representing how likely a sentence in the target language is to be said in the first place, and build a statistical model for translation, representing how likely a sentence in the target language would translated back into f. Most successful SMT systems are explained by Koehn *et al.* ([Koehn et al., 2003]), Brown *et al.* ([Brown et al., 1993]), and Chiang ([Chiang, 2005]). SMT is explained in further detail, in Section 2.3.6.

2.2 MT between English and Turkish

Turkish is an agglutinative language with free constituent order, and the syntactic relations are mostly determined by morphological features of the words. Therefore, morphological analysis is essential to develop proper Natural Language Processing (NLP) tools

for Turkish. The commonly used morphological analyzer for Turkish was first introduced by Oflazer ([Oflazer, 1994]), a two-level analyzer implemented in PC-KIMMO environment ([Koskenniemi, 1984]). An agglutinative morphology also implies ambiguity in the morphological analysis of a word. Almost half of the words in a Turkish text are morphologically ambiguous, hence morphological disambiguation is necessary to achieve an accurate analyzer. There are many morphological disambiguators and taggers for Turkish, described by Oflazer and Kuruöz ([Oflazer and Kuruöz, 1994]), Hakkani-Tür *et al.* ([Hakkani-Tür et al., 2000]), Yuret and Türe ([Yuret and Türe, 2006]), and Sak *et al.* ([Sak et al., 2007]).

The first work on an MT system between English and Turkish was in 1981, in an M.Sc. thesis ([Sagay, 1981]). This work has been developed into an interactive English to Turkish translation system, Çevirmen. Turhan describes a transfer-based translation system from English to Turkish ([Turhan, 1997]), and an interlingua-based approach for translation from English to Turkish is shown by Hakkani *et al.* ([Hakkani et al., 1998]). There has also been recent work on implementing a wide-coverage grammar for Turkish: Çetinoğlu and Oflazer state the work of developing a Lexical Function Grammar for Turkish ([Özlem Çetinoğlu and Oflazer, 2006]). Oflazer and El-Kahlout describe the initial explorations of a Statistical MT system from English to Turkish ([Oflazer and İlknur Durgar El-Kahlout, 2007]).

2.3 Classical Approaches to MT

The well-known Vauquois triangle (Fig. 2.1) summarizes the relation between the three main steps of traditional machine translation: Analysis, transfer and generation. First, the source sentence is analyzed into an intermediate representation (Analysis), then this representation is transferred to the target language (Transfer), and finally generated into a sentence (Generation). Therefore, the idea is to take a sentence in SL and represent it in such a way that it can be transferred and re-generated into a sentence in TL. However, in practical MT systems, some of these three steps may be skipped or the approach may focus on other steps.

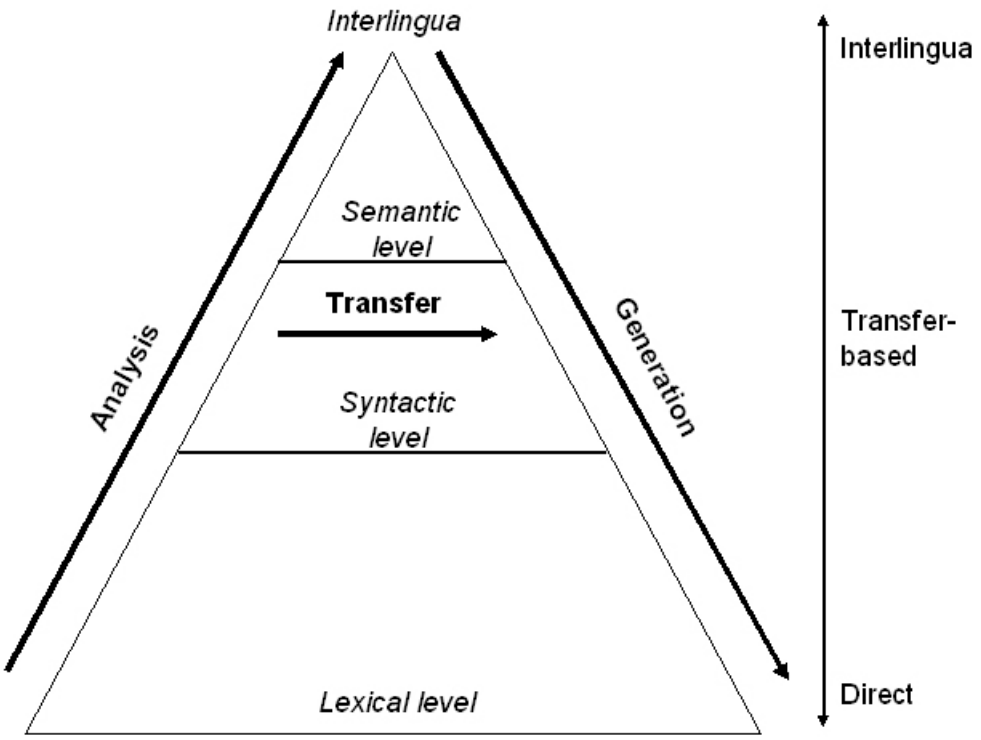


Figure 2.1: Vauquois triangle

For example, word-by-word translation requires no analysis or generation, but only the transfer step. On the other hand, interlingual translation focuses on analysis of the sentence to find a language-independent representation that captures the structure and semantics of it. After this deep analysis, it can skip the transfer step and generate a sentence in any language that will explain the interlingual representation. The word-by-word approach corresponds to the base edge of the triangle, while translation in an interlingual approach occurs at the top corner. On the mid-way of these two extreme approaches, transfer-based systems require only syntactic analysis, and a consequent transfer of the syntactic structures.

Approaches to machine translation can be analyzed according to two dimensions: Knowledge acquisition and knowledge representation. Knowledge acquisition specifies how knowledge is acquired (all manual to fully automated), and knowledge representa-

tion specifies how knowledge is represented (deep to shallow). In the following section, various MT approaches are examined according to where they fit in terms of knowledge acquisition and representation methods, and how the three steps of MT are implemented.

2.3.1 Human Translation

Human translation requires all of the three steps work internally in human mind. A translator first understands the source sentence (internally converts the semantics of the sentence into some representation), then does a structural transfer, and finally generates the target sentence from this representation. In this approach, knowledge is acquired both statistically (based on life-long exposure to language) and manually (studying linguistics at school, memorizing meaning/translation of words). The representation of knowledge is deep, a sentence is represented by its "meaning", and translated into the source language, based on this knowledge.

Human translation is the motivation of all research in MT. Various MT approaches, described below, try to mimic the way a human translates. Each MT approach is successful at some extent, but none of the current MT systems is a perfect alternative to human translation.

2.3.2 Word-by-word Machine Translation

Word-by-word translation basically aims to find a translation for each word in a sentence. It is based on the transfer step, and skips the analysis of the sentence, which places it on the base edge of the Vauquois triangle. This approach represents knowledge at the shallowest level: A sentence is generally represented by a sequence of word roots. See the example below:

Source sentence	*Ali kötü adamı evde tokatlamadı*
Word-by-word translation	*Ali bad man home slap*
Reference translation	*Ali did not slap the bad man at home*

Knowledge is acquired from a manually or automatically created dictionary. Word-by-word translation is easy to implement, and it usually gives a rough idea about the source sentence. However, the translation output is far from well-formed language, and the meaning may become distorted especially when translating from agglutinative languages like Turkish.

Word-by-word translation from German to English was attempted in 1950, and the researchers concluded that such an approach was useless ([Oswald,]). The article *der* in German could be translated into many different forms in English, such as *the*, *of the*, *for the*, *the*, *he*, *her*, *to her*, and *who*. This result proposed some analysis of the source sentence, and re-ordering of constituents to capture syntactic differences between the SL and TL.

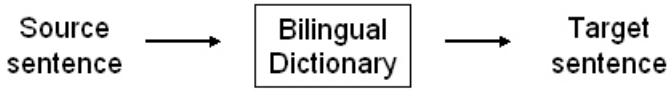


Figure 2.2: Translation procedure for word-by-word approach

2.3.3 Direct Machine Translation

Direct translation is a variation of the word-by-word approach: Each word in the source sentence is analyzed at a shallow (lexical/morphological) level, transferred to the TL by lexical translation and some local reordering, and fed to a morphological generator at the generation step. The same sentence is translated by direct approach as follows:

Source sentence	*Ali kötü adamı evde tokatlamadı*
Morphological Analysis	Ali kötü adam+Acc ev+Loc tokatla+Neg+Past[1]
Lexical transfer	Ali bad man home+Loc slap+Neg+Past
Local reordering	Ali slap+Neg+Past bad man home+Loc
Generation	Ali did not slap bad man at home

[1]Acc: accusative case, Loc: locative case, Neg: negative sense, Past: past tense

This approach represents each word in a sentence by its morphological features, and uses lexical rules to reorder constituents while doing transfer. Writing these rules does not require much linguistic expertise, and can be finished in a relatively short time with less effort, compared to approaches requiring deeper analysis.

Direct translation has been favored especially in the early years of MT research. The GAT Russian-English system implemented at Georgetown University and the Systran (System Translation) ([Hutchins and Somers, 1992]) project developed as a continuation of GAT are the most typical examples of direct translation approaches. The Systran project has continued to produce versions of the Russian-English system for many other language pairs as well ([Hutchins and Somers, 1992]).

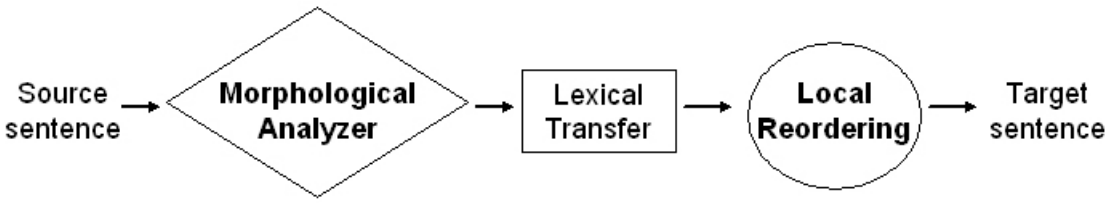


Figure 2.3: Translation procedure for direct approach

2.3.4 Interlingua-based Machine Translation

The goal of the interlingua-based approach is to form a language-independent representation (called "interlingua"), into which the source sentence is analyzed and from which the target sentence is generated. Therefore, there is no transfer step and this approach is placed on the top corner of the Vauquois triangle. Representation of knowledge is at the deepest level; the source sentence is analyzed both syntactically and semantically. A transformation from sentence to interlingual representation should be manually designed by implementers.

In order to find an interlingual representation of the sentence *Ali kötü adamı evde tokatlamadı*, we need to define the relationships NOT(SLAP(ALI, MAN, AT(HOME), WHEN(PAST))), HASCHARACTER(MAN, BAD), etc. This may seem straightforward for this example, but the concept of a global representation of semantics turns

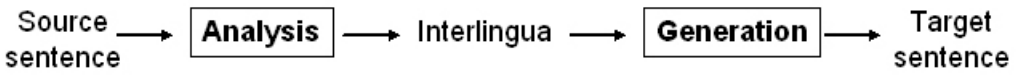


Figure 2.4: Translation procedure for interlingua-based approach

out to be very complicated. Creating a representation that covers all possible meanings, entities, and relationships in a sentence is usually not possible for large domains. Therefore, interlingua-based approach is mostly used in subdomains such as air travel, hotel reservation systems, or repair manuals. An advantage is that one does not need to implement $n(n-1)$ transfer modules for a multilingual translation system between n languages; n analyzers and n generators are sufficient. This is a motivation for communities like the European Union where a many-to-many translation system is required.

The KANT project at Carnegie Mellon University is one example to an interlingual approach ([Nirenburg, 1992]), using a logic-based knowledge representation as the "interlingua". Another interlingua-based MT system is the Rosetta project ([Appelo et al., 1987]), which uses the Montague grammar theory to link syntax and semantics ([Hutchins and Somers, 1992]). The Distributed Language Translation (DLT) project, based on a prototype written in Prolog and using an intermediate language called Esperanto, has a goal of building an MT system to translate between European languages ([Witkam, 1988]).

2.3.5 Transfer-based Machine Translation

The idea in transfer-based translation is to do a "transfer" between language-dependent abstract representations, instead of sentences. The analysis step consists of mapping the source sentence into this abstract representation, which is transferred into a similar representation in the target language. Finally, this form is mapped to a sentence in TL, during the generation step.

Transfer-based translation is placed in the middle of the Vauquois triangle, depending on how deep an analysis is required. The abstract representation is usually

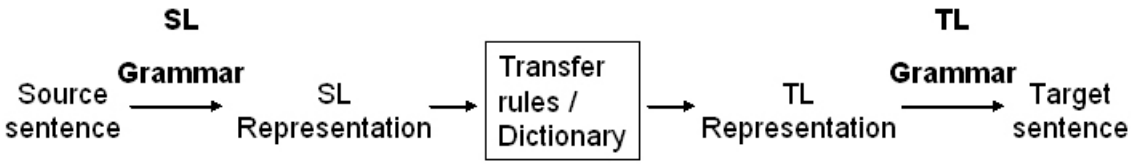


Figure 2.5: Translation procedure for transfer-based approach

the syntactic tree of the sentence, which can be derived by parsing the sentence. The syntactic transfer between corresponding sentences in Turkish and English is shown in Fig. 2.6. Turkish noun phrases *mavi ev+in* and *duvar+ı* are transferred into corresponding English noun phrases *the blue house* and *the wall*, respectively. The suffix *+in* is mapped to the preposition *of* on the English side.

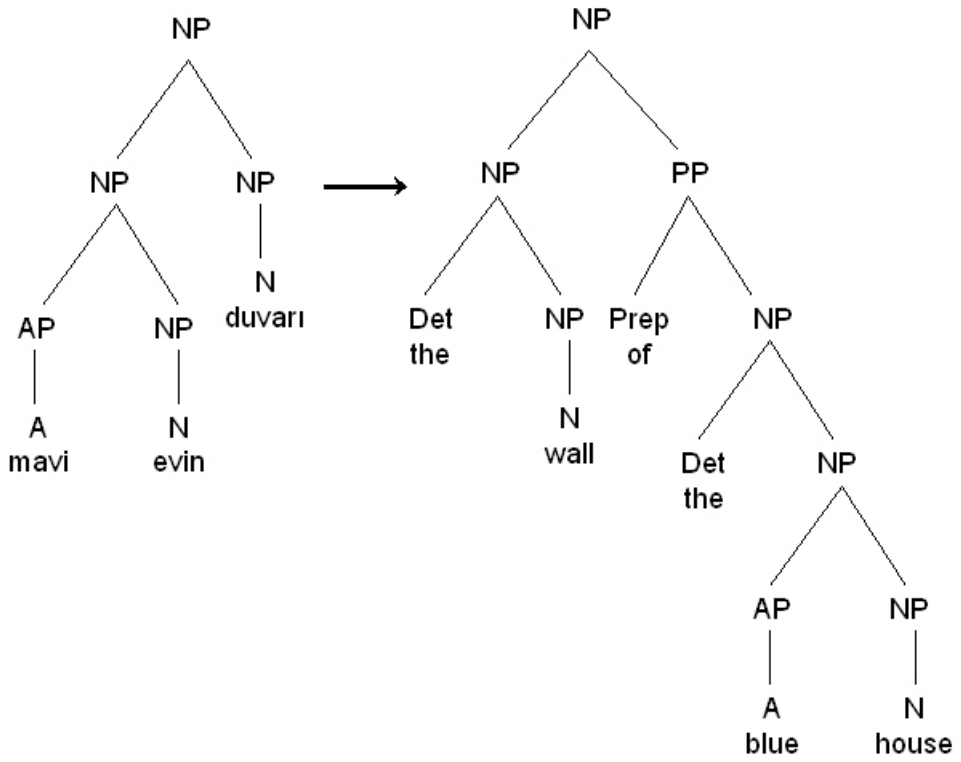


Figure 2.6: Example transfer of syntactic trees

In transfer-based translation, knowledge representation is not as deep as in the interlingual approach. The analysis and generation steps are easier than in interlin-

gual approach, since the representation is language-dependent. Transfer rules play an important role in handling the structural differences between the source and target languages, therefore it becomes easier to implement this part when the languages are similar. On the other hand, a separate set of transfer rules is required for translation of each language pair. Therefore, a transfer-based approach is costly for multilingual translation systems. Instead of manually crafted transfer rules, using machine learning techniques to learn these rules overcomes this disadvantage. Probst ([Probst, 2002]) and Lavoie *et al.* ([Lavoie et al., 2002]) describe MT systems that learn transfer rules automatically.

There are many examples of transfer-based machine translation systems. The SUSY project started around 1970, based on the successful Systran prototype; it focused on translating from and into German ([Maas, 1977]). Meteo, a French-English MT system, translated weather reports in Montreal, Canada ([Buchmann et al., 1984]). Metal is a German-English transfer-based translation system, which was implemented in late 1980s by Siemens ([Bennett and Slocum, 1985]). One of the biggest MT projects was Eurotra, a multilingual translation system, which supported translation between 72 pairs of 9 European languages ([Varile and Lau, 1988]). GETA is an MT system for translation from and into French, designed by a research group in University of Grenoble, led by Bernard Vauquois ([Joscelyne, 1987]).

2.3.6 Statistical Machine Translation

Statistical Machine Translation (SMT) is a variation of MT, which makes use of statistical tools to determine the most probable translation of a sentence. More specifically, SMT views the translation process as a "noisy channel": The sentence e is transmitted through a "noisy channel", and turns into f. The aim is to find the e such that the probability of e being the translation of the observed output f is maximized.

$$e^* = \arg\max_e P(e|f) \tag{2.1}$$

Instead of trying to approximate this probability model accurately with joint distribu-

tion, we decompose the problem using Bayes' rule.

$$e^* = \arg\max_e P(f|e)P(e)/P(f) = \arg\max_e P(f|e)P(e) \tag{2.2}$$

The denominator $P(f)$ can be neglected, since it is constant for each e. Observe that Equation 2.2 captures the essence of translation better than Equation 2.1, by viewing the process in two separate parts. In Equation 2.1, a model for $P(e|f)$ needs to describe how likely f is translated into e, as well as how well-formed an English string e is. In Equation 2.2, a model for $P(f|e)$ concentrates only on the probability that e is a translation of f, regardless of how well-formed a French string f is. Additionally, a model for $P(e)$ explains the probability of e being an English string, unrelated to the translation process. The former model is called the translation model, while the latter is called the language model ([Brown et al., 1993]). The *argmax* operator encodes the process of searching the English string e that maximizes the given probability. This process, called "decoding", is proven to be NP-hard by Knight ([Knight, 1999]).

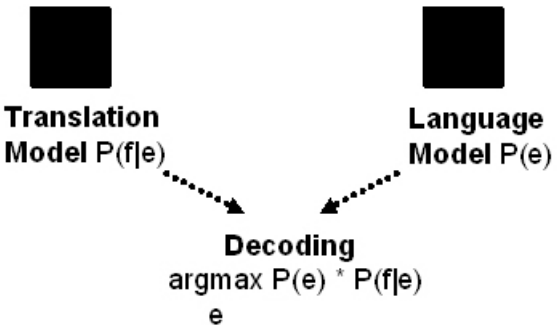


Figure 2.7: Statistical Machine Translation

Language Model

For a sentence $e = w_1...w_n$, $P(e)$ can be calculated as following:

$$\begin{aligned} P(e) &= P(w_1)P(w_2|w_1)P(w_3|w_2,w_1)...P(w_n|w_{n-1},w_{n-2},...,w_1) \\ &= P(w_1)\prod_{i=2}^{n} P(w_i|w_{i-1},w_{i-2},...,w_1) \end{aligned}$$

Assuming that each word is independent, we only need to find the probability of each word separately.

$$P(e) = \prod_{i=1}^{n} P(w_i)$$

If we assume that each word is dependent only to the previous word, we have

$$\begin{aligned} P(e) &= P(w_1) \prod_{i=2}^{n} P(w_i|w_{i-1}) \\ &= P(w_1) \prod_{i=2}^{n} \frac{P(w_{i-1}w_i)}{P(w_{i-1})} \end{aligned}$$

This is called a bigram model. A more realistic assumption would be that each word depends on the last two words, which is called a 3-gram model.

$$\begin{aligned} P(e) &= P(w_1)P(w_2|w_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, w_{i-2}) \\ &= P(w_1) \frac{P(w_1w_2)}{P(w_1)} \prod_{i=3}^{3} \frac{P(w_{i-2}w_{i-1}w_i)}{P(w_{i-1}w_{i-2})} \end{aligned}$$

Consider the sentence *I watched the bird with binoculars*. For a 3-gram model, the score of this sentence is calculated as follows:

$$\begin{aligned} P(I\ watched\ the\ bird\ with\ binoculars) = & P(I) \times P(watched|I) \\ & \times P(the|I, watched) \\ & \times P(bird|watched, the) \\ & \times P(with|the, bird) \\ & \times P(binoculars|bird, with) \end{aligned}$$

Each prior probability is found by counting occurrences in given contexts. For example, the first term is the number of occurrences of I divided by number of all words in the model. The second term is the number of occurrences of *I watched* divided by

number of occurrences of *I*. Other terms are calculated similarly, and the product gives the probability of the sentence.

$$P(I) = \frac{\#\text{ occurrences of } I}{\#\text{ of words in the model}}$$
$$P(watched|I) = \frac{\#\text{ occurrences of } I\ watched}{\#\text{ occurrences of } I}$$
$$P(the|I, watched) = \frac{\#\text{ occurrences of } I\ watched\ the}{\#\text{ occurrences of } I\ watched}$$

Each of these models contain different probability values to estimate, which are called model parameters. The parameters are estimated from a monolingual corpus of the TL. A monolingual corpus consists of a large set of words in a language. For instance, The Linguistic Data Consortium (LDC), a consortium that creates, collects, and shares linguistic data, has released the Web 1T 5-gram Version 1 English corpus. It contains over 1 trillion tokens, 95 billion sentences, 13.5 million 1-grams, 314 million 2-grams, and 977 million 3-grams ([of Pennsylvania,]).

Probability values of each n-gram is calculated by counting number of occurrences in the corpus. Larger context models can be more accurate, but may suffer from the data sparseness problem. For language models created from sparse data, some strings may not occur at all. To overcome this, *smoothing* is used to adjust the model to compensate data sparseness. There are many smoothing techniques that handle this issue differently, but any smoothing technique should at least assign non-zero values to strings not occurring in the data ([Zhai and Lafferty, 2004]).

Translation Model

Similar to creating a language model, translation models are created using a bilingual corpus of the SL and TL. There are several models for this procedure ([Brown et al., 1993]), but the general idea is to find a mapping for words in the source sentence into words in the target sentence. The IBM Model 3 ([Brown et al., 1993]) is based on this idea.

The parameters of Model 3 for translation from French to English are the following:[2]

- Translation parameter $t(f|e)$: probability of e being translated into f.
- Fertility parameter $n(\phi|e)$: probability that e is mapped to ϕ French words.
- Distortion parameter

  $d(i|j)$: probability that English word in position j is mapped to a French word in position i.

  $d(i|j, v, w)$: probability that English word in position j is mapped to a French word in position i, given that English has v and French has w words.

These parameters are estimated after words are aligned by the Expectation Maximization (EM) algorithm, and used to create a model that explains the translation of e into f ($P(f|e)$). The system finds the most probable translation of each word, and then finds the most probable order of these translations. Readers should refer to Brown *et al.* ([Brown et al., 1993]) for further details. Although this has been a successful model of translation, it cannot cover cases where several words in SL are aligned to a single word in TL. Phrase-based MT is an extension to the idea in Model 3, based on the goal of finding alignments between phrases in the SL and TL, not just words. This approach captures some of the syntactic transformation between languages and the semantics of a sentence better.

For example, the word *interest* in the sentence *I have no interest in money* means something completely different than the *interest* in *The interest rate is 9%*. *interest* is a part of the phrase *interest in* in the first sentence and *interest rate* in the second sentence, and the word should be treated in that sense. With a large amount of bilingual data, translations of very long phrases (even sentences) can be extracted automatically based on this idea. Phrase-based MT approaches are described by Koehn *et al.* ([Koehn et al., 2003]) and Chiang ([Chiang, 2005]).

The advantage of SMT is that most of the effort needed by human in other approaches are delegated to computers. Given enough training data, computers can learn

[2]Here, variables e and f stand for words, instead of sentences.

to translate between any language pair. Certain patterns of syntactic transformation between a pair of sentences can be learned by SMT, even though there is no explicit knowledge about the syntactic structure of either language. On the other hand, this means that an SMT system does translation by "the magic of linguistic data and statistics", instead of learning the "true" concept of translation. It may translate a sentence perfectly, but produce nonsense for a syntactically very similar other sentence, if some part of it has not been observed in the training data. This is why researchers have explored translation systems that combine the advantages of traditional and statistical approaches.

2.3.7 Hybrid Machine Translation

Hybrid approach to MT is based on the idea that syntactic and morphological information can be helpful to analyze and transfer sentences, and statistical tools can help solve ambiguities that arise in the process. Knight *et al.* ([Knight et al., 1995]) describe a hybrid MT system that finds an ambiguous semantic representation of the source sentence, which is disambiguated using a language model of TL. The "generation-heavy" MT system explained by Habash ([Habash, 2002]) and Ayan *et al.* ([Ayan et al., 2004]) finds a set of hypothesis translations using symbolic methods, and makes use of statistical approaches to find the most probable translation. Statistical tools can also be used to learn transfer rules, which are then used to transfer syntactic representations of the source and target languages ([Probst, 2005]).

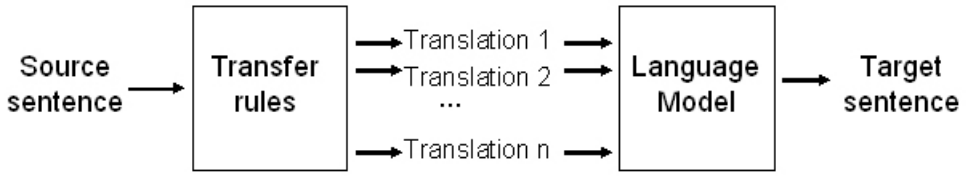


Figure 2.8: Hybrid approach

Chapter 3

A HYBRID MT SYSTEM FROM TURKISH TO ENGLISH

Our work consists of a hybrid approach to Turkish-to-English machine translation. We call our system hybrid, because it combines the transfer-based approach with statistical approaches. In this section, we first give a motivation of this approach, then summarize the procedure and structure of our system. Finally, we provide the reader with examples of input and output of the system.

3.1 Motivation

As explained in Section 2.3.7, hybrid approaches to MT have been useful to combine the advantages of symbolic transfer systems and statistical approaches. Transfer-based systems are capable of representing the structural differences between the source and target languages. On the other hand, statistical approaches have proven to be helpful at extracting knowledge about how well-formed and meaningful a sentence or translation is.

Our system uses manually crafted transfer rules to parse the Turkish sentence and map the parse tree into corresponding parse trees in English. Then, an English language model is used to choose the most probable translation. The first part corresponds to the traditional transfer approach, while the second part makes use of statistical MT techniques.

3.2 Overview of the Approach

3.2.1 The Avenue Transfer System

The Avenue project ([Peterson, 2002]) is a machine translation project that has two main goals: (i) to reduce development time and cost of MT systems, and (ii) to reinstate the use of indigenous languages officially in other countries. Different research groups around the world use the Avenue transfer system in order to create MT systems for their local languages. The system consists of a grammar formalism, which allows one to create a parallel grammar between two languages; and a transfer engine, which transfers the source sentence into possible target sentence(s) using this parallel grammar.

A parallel grammar between Turkish and English contains rules that describe the structure of all well-formed Turkish sentences and the structure of the corresponding English translations of these sentences. The parallel grammar consists of a set of lexical and transfer rules. Lexical rules serve as a Turkish-English bilingual dictionary, that transfers each word to its English translation. Transfer rules serve as a syntactic transfer mechanism, that parses a Turkish sentence and transfers the possible parse trees into corresponding parse trees in English.

Our system takes a Turkish sentence as input, and finds all morphological analyzes of each word by feeding it to a Turkish morphological analyzer ([Oflazer, 1994]). All of the analyses are converted into a lattice that Avenue understands. Using the parallel grammar, Avenue finds all possible English translations of the input sentence. Finally, an English language model is applied to find the most probable translation.

3.3 Challenges in Turkish

As mentioned in Section 2.2, Turkish has an agglutinative morphology. This means that a single word may contain many different morphemes, with different morphological features. For instance, the root of the word *arkadaşımdakiler* is *arkadaş* (*friend*), and the suffixes *-ım*, *-da* and *-ki* indicate various properties about the root word. *-ım* is a first person singular possessive marker, changing the meaning into *my friend*; *-da* is a locative case marker, which changes the meaning into *at my friend*; *ki* changes the

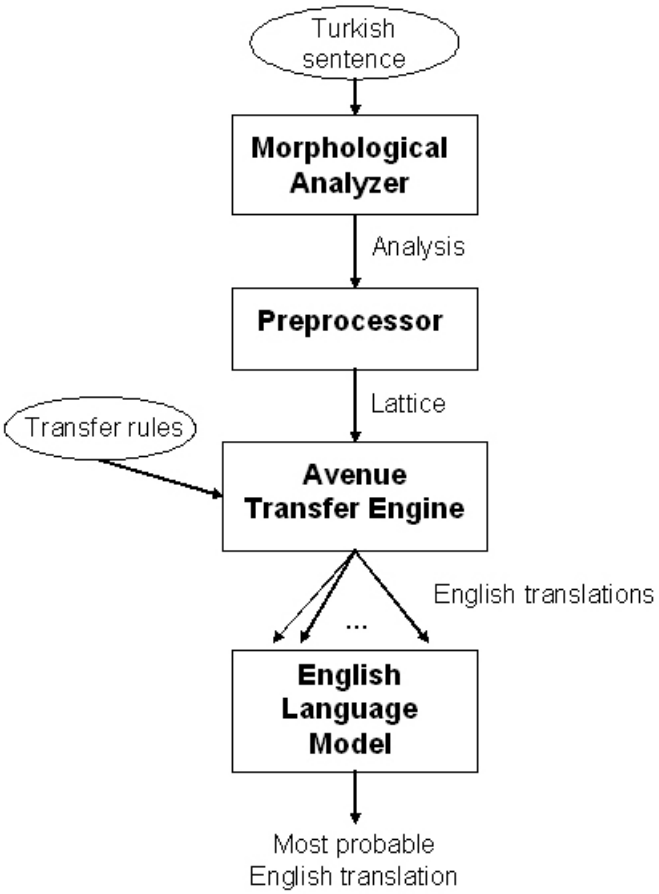


Figure 3.1: Overview of our hybrid approach

noun into an adjective, such that *arkadaşımdaki* means *(that is/are) at my friend*; and finally *-ler* changes the part-of-speech from adjective to a plural noun, changing the meaning to *the ones (that are) at my friend*. Notice that the case suffices at the end of the Turkish root correspond to prepositions preceding the English root. This example shows the morphological and grammatical distance between English and Turkish. This is one of the challenges when translating from Turkish to English, which we try to overcome by doing a morphological analysis on the source sentence.

The word order also indicates the structural differences of Turkish and English. Even though the word order of Turkish is mainly Subject-Object-Verb (SOV), words may change order freely. On the other hand, English has a rather strict Subject-Verb-Object (SVO) word order. A parallel grammar is used to handle the word order

differences. The fact that Turkish has free word order also makes it computationally difficult when grammatically parsing a sentence.

Another challenge of Turkish is about some verb markers that do not have a direct equivalent in other languages. Turkish verbs can take consecutive causative markers, which is meaningful in Turkish, but hard to translate to English. For example, consider the word *yaptırdım*, which consists of the verb root *yap* and a causative marker with past tense and first person singular possession. Although this case can be simply translated into English as *I had/made/caused (someone) do*, the verb may take another causative marker and become *yaptırttım*. This has an awkward translation as *I had (someone) make (someone else) do*, where the *someone* and *someone else* can only be determined from context. Another extension is *yaptırabildim*, which is translated as *I was able to cause (someone) do*, and another is *yaptırabilirdim* translated as *I could be able to make (someone) do*. Extracting these by statistical techniques may not be plausible, so manually written transfer rules may help translating such forms.

The agglutinative nature of Turkish has a side effect of creating ambiguous analyses. As a famous example, the word *koyun* has five morphological analyses, corresponding to five different meanings:

1. *sheep*
2. *your bay*
3. *of the bay*
4. *put!*
5. *your dark-colored one*

Almost half of the words in a Turkish running text are morphologically ambiguous ([Yuret and Türe, 2006]). Even the commonly used two possessive markers, third person singular and second person singular, may cause ambiguity. The first two nouns in the sentence *silahını evine koy*, may be interpreted as either first or second person singular. Based on this interpretation, the English translation will be one of the following:

- *put your gun into your house*
- *put his/her/its gun to your house*
- *put your gun to his/her/its house*
- *put his/her/its gun to his/her/its house*

It is difficult to distinguish between the possible translations in this case, but statistical techniques can be used to pick the translation which is most probable in a given context.

As a conclusion, there are many challenges about translating from Turkish to English. We claim to overcome some of these difficulties by a hybrid MT approach that uses a morphological analyzer for analysis, a manually-crafted parallel grammar for transfer, and statistical methods for decoding.

3.4 Translation Steps

In this section, we describe the three aspects of our approach in detail: Morphological Analysis, Avenue Transfer System, and Language Modeling.

3.4.1 Morphological Analysis

Morphological analysis is the study of the internal structure of words in a language. This internal structure consists of the subparts and features of a word, which are called morphemes. A word may have more than one morphological analysis, corresponding to different structural interpretations of the word. For instance, the word *books* may be the present tense of verb *book* or the plural form of noun *book*. A morphological analyzer is a tool that finds all morphological analyses of a given word. Since each analysis corresponds to different semantic and syntactic interpretations of words, it is essential to find all analyses.

In Turkish, we represent the morphological analysis of a word by a sequence of inflectional groups (IGs), each separated by a derivational boundary (DB). IGs include morphological features of the root and derived forms. For instance, the word

sağlamlaştırdıklarımızdaki has five IGs:

```
sağlam+Adj^DB
+Verb+Become^DB
+Verb+Caus+Pos^DB
+Noun+PastPart+A3Sg+P1Pl+Loc^DB
+Adj+Rel
```

Each marker with a preceding `+` is a morphological feature of Turkish. For instance, `P1Pl` corresponds to first person plural possession of nouns, `A3Sg` corresponds to third person singular agreement, and `Pos` corresponds to positive verbs. Each group of features separated by a `^DB` is an IG. For instance, `+Verb+Become` indicates a derivation of the adjective *sağlam* (*strong*), into a verb *sağlamlaş* (*become strong*).

We use a Turkish morphological analyzer ([Oflazer, 1994]) that uses 126 of these morphological features to describe analyses of Turkish words. Using this analyzer, we represent an analysis of a sentence as a sequence of IGs. Consider the following sentence as input:

adam evde oğlunu yendi

Firstly, each word in the sentence is analyzed by the morphological analyzer. If there are more than one analyses for a word, each of the analyses are considered separately. Table 3.1 shows the analysis output of the sample sentence.
Then, the morphological analysis of the sentence is one of the following:

$$S_1 = IG_{111} + IG_{211} + IG_{311} + IG_{411} + IG_{412}$$
$$S_2 = IG_{121} + IG_{211} + IG_{311} + IG_{411} + IG_{412}$$
$$S_3 = IG_{111} + IG_{211} + IG_{321} + IG_{411} + IG_{412}$$
$$S_4 = IG_{121} + IG_{211} + IG_{321} + IG_{411} + IG_{412}$$
$$S_5 = IG_{111} + IG_{211} + IG_{311} + IG_{421} + IG_{422}$$

Word	Morphological Analysis	IGs*
adam	ada+Noun+Nom+P1Sg+A3Sg	IG_{111}
	adam+Noun+Nom+PNon+A3Sg	IG_{121}
evde	ev+Noun+Loc+Pnon+A3Sg	IG_{211}
oğlunu	oğul+Noun+Acc+P2Sg+A3Sg	IG_{311}
	oğul+Noun+Acc+P3Sg+A3Sg	IG_{321}
yendi	ye+Verb^DB+Verb+Pass+Pos+Past+A3sg	IG_{411}^DB+IG_{412}
	yen+Noun+A3sg+Pnon+Nom^DB+Verb+Zero+Past+A3sg	IG_{421}^DB+IG_{422}
	yen+Verb+Pos+Past+A3sg	IG_{431}

* IG_{ijk} denotes the k^{th} IG of the j^{th} analysis of the i^{th} word

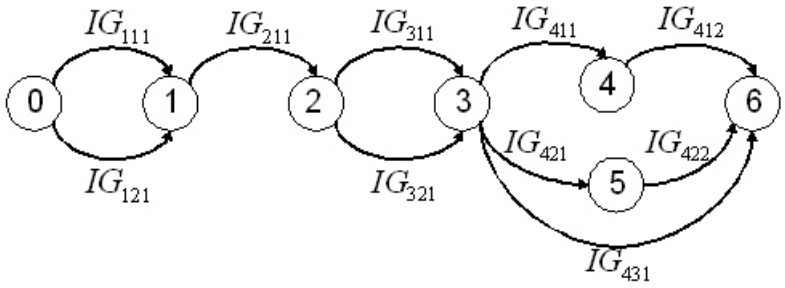
Table 3.1: Morphological analysis of words in the sample sentence

$$S_6 = IG_{121} + IG_{211} + IG_{311} + IG_{421} + IG_{422}$$
$$S_7 = IG_{111} + IG_{211} + IG_{321} + IG_{421} + IG_{422}$$
$$S_8 = IG_{121} + IG_{211} + IG_{321} + IG_{421} + IG_{422}$$
$$S_9 = IG_{111} + IG_{211} + IG_{311} + IG_{431}$$
$$S_{10} = IG_{121} + IG_{211} + IG_{311} + IG_{431}$$
$$S_{11} = IG_{111} + IG_{211} + IG_{321} + IG_{431}$$
$$S_{12} = IG_{121} + IG_{211} + IG_{321} + IG_{431}$$

The selection of an analysis S_i, $i = 1 \ldots n$ formed by possible word analyses can be viewed as selecting paths from a directed graph (or lattice), where each word or derivational boundary is viewed as a vertex and each IG is viewed as an edge between the vertices corresponding to the DBs surrounding it. The lattice that expresses the above analysis is shown in Fig. 3.2.

This lattice can be represented by a sequence of lists, where each list contains the start and end vertex number, and the features of the analysis corresponding to the edge in between. The sequence of lists representing the above lattice is shown in Fig. 3.2.

After analyzing each word in a sentence, a preprocessor converts the analyzer's output into this lattice. Each list should contain at least the four entries `SPANSTART`, `SPANEND`, `LEX` and `POS`. `SPANSTART` and `SPANEND` indicate the start and end vertices, `LEX` indicates the root/lexicon and `POS` indicates the part-of-speech of a list.




```
IG111: ((spanstart 0)
        (spanend 1)
        (lex ada)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON 1)
        (POSS-NUMBER Sg)
        (CASE Nom))

IG121: ((spanstart 0)
        (spanend 1)
        (lex adam)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON None)
        (POSS-NUMBER None)
        (CASE Nom))

IG211: ((spanstart 1)
        (spanend 2)
        (lex ev)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON None)
        (POSS-NUMBER None)
        (CASE Loc))

IG311: ((spanstart 2)
        (spanend 3)
        (lex ogul)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON 2)
        (POSS-NUMBER Sg)
        (CASE Acc))

IG321: ((spanstart 2)
        (spanend 3)
        (lex ogul)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON 3)
        (POSS-NUMBER Sg)
        (CASE Acc))

IG411: ((spanstart 3)
        (spanend 4)
        (lex ye)
        (pos Verb))

IG412: ((spanstart 4)
        (spanend 6)
        (pos Verb)
        (lex Passive)
        (POLARITY Positive)
        (TENSE Past)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg))

IG421: ((spanstart 3)
        (spanend 5)
        (lex yen)
        (pos Noun)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg)
        (POSS-PERSON None)
        (POSS-NUMBER None)
        (CASE Nom))

IG422: ((spanstart 5)
        (spanend 6)
        (pos Verb)
        (lex Zero)
        (TENSE Past)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg))

IG431: ((spanstart 3)
        (spanend 6)
        (lex yen)
        (pos Verb)
        (POLARITY Positive)
        (TENSE Past)
        (AGR-PERSON 3)
        (AGR-NUMBER Sg))
```

Figure 3.2: The lattice representing the morphological analysis of a sentence

3.4.2 Transfer

In this section, we first describe the rule formalism by examples, and then show how the transfer engine applies these rules to translate Turkish text into English text.

Rule Formalism

All rules have a unique identifier, indicated by the top constituent symbol and an integer. The head of the rule follows this identifier, which consists of production rules for both source and target sides. The source production rule is used for analysis of Turkish text, and the target production rule is used for transfer and generation of English text. At the beginning of the head, the LHS of the source and target production rules are shown, separated by `::`. Note that the feature structure of the first `S` will be referred as `X0`, and the second `S` will be referred as `Y0` hereafter. Following the symbol `:`, the right hand side (RHS) of the production rules are indicated in brackets. The RHS of the source production rule is transferred into the RHS of the target production rule. The feature structure of each source constituent of the RHS is referred as `X` followed by its position index. Similarly, target constituents are referred as `Y` followed by its position index.

In the example in Fig 3.3, the unique rule identifier is `{S,1}`. The head in this example is `S::S : [SUBJ OBJ VP] -> [SUBJ VP OBJ]`. Here, the first `S` refers to the left hand side (LHS) of the source production rule, and the second `S` refers to the constituent it transfers into, which is the LHS of the target production rule. `SUBJ` is referred as `X1`, `OBJ` as `X2`, and `VP` as `X3` throughout the rule. The corresponding target constituents `SUBJ`, `VP`, and `OBJ` are referred as `Y1`, `Y2`, and `Y3`, respectively.

Following the head of the rule, the body of the rule contains a list of alignments and equations. The alignments indicate which source constituent aligns to which target constituent. Equations have different structure and functionality; there are analysis equations, constraining equations, transfer equations, and generation equations. Analysis equations copy some of the feature structure of descendants of `X0` into `X0` when parsing the rule; transfer equations transfer some of the feature structure of `X0` into `Y0`; and generation equations copy some of the feature structure of `Y0` into its descendants. The transfer equation describes how features are passed sideways (i.e., from source side to target side) and the generation equation describes how features are transferred on the target side. Finally, constraining equations ensure the agreement of certain features of the source constituents.

In Fig. 3.3, the alignments `(x1::y1)`, `(x2::y3)`, and `(x3::y2)` indicate the order of alignments between source and target constituents. The first three equations are

```
{S,1}
S::S : [SUBJ OBJ VP] -> [SUBJ VP OBJ]
(
;Constituent alignment
(x1::y1)
(x2::y3)
(x3::y2)

;Analysis
((x0 subj) = x1)
((x0 obj) = x2)
((x0 verb) = x3)

;Unification constraints
((x2 CASE) =c (x3 casev))
((x1 AGR-PERSON) = (x3 AGR-PERSON))
((x1 AGR-NUMBER) = (x3 AGR-NUMBER))

;Transfer
((y0 TENSE) = (x0 TENSE))

;Generation
(y0 = y2)
)
```

Figure 3.3: Sample transfer rule in Avenue

analysis equations. They copy the feature structure of `x1` into the `subj` feature of `x0`, and similarly `x2` and `x3` into the `obj` and `verb` features of `x0`. The next three equations are constraining equations. The first equation ensures the `CASE` feature of `x2` is identical to the `casev` feature of `x3`. This actually serves for the case agreement of the verb and object in a Turkish sentence. The symbol `=c` guarantees both sides of the equation are non-empty, so that the rule will not unify if one of the features is missing. On the other hand, the next two equations will unify even if one of the `AGR-PERSON` and `AGR-NUMBER` features of `x1` and `x3` are missing. This equation checks for the agreement of the subject and verb of a sentence. Next comes the transfer equation, which transfers some features of `x0` into `y0`. In the example, the `TENSE` feature of `x0` is copied to `y0`. Finally, there is a generation equation, which copies features of `y2` into `y0`.

```
{NP,11}
NP::NP : [N] -> ["the" N]
(
;Constituent alignments
(x1::y2)

;Analysis
(x0 = x1)
((x0 TYPE) <= np)
((x0 DEF) <= yes)

;Transfer
(y0 = x0)

;Generation
(y0 = y2)
)
```

Analysis equations may transfer the entire feature structure of a constituent to the upper level, as shown in the above rule. Additional features can be included as well, such as features `TYPE` and `DEF` are added to `x0` in the example. This rule also illustrates the inclusion of target constituents that are not aligned to any source constituent. *the*

is inserted only on the English side, since Turkish noun phrases do not have preceding articles.

Lexical rules are special forms of transfer rules, where the RHS of the production rules (`x1` and `y1`) consist of a single word. In the following example, these words are *yüz* and *face*. The LHS constituents of lexical rules (`x0` and `y0`) indicate the part-of-speech of these words, which is `N` in this example. For words which can be analysed as different part-of-speech values, we include a constraint on the word's `POS` value and separate rules for each of these values. The rules for noun, verb, and cardinal analyses of the word *yüz* are shown below.

```
{N,10613}
N::N |: ["yuz"] -> ["face"]
(
;Constituent alignment
(X1::Y1)

;Unification constraint
((x0 POS) =c "Noun")
)
```

```
{V,2648}
V::V |: ["yuz"] -> ["swim"]
(
;Constituent alignment
(X1::Y1)

;Unification constraint
((x0 POS) =c "Verb")
)
```

```
{Card,1041}
Card::Card |: ["yuz"] -> ["hundred"]
(
;Constituent alignment
(X1::Y1)

;Unification constraint
((x0 POS) =c "Num")
)
```

Transfer process

The lattice in Fig. 3.2 is the input to the transfer engine. In this lattice, the morphological features of each IG is shown by a corresponding feature structure, and its place in the lattice is represented by features `SPANSTART` and `SPANEND`. The Avenue transfer engine searches for a complete path in the lattice, by applying transfer rules to candidate paths until a constituent that covers the entire lattice is found. Our lattice starts at vertex 0 and ends at vertex 6, so the transfer engine should consider a path that covers these vertices. For instance, `IG111-IG211-IG311-IG411-IG412` and `IG121-IG211-IG321-IG431` are sequences of IGs that are candidates for a complete

path. Fig. 3.4 shows these two paths, respectively. A sequence of IGs is a complete path if and only if it covers all of the lattice and it is accepted by the parallel grammar.

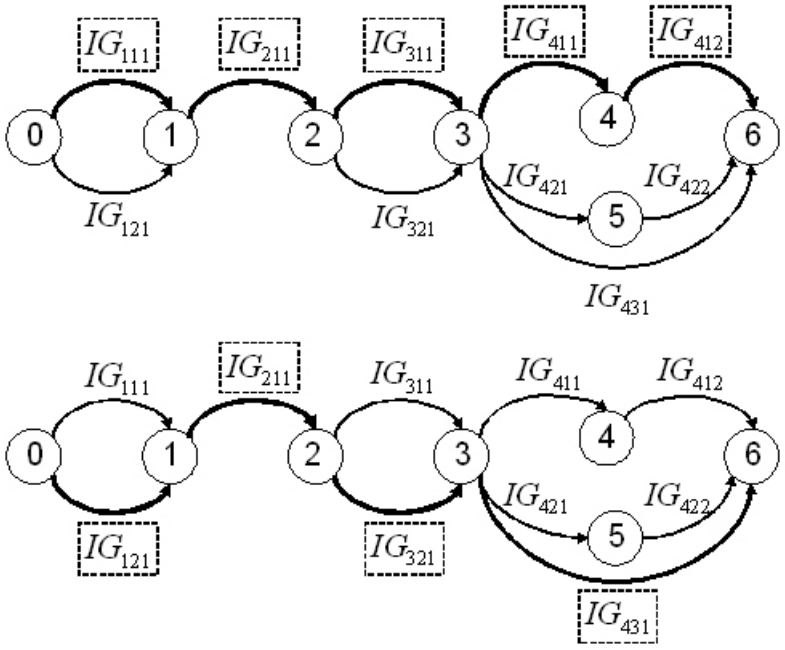


Figure 3.4: Two candidate paths in the lattice

The transfer engine ensures that a path is accepted by the parallel grammar by the following procedure. First, each IG is assigned a constituent and a lexical translation, using the relevant lexical rule. Then, the transfer engine parses this sequence of constituents by a bottom-up procedure, until it finds all parse trees of the sentence. As it is parsing the constituents, it will also transfer a corresponding tree structure on the English side. This is accomplished by applying the transfer rules consecutively.

Let us examine this process for the first IG (*ada*+*m*)in the sample lattice. Since the feature structure of this IG has a `LEX` value *ada* and `POS` value *Noun*, transfer engine searches for a lexical rule for the Turkish noun *ada*. The corresponding rule is shown below.

```
{N,4152}
N::N |: ["ada"] -> ["island"]          ;Unification constraint
(                                      ((x0 POS) =c "Noun")
;Constituent alignment                 )
(x1::y1)
```

A constituent of type `N` is created, and morphological features of the IG is copied

to this constituent. Then, the engine considers transfer rules with a source constituent of type N on the RHS. The relevant transfer rule is the following:

```
{NC,1}
NC::NC : [N] -> [N]
(
;Constituent alignment
(x1::y1)

;Analysis
(x0 = x1)

;Transfer
((y0 AGR-NUMBER) = (x0 AGR-NUMBER))

;Generation
(y0 = y1)
)
```

As a consequence, a constituent of type NC is created with features copied from previous constituent of type N. A search starts for transfer rules with a source constituent of type NC on the RHS, and the following rule is applied:

```
{NP,7}
NP::NP : [NC] -> ["my" NC]
(
;Constituent alignment
(x1::y2)

;Analysis
(x0 = x1)
((x0 DEF) <= yes)
((x0 POSS) <= yes)
((x0 TYPEP) <= n)

;Unification constraints
((x1 POSS-PERSON) =c 1)
((x1 POSS-NUMBER) =c Sg)

;Transfer
(y0 = x0)

;Generation
(y0 = y1)
)
```

Since the POSS-PERSON and POSS-NUMBER features of the NC constituent have values 1 and Sg (copied from the feature structure of IG *ada+m*), this rule unifies. The unification creates a constituent of type NP, with additional features (DEF yes), (POSS yes) and (TYPEP n). This NP can be parsed into either a SUBJ or OBJ constituent, since a subject or object of Turkish sentences may be nominative noun phrases. In either case, the final feature structure of the constituent will be as follows:

```
((SPANSTART 0)
 (SPANEND 1)
 (LEX ada)
 (POS Noun)
 (AGR-PERSON 3)
 (AGR-NUMBER Sg)
 (POSS-PERSON 1)
 (POSS-NUMBER Sg)
 (CASE Nom)
 (DEF yes)
 (POSS yes)
 (TYPEP n))
```

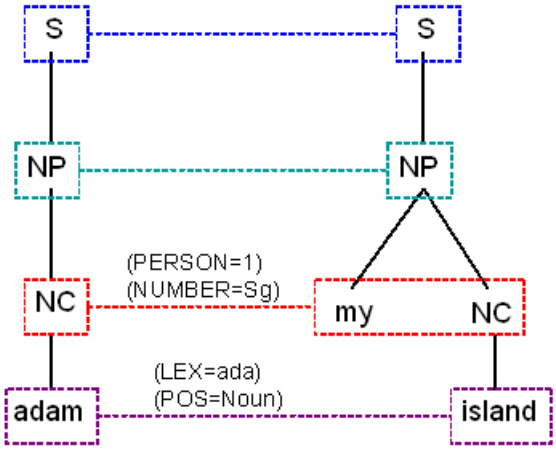


Figure 3.5: A parse tree of the IG *ada+m*

Fig. 3.5 illustrates the tree corresponding to the first IG *ada+m* parsed as `SUBJ`. In order to find a complete path, let us consider the other IGs in the sequence `IG111 IG211 IG311 IG411 IG412`. The second IG `IG211` is parsed as `N->NC->NP->Adjunct`, with a possible translation *at home*; and `IG311` is parsed as `N->NC->NP->OBJ`, with a possible translation *your son*. The remaining two IGs are a verb and a verb marker, so they should be treated together (note that the transfer engine does not know this beforehand, so it needs to search for any combination of IGs that can be parsed by the grammar). The following rule inserts the verb *be* (`y1`) in a form that agrees with the source verb marker (`y2`), and enforces the target verb (`y2`) to be in past participle form. Then, the IGs are parsed as a constituent of type `Vpass` with translation *was eaten*.

```
;;dov +ulurum --> am beaten
{Vpass,41}
Vpass::Vpass : [Vc VVpass] -> [VVpass Vc]
(
(x1::y2)
(x2::y1)

(x0 = x2)

((y1 TENSE) = (x2 TENSE))
((y1 AGR-PERSON) = (x2 AGR-PERSON))
((y1 AGR-NUMBER) = (x2 AGR-NUMBER))
((y1 POLARITY) = (x2 POLARITY))

((y2 TENSE-ASPECT-MOOD) = PastPart)

(y0 = y1)
)
```

As a result, the sequence of IGs is parsed as the sequence of constituents `[SUBJ ADJUNCT OBJ Vpass]` or `[OBJ ADJUNCT OBJ Vpass]`. There is no transfer rule with one of these sequences as its RHS, therefore they do not unify as a sentence. Hence, we say that the sequence

IG111	IG211	IG311	IG411	IG412
ada+P1Sg	ev+Loc	oğul+Acc+P2Sg	ye	+Pass+Past

is not a complete path. The linguistic reason why this parse did not form a sentence is the fact that sentences with passive verbs do not take objects in Turkish, and nominative objects should be next to the verb of a sentence. All sequences that end with `IG411-IG412` are also eliminated due to the same reason. Similarly, the sequences ending with `IG421-IG422` do not unify because a copula sentence cannot take objects.

After failing these candidates, the transfer engine tries different path candidates. For instance,

IG111	IG211	IG311	IG431
ada+P1Sg	ev+Loc	oğul+Acc+P2Sg	yen+Past

is parsed as the sequence of constituents `[SUBJ ADJUNCT OBJ Vfin]`, which unifies with the following rule:

```
;;Ben evde kediyi gordum -> I saw the cat at home
{S,10}
S::S : [SUBJ Adjunct OBJ Vfin] -> [SUBJ Vfin OBJ Adjunct]
(
;Constituent alignments
(x1::y1)
(x2::y4)
(x3::y3)
(x4::y2)

;Analysis
(x0 = x4)

;Unification constraints
((x3 CASE) =c (*or* Nom Acc))
((x1 AGR-PERSON) = (x4 AGR-PERSON))
((x1 AGR-NUMBER) = (x4 AGR-NUMBER))

;Transfer
(y0 = x0)
)
```

Since the object's case and the subject's person agrees with the verb, a constituent of type `S` is created, for which the translation is *My island beat your son at home*. Even though it does not make much sense, this sentence is a correct translation of the source sentence, and it is well-formed English. The transfer system is only concerned with an output that is a well-formed translation of the source sentence. Finding the translation that is most meaningful is the task of the language model, which is described in Section 3.4.3.

The transfer engine will continue its search until it finds all translations that the parallel grammar can produce. Another sequence of IGs is `IG121-IG211-IG311-IG431`, which is parsed similar to the previous sequence, except for the output translation *The man beat your son at home*. And the sequences `IG121-IG211-IG321-IG431` and `IG111-IG211-IG321-IG431` are translated as *The man beat his son at home* and *My island beat his son at home*, respectively. Table 3.2 summarizes the results of this translation.

Complete path				**Translation**
IG111	IG211	IG311	IG431	*My island beat your son at home*
ada+P1Sg	ev+Loc	oğul+Acc+P2Sg	yen+Past	
IG121	IG211	IG311	IG431	*The man beat your son at home*
adam	ev+Loc	oğul+Acc+P2Sg	yen+Past	
IG121	IG211	IG321	IG431	*The man beat his son at home*
adam	ev+Loc	oğul+Acc+P3Sg	yen+Past	
IG111	IG211	IG321	IG431	*My island beat his son at home*
ada+P1Sg	ev+Loc	oğul+Acc+P3Sg	yen+Past	

Table 3.2: Paths and translations of the sentence *adam evde oğlunu yendi*

3.4.3 Language Modeling

A *language model* (LM) of a language L is an estimation of the probabilistic distribution over strings in L. An LM assigns a probability to each string, representing the likelihood of the string to occur as a sentence. The model is estimated statistically from a large corpus of sentences in that language. In order to simplify this calculation, we assume that the probability of a word to occur in a context, depends only on the preceding words in that context. An N-gram model is a language model in which the probability of the occurrence of a word is assumed to depend only on the previous $N-1$ words. For more details, please refer to Section 2.3.6.

Language modeling has many applications in natural language processing, such as part-of-speech tagging, word segmentation. There are several toolkits to create language models directly from text. One such package is the SRI Language Modeling (SRILM) toolkit ([Stolcke, 2002]), a collection of programs and scripts that allows one to both create and experiment with language models. A newer language modeling technique, introduced in ([Zhang and Vogel, 2006]), is to use suffix arrays to create language models. Authors claim that a suffix array language model can deal with large amounts of data very efficiently.

The Avenue transfer system allows the user to load language models into the system. After finding all possible translations $T_1, \ldots, T_n$ of the source sentence S, the system will calculate the prior probability of each translation to occur as a sentence in the target language. Given that the system uses language model L, the "best" translation of S is determined as follows:

$$T^* = \arg \max_{T_i, i=1\ldots n} P_L(T_i) \tag{3.1}$$

The process of finding the most likely translation is called decoding. For the sample Turkish sentence *Adam evde oğlunu yendi*, all possible English translations are shown in Table 3.2. After finding these translations, the transfer engine calculates probability values (or scores) for each of these sentences using an English suffix array language model, which is created and loaded into the system beforehand.

Our language model assigns scores to the translations as in Table 3.3. According to these results, the transfer system will pick the third translation, since it has a higher probability of being observed as an English sentence. In other words, the sentence *The man beat his son at home* is more likely to be said in English, compared to the other three alternatives.

Translation	**Log Probability**
My island beat your son at home	-29.5973
The man beat your son at home	-27.1953
The man beat his son at home	-23.7629
My island beat his son at home	-26.1649

Table 3.3: LM scores of translations of the sentence *adam evde oğlunu yendi*

Actually, the transfer engine has a complex way of handling the decoding step. It tends to select complete paths which correspond to complete translations, however it may consider partial translations and combine them together to form a translation. So, besides the four complete translations in Table 3.2, Avenue may examine partial translations such as

- *adam* $\Rightarrow$ *the man*
- *evde* $\Rightarrow$ *at home*
- *oğlunu yendi* $\Rightarrow$ *he beat his son*

and combine them into a translation

the man at home he beat his son

As any English speaker can understand that this is nonsensical English, the language model also assigns a very low score to this sentence. The multiplier that lowers the score is $P(beat|home, he)$, which has a probability of 4.23×10^{-6}. The total score of this sentence is -28.1472, and extra parameters are added by Avenue to penalize partial translations. Technical details of Avenue transfer system are not discussed in

this thesis, but a complete translation is always preferred to a partial one if it does not have a very low LM score.

In sum, our system works in three stages. First, the Turkish sentence is morphologically analyzed and represented as a lattice. Avenue transfer engine parses this lattice, transfers its structure to English, and generates possible English translations, using a set of manually-crafted transfer rules. Finally, an English language model is used to pick the translation that looks "best", based on statistical calculations. Partial translations are also scored by the LM, but Avenue prefers complete translations when available.

3.5 Linguistic Coverage and Examples

The system we describe covers the translation of most of the noun phrase structures in Turkish. Sentences can be translated relatively easier, when a wide coverage of noun phrases is accomplished. In this section, we first examine the noun phrases that can be covered by our system; sentences are described later. For a complete list of the transfer rules, please see the Appendix.

3.5.1 Noun Phrases

The case and possession information appears as suffixes in Turkish nouns. For instance, *kitaplarım* is the first person singular possessive, plural version of *kitap* (*book*), thus should be translated as *my books*. This is handled by passing the plural marker, and adding a constant (*my*, *your*, *his*, ...) to the English side. The following describes one of these rules:

```
{NP,7}
NP::NP : [NC] -> ["your" NC]
(
(x1::y2)

(x0 = x1)
((x0 def) <= yes)
((x0 poss) <= yes)
((x0 typep) <= n)
((x1 POSS-PERSON) =c 2)
((x1 POSS-NUMBER) =c Sg)

((y0 POSS-PERSON) = (x0 POSS-PERSON))
((y0 POSS-NUMBER) = (x0 POSS-NUMBER))

(y0 = y2)
)
```

Notice how the POSS-PERSON and POSS-NUMBER features are transferred by transfer equations, and the constant *my* is included at the target side. Similar rules can be written for *your*, *his*, *her*, *its*, *our*, and *their*.

Noun phrases have features def, poss, and typep, that indicate various properties. def has two possible values, yes or no, corresponding to definite and indefinite nouns. For instance, the word *kitap* can be translated as *book* or *the book*, which are distinguished by the value of def; the value is no for the first translation and yes for the second translation. POSS has the same two possible values, but has a more complicated meaning. The word *kitabım* should be translated as *my book*, but sometimes it is interpreted as *book* at intermediate steps. An example is adjective phrases such as *mavi kitabım*, of which the translation is *my blue book*. This is accomplished by the following rule:

```
;;mavi kitabim -> my blue book
{NPAdj,4}                                    ((x2 POSS-PERSON) =c 1)
NPAdj::NPAdj : [AP NP] -> ["my" AP NP]       ((x2 POSS-NUMBER) =c Sg)
(                                            ((x2 typep) =c (*or* n nn2))
(x1::y2)                                     ((x2 def) =c (*not* yes))
(x2::y3)                                     ((x1 typep) = (*not* num))

(x0 = x2)                                    (y0 = x0)
((x0 def) <= yes)
((x0 poss) <= yes)                           (y0 = y3)
((x0 typep) <= an)
```

Notice that the word *kitabım* should be first interpreted as *book*, and then *my* will be included before *mavi* in this rule. For this reason, possessive nouns such as *kitabım* are translated as both *book* and *my book*, with a poss value no and yes, respectively. poss has value no if a noun is possessive and it has not been translated appropriately, it is yes otherwise. The third feature typep indicates the type of a noun phrase, which can take a value n for nouns, an for adjective-noun phrases, and so on. See Appendix for a complete list of features and possible values. Let us now examine different types of noun phrases by relevant rules and examples.

Noun-Noun phrases

In Turkish, two consecutive noun phrases may combine into a larger noun phrase. More specifically, if a non-possessive genitive or nominative noun phrase precedes another

noun phrase with third person singular possession, they form a noun-noun phrase. For instance, *kitabımın kapağı* can be translated as *cover of my book*, *the cover of my book*, or *my book's cover*. Here, *kitabım* corresponds to *my book*, and *kapağı* corresponds to *cover* or *the cover*. There are two ways to translate these noun phrases into English; either the order of the noun phrases are reversed and an *of* is placed in the middle, or the order remains same and an *'s* is placed in the middle. The following is a rule for the latter transformation.

```
;;kitabin kapagi -> book 's cover
{NPnn,1}
NPnn::NPnn : [NP NP] -> [NP "'s" NP]
(
(x1::y1)
(x2::y3)                                    ((x2 POSS-PERSON) =c 3)
                                            ((x2 POSS-NUMBER) =c Sg)

;Analysis
(x0 = x2)                                   ((x2 def) =c no)
                                            ((x1 poss) =c yes)

((x0 def) <= (x1 def))
((x0 poss) <= yes)                          (y0 = x0)
((x0 typep) <= nn)                          )


;Constraints
((x1 CASE) =c Gen)
((x1 POSS-PERSON) =c None)
((x1 POSS-NUMBER) =c None)
```

In the first rule, the first five constraints describe the condition we mention above: first noun is non-possessive and genitive, and second noun is third person singular possessive. The next two constraints indicate the allowed types of noun phrases to form a noun-noun phrase. The following equation (`((x2 def) =c no)`) ensures that the second NP is translated in indefinite form, which prevents *kitabımın kapağı* to be translated as *my book's the cover*. Finally, the last constraint (`((x1 poss) =c yes)`) ensures that the first NP is translated with a possessive marker, which prevents the phrase to be translated as *book's cover*.

The analysis equation (`((x0 def) <= (x1 def))`) marks the noun-noun phrase as definite if the first NP is definite, and indefinite otherwise. The next two equations say that the two features `poss` and `typep` of the noun-noun phrase is `yes` and `nn`,

respectively.

Another kind of noun-noun phrase is a nominative noun phrase preceding a noun phrase with some possessive marker. For instance, *kitap kapağı* can be translated as *book cover*, *the book cover* or *his/her/its book cover*. In order to implement this structural transformation, we add a constant (either *the* or one of the possessive markers) at the target side before the two noun phrases. A sample rule is shown below.

```
;;kitap kapagi -> his book cover
{NPnn,16}
NPnn::NPnn : [NP NP] -> ["his" NP NP]
(
(x1::y2)
(x2::y3)

;Analysis
(x0 = x2)

((x0 def) <= yes)
((x0 poss) <= yes)
((x0 typep) <= nn2)

;Constraints
((x1 CASE) =c Nom)
((x1 POSS-PERSON) =c None)
((x1 POSS-NUMBER) =c None)
((x2 POSS-PERSON) =c 3)
((x2 POSS-NUMBER) =c Sg)

((x1 typep) =c (*or* n nn2))
((x2 typep) =c n)

((x1 def) =c no)
((x2 def) =c no)

(y0 = x0)
)
```

Other cases are handled by similar rules. Table 3.4 summarizes some sample noun-noun phrases that can be translated by our system.

Adjective-Noun phrases

An adjective phrase followed by a noun phrase is an adjective-noun phrase, where an adjective phrase can be

- a single adjective (e.g. *mavi*⇒*blue*),
- consecutive adjectives (e.g. *büyük mavi*⇒*big blue*),
- an adjective preceded by an adverb (e.g. *çok büyük*⇒*very big*), or

Noun-Noun phrase	**Translations**
kitaplarımın kapakları	*the covers of my books* *my books' covers*
kitap kapağım	*my book cover*
alarm sistemi	*alarm system* *the alarm system* *his alarm system* *her alarm system* *its alarm system*
tarih dersi kitabı kapağı	*the book cover of the history class*

Table 3.4: Sample noun-noun phrase translations

- a non-adjective phrase with a suffix that turns its part-of-speech to adjective (e.g. *evimdeki⇒(the one) in my house*).

As you may notice, the first three cases are identical in Turkish and English. We call adjective phrases corresponding to the last case *posterior* adjective phrases, because they are appended to the end of the noun phrase in English. For instance, *evimdeki kedi* is translated as *the cat in my house* in English, where *evimdeki* corresponds to *in my house* and *kedi* corresponds to *the cat*. We separate normal adjective phrases from posterior ones by the two constituent names `AP` and `APost`.

When translating adjective-noun phrases, a constant is included at the English side according to the possessive properties of the noun phrase. For example, *büyük mavi kitabım* is translated as *my big blue book*. As our system can handle adjective-noun phrases and noun-noun phrases, these can be combined to translate more complicated noun phrases such as *büyük mavi kitabımın kapağı* (Eng. *my big blue book's cover* or *the cover of my big blue book*). Sample adjective-noun phrase translations are shown in Table 3.5.

Adjective-Noun phrase	**Translations**
mavi kitabım	*my blue book*
mavi kitap kapağım	*my blue book cover*
mavi evime ait büyük kitap kapakları	*big book covers belonging to my blue house* *their big blue covers belonging to my blue house*

Table 3.5: Sample adjective-noun phrase translations

Phrases including determiners

Some noun phrases include determiners, and this should be handled separately during translation. In Turkish, determiners like *bu*, *şu*, *o* precede noun phrases just like in English. However, the same determiner is used for both singular and plural noun phrases in Turkish; for instance, *bu mavi kitap* means *this blue book*, while *bu mavi kitaplar* means *these blue books*. The determiner *bu* is translated as *this* or *these* according to the plurality of the noun phrase it precedes. In our system, this is accomplished by translating *bu* as *this* and *these*, then checking for agreement of the determiner and noun phrase:

```
;bu kitap -> this book
{NPDet,1}
NPDet::NPDet : [ADet NP] -> [ADet NP]  ((x2 def) =c no)
(                                      ((x2 typep) =c (*not* nn arel an))
(x1::y1)                               ((x2 POSS-PERSON) =c (*or* None 3))
(x2::y2)                               ((x2 POSS-NUMBER) =c (*or* None Sg))
                                       ((x1 agr) =c (x2 agr))
(x0 = x2)
                                       (y0 = x0)
((x0 def) <= yes)                      )
((x0 typep) <= adet)
((x0 of) <= no)
```

In the above rule, `ADet` is a constituent for adjective phrases containing a determiner. For noun phrases with possessive nouns, a constant should be added to the target side. For example, *bu kitap kapağım* is translated as *this book cover of mine*. Similar rules are written to handle these cases as well.

The determiner *bir* should be treated separately, because it is translated differently and it may also be interpreted as the number 1. The following two examples illustrate the two behaviors of *bir*:

mavi bir kitap ⇒ *a blue book*
bir mavi kitap ⇒ *one blue book* or *a blue book*

Notice that the determiner *bir* does not precede adjectives, but instead comes after adjectives and right before the noun of a Turkish noun phrase. However, the English translation, which is the determiner *a*, precedes adjectives like other determiners in English. In order to cover this difference in our system, we separate determiners into

two groups: `Det1`, and `Det2`. `Det1` refers to regular determiners explained above, and `Det2` refers to *bir* in the following rules:

```
;;[bu iyi] insan -> [this nice] person
{ADet,1}
ADet::ADet : [Det1 AP] -> [Det1 AP]
(
(x1::y1)
(x2::y2)

(x0 = x1)

(y0 = x0)
)
```

```
;;[iyi bir] insan -> [a nice] person
{ADet,2}
ADet::ADet : [AP Det2] -> [Det2 AP]
(
(x1::y2)
(x2::y1)

(x0 = x1)

(y0 = x0)
)
```

The distinction between *a* and *an* is not handled by the grammar, since the language model can easily eliminate the incorrect one.

Phrases with Prepositions/Postpositions

Noun phrases followed by postpositions is the usual case in Turkish, while preposition phrases serve this task in English. For instance, *arkadaşım için* consists of a noun phrase (*arkadaşım*) followed by a postposition (*için*). The equivalent phrase in English is *for my friend*, where *için* and *arkadaşım* correspond to *for* and *my friend*, respectively. Therefore, we first translate the pre/postposition and noun phrase, then reverse their order. A noun phrase followed by a postposition is parsed as an adverb phrase:

```
;;adamla birlikte -> with the man
{AdvP,1}
AdvP::AdvP : [NP Postp] -> [Postp NP]
(
(x1::y2)
(x2::y1)

(x0 = x1)
((x2 SUBCAT) =c (x1 CASE))
((x1 poss) =c yes)

((x0 typep) <= postp)

(y0 = x0)
)
```

The `SUBCAT` feature of the postposition constituent indicates the case of the noun phrase that precedes it. There are noun phrases where the second noun acts like a postposition, so some rules are written to cover these phrases. For instance, *arkadaşım*

means *my friend* and *yüzünden* means *from your/his/her face*. Following a noun phrase, *yüzünden* may be translated as *because of*. For example, *arkadaşım yüzünden* can be translated as *from my friend's face* or *because of my friend*. We write a transfer rule that translates this phrase as the latter:

```
;;arkadasim yuzunden -> because of my friend
{NP,10}
NP::NP : [NP NP] -> ["because of" NP]
(
(x0 = x1)

((x2 lex) =c "yz")
((x2 CASE) =c Abl)
((x2 POSS-PERSON) =c (x1 AGR-PERSON))
((x2 POSS-NUMBER) =c (x1 AGR-NUMBER))
                                        ((x2 typep) =c n)
                                        ((x1 CASE) =c Nom)
                                        ((x1 poss) =c yes)

                                        (y0 = x0)
                                        )
```

Pronoun-Noun phrases

Pronouns and pronoun-noun phrases are covered by our grammar. Pronouns are parsed and translated by lexical rules, where the first person singular pronoun is handled carefully; the nominative case (*ben*) is translated as *I*, accusative, dative, ablative, and locative cases (beni, bana, benden, bende) are translated as *me*, and the genitive case (*benim*) is translated as *mine*.

Genitive pronouns followed by a noun phrase form a pronoun-noun phrase, where the possession of the noun phrase should agree with the pronoun. For instance, *benim kedim* means *my cat* where *benim* is the genitive case of first person singular pronoun *ben* and *kedim* is the first person singular possessive version of noun *kedi*. The following rule describes this translation.

```
;;senin kitabin -> your book
NPpron::NPpron : [Pron NP] -> [Pron NP]
(
(x1::y1)
(x2::y2)

;Analysis
(x0 = x2)

((x0 def) <= yes)
((x0 typep) <= nn)
```

```
;Constraints
((x1 CASE) =c Gen)
((x2 POSS-PERSON) =c (x1 POSS-PERSON))
((x2 POSS-NUMBER) =c (x1 POSS-NUMBER))
((x2 def) =c no)

;Transfer
(y0 = x0)
)
```

Another issue is the translation of the pronoun *kendi*, which takes a suffix that determines its possession, and therefore its translation. For example, *kendim* is parsed as the pronoun *kendi*, followed by a morpheme that marks it as first person singular. Thus, it is translated as *myself*. This is covered by separate rules for each case, illustrated in rules below.

```
{Pron,12}
;;kendim -> myself
Pron::Pron : [Pron2 VV] -> ["myself"]
(
(x0 = x2)

((x0 lex) <= (x1 lex))
((x0 pos) <= (x1 pos))
((x0 CASE) <= Nom)
((x0 def) <= yes)
```

```
((x2 lex) =c Reflexive)
((x2 POSS-PERSON) =c 1)
((x2 POSS-NUMBER) =c Sg)

(y0 = x0)
)
```

In its plain form, *kendi* can also occur in pronoun-noun phrases with a meaning *own*. For instance, *kendi kitabım* means *my own book* because the noun *kitabım* is first person singular, and other cases are translated similarly. This is also covered by the grammar.

Conjunctions

Our system covers two different types of conjunctions:

- Conjunctives that conjoin two noun phrases as in *kediyi ve köpeği* (*the cat and dog*)

- Conjunctives that follow a noun phrase as in *kediler falan* (*cats or so*)

For the first type of conjunction, the case of the two noun phrases should agree or the first noun phrase should be nominative. The following two rules handle this type of conjunctions, and the second type is covered similarly.

```
;;kediyi ve kopekleri -> the cat and dogs
{NPconj,1}
NPconj::NPconj : [NP CONJ1 NP] -> [NP CONJ1 NP]
(                                        ((x1 CASE) =c (x3 CASE))
(x1::y1)                                 ((x2 lex) =c (*not* "ki"))
(x2::y2)                                 ((x1 poss) =c yes)
(x3::y3)

                                         (y0 = x0)
(x0 = x3)                                )


((x0 CASE) <= (x3 CASE))
((x0 poss) <= yes)
```

```
;;kedi ve kopegi -> the cat and dog
{NPconj,2}
NPconj::NPconj : [NP CONJ1 NP] -> [NP CONJ1 NP]
(
(x1::y1)                                 ((x1 CASE) =c Nom)
(x2::y2)                                 ((x3 CASE) =c (*not* Nom))
(x3::y3)                                 ((x1 poss) =c yes)

(x0 = x3)                                (y0 = x0)
                                         )
((x0 CASE) <= (x3 CASE))
((x0 poss) <= yes)
```

3.5.2 Sentences

As mentioned before, translating sentences is relatively easy when noun phrases are parsed and translated. First, we describe consituents that form a sentence.

- `SUBJ` : Subject of a sentece is either a nominative or genitive noun phrase.
- `OBJ` : Object of a sentence is either a nominative or accusative noun phrase.

- `OBJ-THETA` : Alternative object of a sentence is either an ablative, locative or dative noun phrase.
- `ADJUNCT` : Adjunct of a sentence is either an ablative, locative or dative noun phrase.
- `AdvP` : Adverb phrase of a sentence is either an adverb, a noun phrase followed by a postposition or an incomplete sentence acting like an adverb.
- `Vfin` : Final verb of a sentence is a verb with tense, agreement, and optionally passive or causative features.
- `Vcop` : Verb of a copula sentence is actually a noun phrase or adjective phrase that behaves like a verb.
- `Vbe` : This constituent is for the special word *var* in Turkish, that is translated as either *be* or *have* in English.

Each sentence is a combination of these constituents, where the only obligation is the presence of one of the verb forms (`Vfin`, `Vcop`, or `Vbe`). Turkish sentences have SOV order, but in practice constituents may be placed freely. Since Avenue transfer formalism does not support constructs to indicate optional constituents or free order, we write separate rules for each permutation or lack of a constituent.

At the English side, the subject is always at the very beginning of the sentence. The subject is followed by a verb form, which is followed by the object or alternative object (intransitive verbs do not take any object). The remaining constituents (a set of adjuncts and adverb phrases) come after the object, and may change order freely. Below is a rule for translation of a typical sentence.

```
;;Ben evi gordum -> I saw the house
{Stemp,100}
Stemp::Stemp : [SUBJ OBJ Vfin] -> [SUBJ Vfin OBJ]
(                                          ((x2 CASE) =c (x3 CASEV))
(x1::y1)
(x2::y3)                                   ((x1 AGR-PERSON) = (x3 AGR-PERSON))
(x3::y2)                                   ((x1 AGR-NUMBER) = (x3 AGR-NUMBER))

(x0 = x3)                                  (y0 = x0)
                                           )
```

Notice that the case of the object and the person and number features of the subject agree with the verb's case, person, and number features. Each verb agrees with objects of a specific case, determined by the feature CASEV. The agreement between the subject and verb prevents the system from accepting sentences like *Adam kediyi gördüm* or *Ben kediyi gördün*.

Using this rule, the system translates *Annem yemeği yedi* as *My mother ate the food*. In Turkish, passive voice of verbs is indicated by a passive marker following the verb. In this case, the subject is nominative, the object is missing, and verbs are translated accordingly. The following rule describes this translation.

```
;;Evler yandi -> The houses were burnt
{Stemp,121}
Stemp::Stemp : [SUBJ Vpass] -> [SUBJ Vpass]
(                                        ((x1 AGR-PERSON) = (x2 AGR-PERSON))
(x1::y1)                                 ((x1 AGR-NUMBER) = (x2 AGR-NUMBER))
(x2::y2)

                                         (y0 = x0)
(x0 = x2)                                )
```

Yemek yendi is translated as *The food was eaten* by this rule. Causative verbs are also indicated by markers in Turkish, and require two objects; one of the objects are caused to perform an action, and the other one is the object of that action. For instance, *Annem yemeği çocuğa yedirdi* is translated as *My mother made/caused the kid eat the food* by the following transfer rule in our grammar.

```
;;Annem yemegi cocuga ye +dirdi -> My mother made the kid eat the food
{Stemp,110}
Stemp::Stemp : [SUBJ OBJ OBJTH V VVfin] -> [SUBJ VVfin OBJTH V OBJ]
(
(x1::y1)                                 ((x4 trans) =c yes)
(x2::y3)                                 ((x2 CASE) = (x4 casev))
(x3::y5)                                 ((x1 AGR-PERSON) = (x5 AGR-PERSON))
(x4::y4)                                 ((x1 AGR-NUMBER) = (x5 AGR-NUMBER))
(x5::y2)

                                         (y0 = x0)
(x0 = x5)                                )
```

If the causative verb is intransitive, then there is only one object; an example is the sentence *Annem çocuğu uyuttu*, which is translated as *My mother caused/made the*

kid sleep.

```
;;Annem cocugu uyuttu -> My mother made the kid sleep
{Stemp,108}
Stemp::Stemp : [SUBJ OBJ V VVfin] -> [SUBJ VVfin OBJ V]
(                                          ((x3 trans) =c no)
(x1::y1)                                   ((x2 CASE) =c (x4 casev))
(x2::y3)                                   ((x1 AGR-PERSON) = (x4 AGR-PERSON))
(x3::y4)                                   ((x1 AGR-NUMBER) = (x4 AGR-NUMBER))
(x4::y2)

                                           (y0 = x0)
(x0 = x4)                                  )
```

Adjuncts (`Adjunct`) indicate time and location, and adverb phrases (`AdvP`) indicate the reason and manner in the sentence. For instance, in the example

Annem yemeğ̆i evde hızlıca yedi (Eng. *My mother ate the food fastly at home*)

the adjunct *evde* (Eng. *at home*) indicates the location and the adverb phrase *hızlıca* (Eng. *fastly*) indicates the manner of the eating.

Subjects are always nominative in complete sentences, but genitive in other incomplete forms of a sentence. An *incomplete sentence* is a sentence with a verb that does not have a tense, but instead a suffix that changes its part-of-speech into noun, adjective or adverb. These sentences are parsed as either noun phrases, posterior adjective phrases, or adverb phrases.

The sentence *annemin yemeğ̆i yediğ̆ini* is a noun phrase, which is the incomplete version of the sentence *annem yemeğ̆i yedi*. Notice that the subject of the incomplete sentence is in genitive case, and the last word is a verb followed by a verb-to-noun suffix. One of the translations found by our system for this form is *that my mother ate the food*.

Another incomplete version of the same sentence is *yemeğ̆i yiyen*. This sentence does not have a subject, and the verb has a verb-to-adjective suffix, which is translated as *that ate the food*. Another example *annem yemeğ̆i yerken* is parsed as an adverb phrase because the verb has a verb-to-adverb suffix. Our system translates this form as *while my mother eats/ate the food*.

The transfer rules that cover these sentential forms are in the Appendix. Now, let us illustrate how our system handles these issues in a complex sentence. Consider

the following sentence:

Annemin yemeği ona yolculukları sevdiren trende kitap okurken yediği söylendi.

Fig. 3.6 shows the parse of this passive voiced sentence, with a differerent color and underlining for each IG. There are three incomplete sentence structures in this sentence, and the figure illustrates how they are parsed and translated. For instance, the word sequence indicated by number 2 is an incomplete sentence with a causative verb, which takes a verb-to-adjective suffix to change the part-of-speech to adjective. This adjective precedes the noun in the adjunct of incomplete sentence number 1. Similarly, the noun phrase formed by the first incomplete sentence serves as the `OBJ` of the complete sentence, and the adverb phrase formed by the third incomplete sentence is an `AdvP` in incomplete sentence number 1.

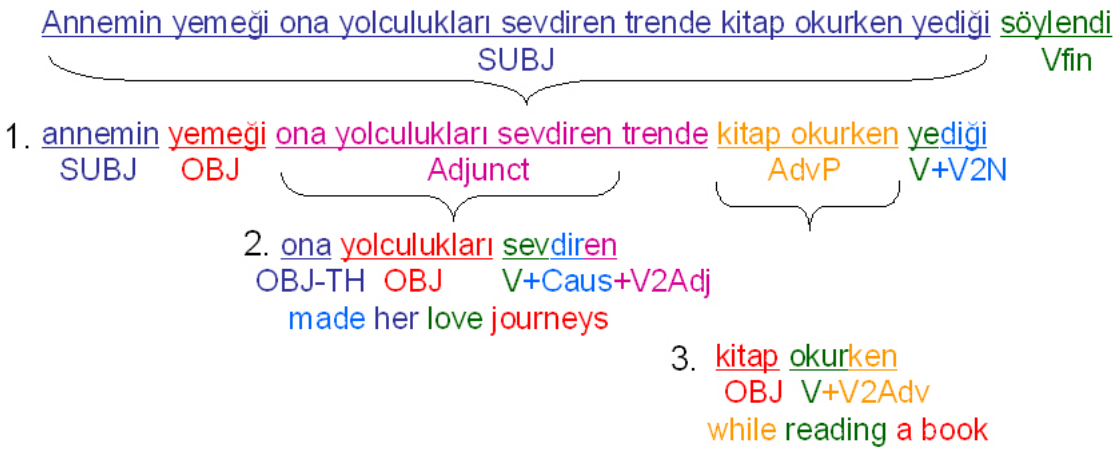


Figure 3.6: Parse and translation of a sample sentence

The translations of 2 and 3 are shown below the sentence, while 1 is translated as *that my mother ate the food while reading a book on the train that made her love journeys*. With these partial translations, our system determines the translation of the complete sentence as

It was said that my mother ate the food while reading a book on the train that made her love journeys.

Chapter 4

Evaluation

4.1 MT Evaluation

Machine Translation systems produce a massive amount of translation sentences as output, each of which should be evaluated as a good or bad translation. MT evaluation is the task of evaluating output translations of machine translation systems, such that the score assigned to a translation coincides with human evaluation. Evaluating MT systems is important because it provides feedback for researchers, a proof of success/failure of a system, and a measure for comparison of alternative systems. Moreover, evaluating MT systems automatically is important because human evaluation is very slow and costly. A computer can evaluate a text containing thousands of sentences in seconds at no cost, while this will probably take weeks and much money by human. In other words, the need of machine power in translation is also valid for evaluation.

There are two dimensions of a good translation: quality and fidelity. Quality stands for a syntactically well-formed output, and fidelity stands for an output that has the same meaning as the input ([Hovy et al., 2002]). In this section, we present three existing evaluation metrics, BLEU, METEOR, and WER. Each of these systems calculate the distance between the system's translation output and a reference translation, by taking into account these two dimensions. Based on this distance, they determine how close the system translation is to the reference translations, and assign a score to each translation.

4.1.1 WER (Word Error Rate)

The most classical approach to MT evaluation is Word Error Rate (WER) ([McCowan et al., 2004]) It is based on the Levenshtein distance, which is the number of insertions, deletions,

and substitutions required to transform the reference translation into the system translation. Let N be the number of words in the reference translation, S be the number of substituted words in the system translation, D be the number of deleted words in the system translation, and I be the number of inserted words in the system translation. Then, the WER is calculated as below.

$$WER = \frac{S + I + D}{N} \tag{4.1}$$

One disadvantage of this metric is the difficulty in interpretation. Since the sum of S, I, and D is not bounded by N, the score can be greater than 1. In these cases, it is difficult to interpret and compare results ([McCowan et al., 2004]).

4.1.2 BLEU (Bilingual Evaluation Understudy)

Bilingual Evaluation Understudy (BLEU) is a metric based on counting the number of common n-grams of words between the system translation S and a set of reference translations $R_1, ..., R_m$ ([Papineni et al., 2001]). First, it counts the number of occurrences of each n-gram c in S, say S_c. Then, it finds the maximum number of occurrences of c in reference translations, say R_c. The minimum of these two gives the number of shared occurrences of c. For each c, the number of shared occurrences are summed up, and this sum is divided by the number of n-grams in the system translation. This is called the precision of s for n-grams, P_n.

$$\begin{aligned} N_c &= \text{count of } c \text{ in } S \\ R_c &= \max_{i=1..m} \{\text{count of } c \text{ in } R_i\} \\ S_c &= \min\{N_c, R_c\} \\ P_n &= \frac{\sum_{\text{n-gram } c \in S} \min(N_c, R_c)}{\sum_{\text{n-gram } c' \in S} N'_c} \end{aligned} \tag{4.2}$$

A weighted geometric average of the P_n's is used to find the precision of a translation. In addition, a *brevity penalty* factor is applied in order to penalize short translations. Let r be the length of the system translation, and c be the length of the reference translation with closest length to r. Then, the brevity penalty is calculated as the following.

$$BP = \begin{cases} 1 & if\ c > r \\ \exp^{1-\frac{r}{c}} & if\ c \leq r \end{cases} \tag{4.3}$$

Then, the BLEU score of a translation is given by

$$BLEU = BP \times \exp(\sum_{n=1}^{N} w_n log p_n) \tag{4.4}$$

BLEU is a commonly used metric for MT evaluation; it assigns translation scores that are highly correlated to human evaluation efficiently ([Papineni et al., 2001]). On the other hand, BLEU only considers the *precision* of n-grams of a translation (i.e., number of n-grams in the system translation that also occur in reference translations) but not the *recall* (i.e., number of n-grams in a reference translation that also occur in the system translation). The reason is that BLEU considers a set of references at the same time, and cannot define the term recall where each reference translation uses different words. This drawback is mentioned by Papineni *et al.* ([Papineni et al., 2001]), Satanjeev and Lavie ([Banerjee and Lavie, 2005]). Satanjeev and Lavie introduce an alternative metric that can overcome this disadvantage of BLEU, besides additional improvements ([Banerjee and Lavie, 2005]).

4.1.3 METEOR

METEOR is an automatic metric for MT evaluation, designed to improve weaknesses of BLEU ([Banerjee and Lavie, 2005]). The authors claim that BLEU does not take into account the recall factor, and the geometric average does not make sense when one of the precision values is zero. METEOR is based on a word-to-word alignment of the system translation to each reference translation separately. An *alignment* between two sentences is defined as a set of mappings, where each word in a sentence is mapped to at most one word of the other sentence. A word can *map* to another word if they are exactly same, their roots are exactly same, or they are synonyms. A *cross* is basically two lines crossing when the sentences are typed out in two rows and a line is drawn for each word mapping.

The alignment that contains maximum number of mappings is chosen; if there are two such alignments, the one with least number of crosses is preferred. For this alignment, two measures are calculated: precision P and recall R. Let S_1 be the number of words in the system translation that are mapped to one word in the reference translation, S_2 be the number of words in the reference translation that are mapped to one word in the system translation, C_1 be the number of words in the system translation, and C_2 be the number of words in the reference translation.

$$P = \frac{S_1}{C_1} \tag{4.5}$$

$$R = \frac{S_2}{C_2} \tag{4.6}$$

Based on these values, a harmonic average with most weight on the recall is calculated as following:

$$Fmean = \frac{10PR}{R + 9P} \tag{4.7}$$

In order to favor longer translations, a penalty factor is introduced. A *chunk* is a sequence of adjacent words, which is mapped to a sequence of adjacent words. After finding the largest chunks in a translation, the penalty can be calculated as follows:

$$Penalty = 0.5 \times \frac{\text{number of chunks}}{\text{number of matched words}} \tag{4.8}$$

The final score of a translation is then computed by the formula

$$Score = Fmean(1 - Penalty) \tag{4.9}$$

Experiments show that the scores assigned by METEOR metric are closer to human evaluation than BLEU ([Banerjee and Lavie, 2005]).

4.2 Test Results

We experimented our system on a set of Turkish noun phrases and sentences. We first translated phrases and sentences with our MT system, then compared results to a set of reference translations.

Noun phrases are translated very accurately, with a BLEU score of 60.38 for a set of 192 noun phrases. The structural transformation of noun phrases is almost perfectly accurate, but the choice of lexical translations reduces the overall BLEU score. Therefore, this score would tend to be greater when evaluated with METEOR, since it takes synonyms into account. Some sample noun phrases and their translations are given below, where a reference translation is written beneath each system translation.

Noun phrase	**System and reference translations**
siyahlarla birlikte bir protesto yürüyüşünde	*in a protest walk with the blacks*
	in a protest walk with the blacks
Elif'in arkasındaki kapıda	*at the door at the back of Elif*
	on the door behind Elif
alışveriş dünyasında	*in the shopping world*
	at the shopping world

Promising results are obtained for sentence translations, even though there are improvements to be made. For 90 sentences with translations less than 15 words, the BLEU score is 27.99. The system does not hault in a meaningful time for longer sentences. Below, there are some sample Turkish sentences and corresponding translations found by our system.

Sentence:	*Kaçtıkça daha büyüdü, bir tutku oldu*
Translation:	*It grew more as escaping, it became a passion*
Sentence:	*Bu tutku zamanla bana acı vermeye başladı*
Translation:	*This passion began to give pain to me with time*
Sentence:	*Perşembe uzun yürüyüşler ve ziyaretler yapıyorum*
Translation:	*I am doing long walks and visits on Thursday*
Sentence:	*Kentin Müslümanların eline geçme olasılığı var*
Translation:	*There is the possibility that the city will be taken over my Muslims*

As a result, the system is quite successful at translating noun phrases and sentence components. It can also parse and translate sentences, although translating sentences becomes computationally challenging as the sentence gets longer.

Chapter 5

Summary and Conclusion

In this thesis, we have introduced a machine translation system from Turkish to English. Our approach is a combination of classical transfer-based approach and statistical approaches. This novel approach allows us to exploit morphological information of Turkish sentences, represent structural differences between Turkish and English by manually written transfer rules, and apply statistical techniques to find the best translation.

The system we present is the first Turkish to English MT system based on a hybrid approach. We implemented a set of transfer rules that describes a mapping from Turkish grammar structure to English grammar structure. We integrated an existing morphological analyzer into the Avenue transfer system, that applies the transfer rules and an English language model to translate Turkish text into English.

There are two advantages of this hybrid approach, that makes it superior to a straightforward transfer-based or statistical approach.

- As a result of the manually crafted transfer rules, our system finds more reliable results when compared to statistical approaches. In other words, the system is structurally sound (i.e., a translation output by the system is structurally a correct translation of the input), but not complete yet. Although the linguistic coverage is wide enough to translate regular sentences, further coverage will locate it nearer to a complete system.

- As a result of statistical techniques, our system handles the ambiguity that is a handicap for most transfer-based approaches. The English language model disambiguates by choosing the translation that "looks most sound", based on statistics extracted by a large corpus of English sentences.

A disadvantage of our system is its computational efficiency, since long sentences may take time to translate. Another drawback is due to the nature of manually written

rules. It is a time-consuming effort for future researchers to first learn the rule formalism and then improve it for wider coverage.

Future research may include further improvement of the transfer rules in terms of computational efficiency and linguistic coverage. Another idea is to learn transfer rules automatically from a Turkish-English parallel corpus. This may be accomplished if a sufficiently large parallel corpus becomes available.

Chapter A

Appendix

There are 43 constituents, here are the count of rules for each:

Constituent	C	Explanation	Example
NP	28	Noun phrase	kitabın kapağı → the book's cover
Vc	15	Verb with no tense	-
Vfin	50	Simple verb	veriyor → am/is/are giving
N	10	Noun	
Vn	18	Noun created from a verb	
A	14	Adjective	
Va	6	Adjective created from a verb	
APost	16	Posterior adjective	evdeki kedi → the cat at the house
Vcop	44	Verb created from noun or adjective	adam kördü → the man was blind
Vpass	38	Passive verb	verildi → was/were given
VVc	2	Causative verb marker	-
VVpass	2	Passive verb marker	-
VVable	2	Able verb marker	-
VVfin	36	Causative verb	yedirdi → caused (someone) to eat
VVcp	28	Causative and passive verb	yedirildi → was caused to eat
NPAdj	12	Adjective-Noun phrase	mavi kitap → blue book
NPconj	3	Conjunctive Noun phrase	kedi ve köpek → the cat and dog
NPDet	8	Determinant-Noun phrase	bu kitap → this book
PronP	2	Pronoun phrase	bana → me
NPnn	26	Noun-noun phrase	kitabın kapağı → the book's cover
NC	1	intermediate form of a noun phrase	-
S	3	sentence	-
Sconj	1	Conjunctive sentence	
Scop	3	Copula sentence	
Stemp	18	intermediate form of a sentence	-
SUBJ	2	Subject	-
OBJ	2	Object	-
OBJTH	3	Alternative object	-
Adjunct	8	Adjunct	-
AdvP	4	Adverb phrase	yavaşça → slowly
AP	6	Adjective phrase	kötü bir → a bad
APsimp	3	Simple adjective phrase	büyük mavi → big blue
ADet	7	Determinant adjective phrase	bu kötü → this bad
Det1	2	Determinant	bu → this
Postp	11	Postposition	önce → before

Table A.1: Explanation and rule count of constituents

Bibliography

[Appelo et al., 1987] Appelo, L., Fellinger, C., and Landsbergen, J. (1987). Subgrammars, rule classes and control in the rosetta translation system. In *Proceedings of the Third conference on European chapter of the Association for Computational Linguistics*, pages 118–133.

[Ayan et al., 2004] Ayan, N. F., Dorr, B. J., and Habash, N. (2004). Application of alignment to real-world data: Combining linguistic and statistical techniques for adaptable MT. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-04)*.

[Banerjee and Lavie, 2005] Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan.

[Bennett and Slocum, 1985] Bennett, W. S. and Slocum, J. (1985). The LRC machine translation system. *Computational Linguistics*, 11(2–3):111–121.

[Bowen,] Bowen, T. S. English could snowball on net. `http://www.trnmag.com/Stories/2001/112101/English_could_snowball_on_Net_112101.html`.

[Brown et al., 1993] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

[Buchmann et al., 1984] Buchmann, B., Warwick, S., and Shann, P. (1984). Design of a machine translation system for a sublanguage. In *Proceedings of the 22nd Annual Meeting on Association for Computational Linguistics*, pages 334–337.

[Chiang, 2005] Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270.

[Europa,] Europa. Translation in the commission: where do we stand eight months after the enlargement? `http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/05/10&type=HTML&aged=0&language=EN&guiLanguage=en`.

[Habash, 2002] Habash, N. (2002). Generation-heavy hybrid machine translation. In *Proceedings of the 2nd International Natural Language Generation Conference (INLG-02)*.

[Hakkani et al., 1998] Hakkani, D. Z., Tür, G., Oflazer, K., Mitamura, T., and Nyberg, E. (1998). An English-to-Turkish interlingual MT system. In *Proceedings of AMTA'98: Conference of the Association for Machine Translation in the Americas*, pages 83–94.

[Hakkani-Tür et al., 2000] Hakkani-Tür, D. Z., Oflazer, K., and Tür, G. (2000). Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36:381–410.

[Hovy et al., 2002] Hovy, E., King, M., and Popescu-Belis, A. (2002). An introduction to MT evaluation. In *Machine Translation Evaluation Workshop of Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1–7, Las Palmas, Canary Islands.

[Hutchins, 1986] Hutchins, J. W. (1986). *Machine Translation: past, present, future*. Halsted Press, New York.

[Hutchins and Somers, 1992] Hutchins, J. W. and Somers, H. L. (1992). *An Introduction to Machine Translation*. London: Academic Press.

[Joscelyne, 1987] Joscelyne, A. (1987). Calliope and other pipe dreams. *Language Technology*, 4:20–21.

[Jurafsky and H.Martin, 2006] Jurafsky, D. and H.Martin, J. (2006). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

[Knight, 1999] Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

[Knight et al., 1995] Knight, K., Chander, I., Haines, M., Hatzivassiloglou, V., Hovy, E., Iida, M., Luk, S. K., Whitney, R., and Yamada, K. (1995). Filling knowledge gaps in a broad-coverage MT system. In *Proceedings of the 14th IJCAI Conference*.

[Koehn et al., 2003] Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

[Koskenniemi, 1984] Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 22nd Annual Meeting on Association for Computational Linguistics*, pages 178–181.

[Lavoie et al., 2002] Lavoie, B., White, M., and Korelsky, T. (2002). Learning domain-specific transfer rules: An experiment with korean to english translation. In *Proceedings of the Workshop on Machine translation in Asia, 19th International Conference on Computational Linguistics*.

[Maas, 1977] Maas, H. (1977). The saarbrcken automatic translation system (SUSY). In *Proceedings of the Third European Congress on Information Systems and Networks*, pages 585–592.

[McCowan et al., 2004] McCowan, I., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., and Bourlard, H. (2004). On the use of information retrieval measures for speech recognition evaluation. IDIAP-RR 73, IDIAP, Martigny, Switzerland.

[Nagao and ichi Tsujii, 1986] Nagao, M. and ichi Tsujii, J. (1986). The transfer phase of the mu machine translation system. In *Proceedings of the Eleventh Conference on Computational linguistics*, pages 97–103.

[Nirenburg, 1992] Nirenburg, S. (1992). *Machine Translation: A Knowledge-based Approach*. Morgan Kaufmann Publishers, Inc., San Mateo, California.

[of Pennsylvania,] of Pennsylvania, U. Linguistic data consortium. `http://www.ldc.upenn.edu/`.

[Oflazer, 1994] Oflazer, K. (1994). Two-level description of Turkish morphology. *Literary and Linguistic Computing*, pages 175–198.

[Oflazer and İlknur Durgar El-Kahlout, 2007] Oflazer, K. and İlknur Durgar El-Kahlout (2007). Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical*

Machine Translation, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

[Oflazer and Kuruöz, 1994] Oflazer, K. and Kuruöz, I. (1994). Tagging and morphological disambiguation of Turkish text. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 144–149.

[Oswald,] Oswald, V. A. Word-by-word translation. Presented at the Conference on Mechanical Translation at Massachusetts Institute of Technology, June 1952.

[Özlem Çetinoğlu and Oflazer, 2006] Özlem Çetinoğlu and Oflazer, K. (2006). Morphology-syntax interface for Turkish LFG. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 153–160.

[Papineni et al., 2001] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

[Peterson, 2002] Peterson, E. E. (2002). Adapting a transfer engine for rapid development machine translation. Master's thesis, Georgetown University, Department of Linguistics.

[Probst, 2002] Probst, K. (2002). Semi-automatic learning of transfer rules for machine translation of low-density languages. In *Proceedings of the Student Session at the 14th European Summer School in Logic, Language and Information (ESSLLI-02)*.

[Probst, 2005] Probst, K. (2005). *Learning Transfer Rules for Machine Translation with Limited Data*. PhD thesis, Carnegie Mellon University, Language Technologies Institute.

[Sagay, 1981] Sagay, Z. (1981). A computer translation from English to Turkish. Master's thesis, Middle East Technical University (METU), Department of Computer Engineering.

[Sak et al., 2007] Sak, H., Güngör, T., and Saraçlar, M. (2007). Morphological disambiguation of Turkish text with perceptron algorithm. In *CICLing 2007*, volume LNCS 4394, pages 107–118.

[Stolcke, 2002] Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *In Proceedings of International Conference on Spoken Language Processing*.

[Turhan, 1997] Turhan, C. K. (1997). An English to Turkish machine translation system using structural mapping. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 320–323.

[Varile and Lau, 1988] Varile, G. B. and Lau, P. (1988). Eurotra: practical experience with a multilingual machine translation system under development. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 160–167.

[Witkam, 1988] Witkam, T. (1988). DLT: an industrial R & D project for multilingual MT. In *Proceedings of the 12th Conference on Computational linguistics*, pages 756–759.

[Yuret and Türe, 2006] Yuret, D. and Türe, F. (2006). Learning morphological disambiguation rules for Turkish. In *Proceedings of the Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*.

[Zhai and Lafferty, 2004] Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214.

[Zhang and Vogel, 2006] Zhang, Y. and Vogel, S. (2006). Suffix array and its applications in empirical natural language processing. Technical Report CMU-LTI-06-010, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.