

AN APPROACH FOR MACHINE TRANSLATION BETWEEN TURKISH AND
SPANISH

by

Metin ŞENKAL

B.S. in C.E.I.S., Bilkent University, 2000

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Engineering

Boğaziçi University

2003

AN APPROACH FOR MACHINE TRANSLATION BETWEEN TURKISH AND
SPANISH

APPROVED BY:

Assist. Prof. Tunga Güngör
(Thesis Supervisor)

Assoc. Prof. Levent Arslan

Prof. A.C. Cem Say

DATE OF APPROVAL: 17.06.2003

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Assist. Prof. Tunga Gngr, for giving a huge amount of time for this thesis, and supporting me from the beginning. Also I want to thank to Assoc. Prof. Levent Arslan and Prof. A.C. Cem Say who participated to my thesis jury.

I wish to express my sincere gratitude to one of my best friends, Onur Kardeř who had shared me his experience and knowledge during all of the development process. Without his endless support and encouragement, I would never achieve my current position.

I would like to thank the members of the Spanish Department of Ankara University TMER Language Education Center. Special thanks to Deniz Mren and Oya Altuncu for their great support on Spanish Grammar.

I am also grateful to my parents and sister, who always encouraged me for this MSc. degree.

ABSTRACT

AN APPROACH FOR MACHINE TRANSLATION BETWEEN TURKISH AND SPANISH

In the present thesis, an interlingua based bidirectional machine translation system between Turkish and Spanish has been discussed. In this study, Turkish and Spanish morphology and syntax rules are examined. Spanish verb conjugation with a great variety of morphological combinations are investigated, a finite state automata is designed to cover 53 different types of irregular Spanish verbs. Additionally, semantic representation models are built for both parsing and generation of Spanish and Turkish sentences. Several methods are developed to resolve different ambiguity types.

The main application developed for this study reads the input files in the source language either in Turkish or Spanish, and outputs the translation (if possible) of the sentence in target language. Other programs can be implemented to translate in interactive mode, so that with incorrect input sentences partial transfers can be possible to guide the user to figure out the correct translation.

ÖZET

BİR İSPANYOLCA TÜRKÇE OTOMATİK ÇEVİRİ SİSTEMİ YAKLAŞIMI

Bu tez çalışmasında, aradil tabanlı, çift yönlü bir İspanyolca Türkçe otomatik çeviri sistemi anlatılmaktadır. Bu çalışmada, Türkçe ve İspanyolca biçimbirim ve sözdizim kuralları ele alınmıştır. İspanyolca çok çeşitli biçimbirimsel fiil çekim kombinasyonları incelendi, 53 çeşit kurlsız İspanyolca fiil çekimini karşılamak için sonlu durumlu makine tasarlandı. Ayrıca, İspanyolca ve Türkçe cümlelerin çözümlenmesi ve üretilmesi için anlambilimsel gösterim modelleri geliştirildi. Farklı anlamsal belirsizlik türlerini çözümlemek maksatlı birkaç farklı yöntem geliştirildi.

Bu çalışma sırasında geliştirilen ana uygulama, Türkçe veya İspanyolca dilinde girilen kaynak dosyaları okuyup, buradaki cümleleri (eğer mümkünse) hedef dilde çıkarır. Etkileşim modunda çalışacak başka uygulamalar da geliştirilebilir; böylece hatalı girilen kaynak cümlelerin kısmi çevrimi mümkün olup, kullanıcıya doğru çeviriyi bulması konusunda yol gösterilebilir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF SYMBOLS / ABBREVIATIONS	xi
1. INTRODUCTION	1
1.1. Outline	3
2. LITERATURE SURVEY	4
2.1. Basic Terminology and Definitions	4
2.2. History of Machine Translation	6
2.2.1 First Generation Systems	7
2.2.2 Second Generation Systems	8
2.2.3 Third Generation Systems	9
2.3. Recent Studies on Turkish and Spanish languages	10
3. MORPHOLOGY	14
3.1. Turkish Morphological Analysis	14
3.2. Spanish Morphological Analysis	16
3.3. Grammatical Categories in the Lexicon	23
4. SYNTAX LEVEL	25
4.1. The Main Syntax Rule	25
4.2. Noun Phrases	27
4.2.1 Simple Noun Phrases	27
4.2.2 Prepositional Noun Phrases	28
4.2.3 Noun Phrases with Adjectives	29
4.2.4 Proper Noun Phrases	29
4.2.5 Definite Noun Phrases	30
4.3. Verb Phrases	30
5. SEMANTICS LEVEL	33
5.1. First Order Predicate Calculus	33

5.2. Lambda Calculus.....	33
5.3. Semantic Representation	34
5.4. Parsing Stage.....	35
5.4.1 Verbs.....	35
5.4.2 The Object Phrase.....	36
5.5. Generation Phase.....	39
5.6. Ambiguity Resolution	40
5.6.1 Lexical Ambiguities	41
5.6.1.1 Category Ambiguity.....	41
5.6.1.2 Homography and Polysemy	42
5.6.1.3 Transfer Ambiguity.....	44
5.6.2 Structural Ambiguity	44
6. APPLICATION and RESULTS	45
6.1. Ambiguity Resolution	47
7. CONCLUSION.....	50
7.1. Future Work.....	50
7.1.1 Complex Sentences.....	51
7.1.2 Incorrect Input Translation.....	51
7.1.3 Specific Domain	51
REFERENCES	53
REFERENCES NOT CITED	55
APPENDIX A: SPANISH VERB TYPES.....	56
A.1. Irregularities in Present Tense.....	56
A.1.1. Verbs with diphthong.....	56
A.1.2. Change in the stem vocale.....	56
A.1.3. Consonants are added in the stem.....	57
A.1.4. Other irregularities.....	57
A.2. Irregularities in Preterite Tense.....	57
A.2.1. Change in stem vocale	57
A.2.1. Strong preterites.....	57
A.3. Irregularities in Future and Conditional Tenses.....	58
A.3.1. Loss of the protonic vocal	58
A.3.2. Loss of vocal and consonant	58

A.3.3. Loss of vocal and addition of consonant.....	58
A.4. Orthographic Modifications for Preserving the Pronunciations	58
A.5. Changes due to Orthographic Rules	59
A.5.1. Loss of atonic i	59
A.5.2. Atonic i changes to y.....	59
A.6. Change of the Orthographic Accent in Verbs Ending in -iar and -uar	59
A.7. Other changes of the orthographic accent.....	59

LIST OF FIGURES

Figure 2.1. The Vauquois triangle	5
Figure 3.1. FSM for Spanish Participles	19
Figure 3.2. FSM for Aorist Tense.....	20
Figure 3.3. FSM for Past Tenses, Definite and Narrative.....	20
Figure 3.4. FSM for Imperative Tense.....	21
Figure 3.5. FSM for Subjunctive Tense, Present and Imperfect	21
Figure 3.6. FSM for Future and Conditional Tenses	22

LIST OF TABLES

Table 4.1 Turkish Phrase Structure Rules.....	26
Table 4.2 Spanish Phrase Structure Rules.....	27

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artificial Intelligence
MT	Machine Translation
MAHT	Machine-Aided Human Translation
HAMT	Human-Aided Machine Translation
FAHQT	Fully Automatic High Quality Translation
TAUM	Traduction Automatique de l'Université de Montréal
GETA	Groupe d' Etudes pour la Traduction Automatique
SUSY	Saarbrücker Übersetzungssystem-Saarbrücken Translation System
METAL	Mechanical Translation and Analysis of Language
LRC	Linguistics Research Center
UPM	Universidad Politécnica de Madrid
UAM	Universidad Autónoma de Madrid
KR	Knowledge Representation
DCG	Definite Clause Grammar
FOPC	First Order Predicate Calculus
FSM	Finite State Machine
LISP	List Processing Language
NFA	Non-Deterministic Finite Automata
NLP	Natural Language Processing
SFG	Systemic-Functional Grammar
SOV	Subject-Object-Verb
SVO	Subject-Verb-Object
TADJ	Turkish Adjective
TADVP	Turkish Adverbial Phrase
TCNP	Turkish Conjunctive Phrase
TNP	Turkish Noun Phrase
TVP	Turkish Verb Phrase
TS	Turkish Sentence
SADJ	Spanish Adjective

SADVP	Spanish Adverbial Phrase
SNP	Spanish Noun Phrase
SVP	Spanish Verb Phrase
SS	Spanish Sentence
\exists	Existential Quantifier
\forall	Universal Quantifier

1. INTRODUCTION

Translation of information between different languages has been a need in society for thousands of years. The first recorded literary translation was done by Livius Andronicus who translated the Odyssey from Greek into Latin in 240 BC. The Rosetta stone, carved in 197 BC, includes a translation of a hieroglyphic passage into Greek and provided the key to deciphering Egyptian hieroglyphics in the early 19th century. Thus, translators have enabled communication and transmission of information from one culture to another for thousands of years.

Translation studies are done by many academic disciplines. They can be listed as linguistics, anthropology, psychology, literary theory, philosophy, cultural studies and various other disciplines of knowledge, as well as on its own techniques and methodologies. Translation requires advanced skills in the source and target languages. Especially in the literary translation which is the translation of poetry, drama and other literary works from one language to another, the ability to choose the correct translation of an element given a variety of factors is vital. By contrast, most of the translations in the world do not contain a high level literary and cultural knowledge. The majority of professional translators are working on translations of scientific and technical documents, commercial and industrial transactions, legal documentation, instruction manuals, technical and medical text books, industrial patents, news reports, etc.

In the past few decades, there have been a number of drastic and important changes in Machine Translation (MT), one of the oldest non-numeric application fields of the computer science. In the middle of the twentieth century, the large mainframe MT systems were originally intended to produce translations fully automatically, but in practice they are rarely used without human intervention. As the complexity of the linguistic problems became more and more apparent, less ambitious goals were determined. Human assisted machine translation in narrow subject domains was a reasonable objective. Current systems are possible to take unedited (raw) source documents and produce good quality output without assistance from professional translators in well-defined areas. At the moment,

there is not a machine translation system which can take any text in any language and generate a perfect translation in any other language without human intervention or assistance.

In the last few years, the need for translation has grown ever more urgent, far beyond the capacity of the professional translators. Due to the growth of telecommunications and internet usage, there is an enormous increase in the information flow across wider global markets requiring translations into more languages; have forced many institutions to devote ever-increasing efforts to computer processing of natural languages. One of the other reasons behind the demand for technical translation is localization problem. The countries exporting technical products, such as software packages, have to ensure that the user manuals and other relevant documents must be accurate, consistent and understandable by the recipients. The texts to which computational techniques are applied are generally non-literary in nature and much of the work is repetitive. The computer technology has been applied in technical translation in order to improve speed and cost factors. Translation by or with the aid of machines can be faster than manual translation and the cost of a translation is much cheaper.

There are many machine translation systems for different languages such as English, French, German, Spanish, Russian, Japanese, etc. Turkish language due to its morphological and grammatical differences remains almost undiscovered. A few academic and commercial systems were developed. With the multiplicity of system types and of research designs in the Turkish MT, the systems can be further extended to a higher coverage, accuracy and fluency. The increasing number of Turkish MT systems, based on linguistic framework, is promising for a high-quality Turkish translation in near future.

The main objective of this thesis is to build a software infrastructure for machine translation between Spanish and Turkish. During the development of the machine translation system, detailed analyses of the differences between Spanish and Turkish languages are performed. Morphological, syntactic and semantic structures of both Turkish and Spanish languages are examined in detail.

1.1. Outline

The next Chapter is a brief summary of some of the previous work on the machine translation area; different MT systems that take place during the progress of this research field. In Chapter 2, history of the machine translation and its definition will be discussed with real world examples. The Turkish and Spanish morphology and their implementation details are investigated in Chapter 3 which is followed by Chapter 4 that presents the works on the syntactic level, word order in Turkish and Spanish sentences, general grammar rules and phrase structure rules. Chapter 5 covers the semantic representation methods that are developed for the intermediate representation of both languages. The last Chapter summarizes the work done and discusses the limitations and the extendibility of the system with possible further developments.

2. LITERATURE SURVEY

In this Chapter, the context in which machine translation takes place and several strategies for machine translation with real-world examples will be discussed.

2.1. Basic Terminology and Definitions

Machine Translation is not in itself an independent field of 'pure' research; it is an interdisciplinary study combining linguistics, computer science, artificial intelligence, translation theory, computational algorithms and data structures, cognitive science, and study of human-computer interaction. The term *Machine Translation* (MT) is the now traditional and standard name for computerized systems. Briefly, the task of MT is to feed a text in one natural language (source language, SL) into a computer and, using a computer program to produce a text in another language (target language, TL), such that the meaning of the TL text is the same as the meaning of the SL text. Earlier names such as *mechanical translation* and *automatic translation* are now rarely used.

The term does not include computer-based translation tools which support professional translators, generally called *Machine-Aided Human Translation* (MAHT). However, it includes the systems in which professional translators or other users assist programs in the production of translations, so called *Human-Aided Machine Translation* (HAMT). The *Computer-Aided* (or *Computer-Assisted*) Translation covers both. But the final goal of MT is fully automated high quality translation process.

The major problems of MT are not computational but linguistic. In brief, they are lexical ambiguity, syntactic complexity, elliptical and 'ungrammatical' constructions, etc. In order to avoid these problems, sometimes input texts may be written in a controlled language, which reduces potential ambiguities and restricts the complexity of sentence structures. This is called *pre-editing*. Alternatively, the output of the MT systems is revised which is known as *post-editing*. Finally the systems may refer problems of ambiguity for resolution to human operators during the run time, which is named as *interactive mode*.

MT systems are normally classified in terms of their basic strategy for carrying out translation. In the *direct approach* systems, there are extensive string pattern matching operations, with some rearrangement of the target string for conformance to the target language word order. Source texts are analyzed no more than necessary for generating texts in the target language.

The second basic type is the *transfer approach* which involves analysis of the source input into a transfer structure in which ambiguities have been resolved irrespective of any other language. After analysis, the source language structure is transferred into a corresponding target language structure which is then used to generate a target text. Analysis and generation programs are specific for particular languages. Differences between languages, in vocabulary and structure, are handled in the intermediary transfer program.

The third type is *interlingua approach* where source language sentences are analyzed into a language-neutral representation, common to more than one language. Thus the translation is done in two steps; from the source language to the interlingua, and from the interlingua into the target language. This strategy eliminates the need for a transfer step. Programs for analysis are independent from programs for generation; in a multilingual configuration, any analysis program can be used together with any generation program.

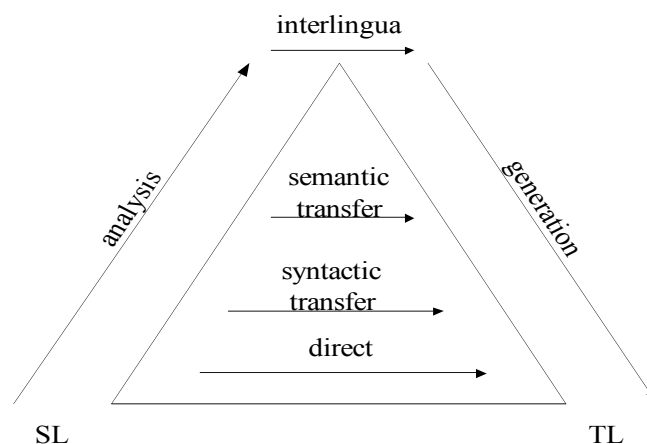


Figure 2.1. The Vauquois triangle

These three types of transfer systems are illustrated using the *Vauquois triangle* shown in Figure 2.1. In the left-hand side of the well-known 'pyramid', the amount of effort necessary for source language analysis is illustrated, and down the right hand side, the amount of effort for target language generation. In the horizontal dimension the amount of effort needed for transfer is shown. The direct method stands at one extreme, the interlingua method at the other, with transfer-based systems in between. At the apex, transfer effort is at a minimum, while analysis and generation are at a maximum. The transfer and interlingua methods use simpler or minimal transfer components compared with direct methods.

Machine Translation is considered as an *AI-complete problem*, at least in the fully automatic high quality translation case which means that it requires a solution to every other major problem in AI. In addition, MT provides a 'test-bed' for theories and techniques developed in some of well-defined areas of AI, such as computational linguistics, knowledge representation, machine learning and search algorithms. Computational linguistics deals with linguistic phenomena. Knowledge representation deals with the formalization of the procedures which represent knowledge about a domain. Machine learning can be used in grammar learning, to obtain new knowledge from data. Different search algorithms can be used to avoid getting stuck in an infinite loop.

2.2. History of Machine Translation

The idea of using computers for translation is almost as old as the computer itself. As one of the oldest fields of computer science, it has been influenced by the politics, science and economics of different periods of modern history.

Some of the ideas that have influenced MT existed in the 17th century. Both Descartes and Leibniz speculated on the creation of dictionaries based on universal numerical codes in order to overcome linguistic barriers. In 1933, a mechanical procedure for carrying out translation was patented by the Russian Petr Smirnov-Troyanskii. The idea of MT is brought to the general notice by a memo from Warren Weaver in July 1949 who proposed specific strategies for using computers to translate natural languages. In the

famous Weaver Memorandum, he wrote “When I look at an article in Russian, I say this is really written in English, but it has been coded in some strange symbols. I will now proceed to decode” [1].

2.2.1 First Generation Systems

Within a few years research had begun in a few US research centers. The first public demonstration of a Russian-English prototype MT system was performed in 1954 [2]. The system was developed at Georgetown University with the IBM™ cooperation and included 250 words and just six grammar rules. Later it led the initiation of MT projects at other places in the world, including Soviet Union.

The first generation systems lacked of detailed syntactic analysis and had no semantic analysis. These systems had “undirectionality of translation, unseparability of linguistic data and processing algorithm, and one-problem-one-solution unvariantness” [3]. The first generation of MT systems adopted direct approach for carrying out translation. In these systems, rules for analysis, transfer and generation were not always clearly separated. Some also merged linguistic data and computer processing rules and routines.

However, the initial common assumption of building Fully Automatic High Quality Translation (FAHQT) systems capable of results indistinguishable from those of human translators was not achieved. In 1966, the Automatic Language Processing Advisory Committee (ALPAC), commissioned by government sponsors of MT, concluded that MT was slower, less accurate and twice as expensive as human translation and stated that “there is no immediate or predictable prospect of useful Machine Translation” [2].

The ALPAC report brought an end to MT research in the United States for over a decade and damaging the public perception of MT. In the following decade MT research continued mainly outside the USA. Continued effort in MT yielded operational systems in the early 1970s. *Systran*, an acronym for 'System Translation', began Russian-English translations for the US Air Force in 1970. Its designer, Peter Toma, was the principal programmer of the SERNA implementation of the Georgetown University GAT system for

Russian-English translation. Although the system was a good example of the direct approach to MT design, it has evolved into a transfer based system in its 20 years of operational service. Its high degree of modularity in the system design provided the developers to produce a wide range of language pairs in the following years. In 1976, the Commission of the European Union installed an English-French version of *Systran* [2]. Further systems for French-English, English-Italian, English-German and other pairs have been developed for the Commission.

2.2.2 Second Generation Systems

The direct approach, a relatively unsophisticated approach to the computational side of the problem of translation, was one of the reasons for the apparent failure of the earliest first generation MT systems. The usage of linguistic theories and representations in the machine translation systems are the main differences between the first generation and second generation systems. By contrast to the failure of direct approach, in second generation systems, two types of indirect approach were developed; the transfer approach and the interlingua approach. The failure of the first generation systems led to the development of sophisticated linguistic models for translation. In particular, there was increasing support for the syntactical analysis of the source language texts in second generation systems.

The Canadian government introduced its bilingual policy, which required all official documentation to be available in both French and English and the Canadian National Research Council began sponsorship of MT research. In 1976, the TAUM (Traduction Automatique de l'Université de Montréal) group was established in Montreal which developed a transfer-based, second generation MT system called *Météo* for translating weather bulletins from English into French. The system was installed in 1976 and has been in daily operation from the following year to present time. The reason behind its success is the completeness and accuracy that is achieved by the restriction to the sublanguage of meteorological forecasts.

At the University of Grenoble, the Grenoble group, formerly known as CETA (Centre à Etudes de la Traduction Automatique-'Centre for the Study of Machine Translation'), concentrated on the development of formalisms for expressing linguistic information. The original CETA system, developed between 1960 and 1970 and including three language pairs (into French from Russian, German and Japanese), was an interlingua system. A change in computer facilities in 1971, among other reasons, led the Grenoble team (renamed GETA: Groupe d' Etudes pour la Traduction Automatique) to rethink the design of their MT system and to design a transfer system, officially known as Ariane, though often referred to simply as the 'GETA system' [2].

In 1972, another second generation system was developed at the Universität des Saarlandes in Saarbrücken, Germany known as SUSY (Saarbrücker Übersetzungssystem-'Saarbrücken Translation System'). It was a multilingual system involving the German, Russian, English, and French languages.

2.2.3 Third Generation Systems

The systems that are in the third generation of machine translation can be characterized as those that include a semantic or understanding capability as well as a general approach to syntax. They aim at storing semantic information in a knowledge base which can then be interrogated by the translation process when needed [4].

The Commission of the European Union was demanding translations of scientific, technical, administrative and legal documentation from and into all the Community languages. The *Eurotra* project from the European Community began in 1982 and was influenced by the work done at Grenoble and Saarbrücken since the 1960s and 1970s. It has been the biggest MT project yet, both in terms of number of personnel and amount of expenditure. More information is given in the following paragraph taken from Hutchins and Somers [2].

In 1978 there were discussions in the Commission on a project for the creation of a machine translation system of advanced design (*Eurotra*) capable of dealing with all the official languages of the Community. At that time these were Danish, Dutch, English, French, German, and Italian; Greek

was added soon afterwards. In 1986 Portuguese and Spanish were added. Eurotra is intended, therefore, to translate between nine languages, in all 72 language pairs. Eurotra has successfully broadened the research base of European computational linguistics, and it has promoted awareness in governmental bodies of the growing economic and cultural importance of basic and applied research in natural language processing.

Another well-known third generation system is METAL (the acronym of Mechanical Translation and Analysis of Language) developed by Linguistics Research Center (LRC) at the University of Texas (Austin, Texas). Like the CETA system at the University of Grenoble, the LRC interlingua design was interlingual only with respect to syntactic representations: translation of lexical items was handled by rules of lexical transfer. In 1978, Siemens began to support the project financially, and the system changed from an interlingua design to an essentially transfer-based approach, not intended to operate fully automatically but to be augmented by sophisticated text-editing facilities and access to large terminology databanks. It is a system for mainframe computers and it represents one of the most advanced operational systems at the present time.

Another third generation MT project is Rosetta, developed at the Philips Research Laboratories in Eindhoven (Netherlands), has its roots in earlier research at Philips on a question-answering system, PHLIQA. The task was to convert a question expressed in English into the logical representation language of the database. The system uses interlingual representations based on the principles of Montague grammar, a theory which directly links syntax and semantics. The Rosetta interlingua representation is defined by the isomorphic grammars of the languages of the system. The system performs translation from Dutch into English or Spanish and from English or Spanish into Dutch.

2.3. Recent Studies on Turkish and Spanish languages

In the latest years, there has been several studies on Turkish language which dealt with different levels of NLP.

Hamzaoglu [5] was one of the first researchers that developed a Turkish Machine Translation system. He built a lexicon-based translator between two very similar

languages, Turkish and Azeri languages. Syntactic structures of sentences are similar for Turkish and the Azeri language; as a result of this similarity, he did not make any syntactic analysis. He carried out the morphological analysis and then tried to derive semantic representations of sentences belonging to these languages. The lexicon has 6,900 root entries and about 10,000 proper nouns.

In 1997, two independent systems, developed at Carnegie Mellon University and Bilkent University, were combined as a knowledge-based machine translation system [6]. A morphological generation system and a tactical sentence generation system had been implemented for Turkish in the framework of a large-scale research project. At the same time, a system which produces an interlingua representation from an English sentence was developed for the KANT project at Carnegie Mellon University, Center for Machine Translation [7]. A mapping system is built in order to transform the interlingua representation of a sentence to an f-structure for Turkish, which will serve as input to the sentence generator. Then the surface forms of the sentences are realized by the morphological generation system. The system was developed in the domain of computer manuals with an example set of 52 sentences; it was able to translate 44 of them correctly and completely. All the sample sentences are produced in the default order in Turkish because the mapping system did not produce an information structure which encodes word order information.

Another transfer-based, human-assisted English to Turkish machine translation system was developed by Turhan [4] as a Ph.D. thesis in the Middle East Technical University in 1998. The source language intermediate representation utilized stores the feature structures and grammatical functions of the input sentence in a language independent formalism allowing the system to extend to other source languages. The important issues dealing with the differences between the languages are reflected as complex transfer rules in the transfer module. Turhan [4] had tested the system in the translation of IBM manuals, and produced around 60 percent understandable and accurate translation with no post-editing involved. There were plans to develop an analyzer for German which would be incorporated into the machine translation system resulting in a multilingual system translating English and German into Turkish.

There is other research in Turkish usually dealing with sublevels of NLP. One of them is Çetinoğlu's M.S. Thesis at Boğaziçi University [8]. She developed an application called TOY, which is a man-machine communication program that provides a human-computer conversation using Turkish sentences. She built a morphological analyzer, derived necessary syntactic rules and worked on the semantic representations. The work done in semantic levels were somehow introductory. In our research, the Turkish morphological analyzer and some of the Turkish syntax rules are based on her thesis.

Kardeş [9], in his M.S. Thesis in Boğaziçi University, developed an application that reads a recipe file, and outputs the semantic representations of Turkish sentences. He derived complex noun and adverbial phrases. His syntactic rules and semantic formulas are taken as a basis for most of the Turkish syntactic and semantic analysis in this thesis.

The ARIES Natural Language Tools [10] make up a lexical platform for the Spanish language, developed by Natural Language Processing Group, a joint group of participants from Universidad Politécnica de Madrid (UPM) and Universidad Autónoma de Madrid (UAM). At the beginning of their study, they claimed that "The real situation of the Spanish language in relation with language technology is far behind most of its European counterparts. This situation is specially worrying if we take into account the spreading of Spanish all around the world and, therefore, the potential market for Spanish processing tools".

A set of tools have been implemented, including tokenizers, spell checker, the modular chart parsing system called NUCLEO , stochastic and neural morphosyntactic taggers, a prototype of morphological analyser/generator called GRAMPAL. The core of the toolbox is its lexical platform.

In 1998 a company, DAEDALUS-Data Decisions and Language S.A. TM, was founded as a spin-off from ARIES research group, partially to develop and market products from ARIES. The initial ARIES toolkit, mostly a research prototype, was greatly expanded and improved; the new project is called STILUS (Servicios Telemáticos de Ingeniería Lingüística) [11]. It is installed for a certain number of commercial applications, ranging from spell, grammar and style checking of texts in Spanish, fuzzy and semantic

search in Spanish, multilingual information retrieval, etc. Instituto Cervantes, the official institution in charge of promoting the Spanish language in the world, use STILUS for high quality spelling, grammar and style checking of Spanish texts.

Prolog is one of the most widely used programming languages in computational linguistics. Some of the features that make Prolog suitable for NLP research is its efficient unification operation, its backtracking property, reversible process and advance list manipulation techniques. Because of above listed features and considering that previous studies are developed in Prolog, it is selected as the programming language in this research.

3. MORPHOLOGY

Morphology is concerned with the structure of words, how a simple word is formed with a sequence of morphemes which are the smallest meaning-carrying units in a language. Two types are distinguished: inflectional morphology and derivational morphology. In inflectional morphology, words vary in their form but their main syntactic category remains the same. On the other hand, in derivational morphology new words are formed. For example, adding the suffix *-ly* turns an adjective into an adverb (e.g. rapidly, slowly). Compounding is concerned with the derivation of new words from the combination of two or more independent words.

Languages can be classified according to the number of morphemes used in inflectional morphology. In isolating languages, there is one morpheme per word and words have little or no internal structure; a good example is Chinese. Polysynthetic languages, such as Eskimo, have a large number of morphemes per word, particularly multiple roots. Eskimo is a polysynthetic language where most of the grammatical meaning of a sentence is expressed by inflections on verbs and nouns. Turkish is an agglutinative language in between isolating and polysynthetic languages, where inflectional suffixes can be added repetitively, one after another. Spanish is also an agglutinative language; instead of adding multiple suffixes, basic suffixes carry multiple meanings. For example, in the word *habl-o* (I speak), person (1st), number (singular) and tense (present) features are encoded in just one morpheme *-o*.

3.1. Turkish Morphological Analysis

The Turkish morphological analyzer gets a correct Turkish word and breaks it down to its stem and suffixes. The previous studies on Turkish morphology are used in this research for the morphological processing. Çetinoğlu [8] has improved Oflazer's [12] nominal and verbal finite state machines (FSMs). The FSMs are based on 22 two-level rules with Turkish lexical and surface forms. The original FSMs were designed as parsers, not generators, with the assumption that the input text is correct.

Çetinoğlu [8] tried to restrict incorrect word generations with different approaches. One of them was inserting flags into the morphological entries to regulate the transitions of verbs. Defining new arguments or flags enables the system to verify if the verb can take the derivational suffix or not.

There are two basic techniques in parsing a word of an agglutinative language, namely, Root Matching (Left-to-right) and Suffix Stripping (Right-to-left). In general, the left-to-right approach of root matching is preferred to the right-to-left approach of suffix stripping [13]. In the first approach, the morphotactics of the language, which arranges the suffix order, reduces the possible suffix alternatives that will be attached to the root. The number of suffixes that may be affixed to a stem is smaller than the number of roots that a suffix may be attached to. At first, the word to be parsed is assumed to be a verb and the transducer tries to find a verb stem that can be a part of the word and later if a verb stem is found the suffixes that can be attached to the verbs are examined. In suffix stripping every time a suffix is stripped off, the algorithm seeks the whole root lexicon to find an appropriate match. In the root-matching algorithm. However, the remainder substring is sought in a small set of suffix lexicon. This increases the efficiency of the root-matching algorithm compared to the suffix-stripping algorithm.

The parsing begins with an initial node and the arguments are passed through the transducer part of the program. ‘AdKök’ (noun stem), ‘FiilKök’ (verb stem), ‘ZamirKök’ (pronoun stem) ‘SıfatKök’ (adjective stem) and ‘ZarfKök’ (adverb stem) stand for the values of the morphological category that we are using to drive the FSM rule invocation. All the dictionary entries are coded, with a label that corresponds to its morphological category. Three of them are the initial nodes in the FSM; ‘FiilKök’ for verbs, ‘AdKök’ for nouns and adjectives, and ‘ZamirKök’ for pronouns. The transducer begins from the initial node, jumps to the next node by following the arcs of Turkish FSM, and traverses the input word. Each time the transducer is called, one morpheme of the whole input word is removed from the input list of characters and the meaning of that morpheme is added to the semantic list. The process is repeated recursively. At the end, if the input list becomes an empty string and the transducer is in one of the final states, the output of the morphological analysis gives the stem and the suffixes. If the transducer fails for verbs, it goes over the noun and the pronouns in the same manner.

This automaton works well if the FSM is formed correctly and fully; so that it always finds the correct parse. But since it works in a non-deterministic manner, it can produce several parses for some cases. For example, for a very simple Turkish word, “*koyun*” it produces four outputs: first one is “*koyun*” in nominative case in the meaning sheep, second output is again in nominative case which means chest, third output is another noun in accusative form which corresponds to your bay. In addition to the noun forms, the last output is verb in imperative form of put. Since FSM lets Prolog try all the alternatives, it gives the correct output, but it also gives three different ones. Those should be eliminated if possible or can be listed to the professional user to select the appropriate one according to the context. But in the morphology level, the analyzer cannot do this distinguishing mechanism. These ambiguities will be handled in higher level analysis, some in syntactic, others in semantic level analysis.

3.2. Spanish Morphological Analysis

In Spanish morphological study, we focused on words. Spanish morphology is not a trivial subject; shows a great variation of morphological combinations. We have listed 53 different verb types that have different word forms based on tense, person and number. Nouns and adjectives have also four different forms, based on gender and number.

Moreno and Goñi [14] list the problems which any morphological processor of Spanish has to deal with as follows:

- A highly complex verb paradigm. For simple tenses, 53 inflected forms of verbs are counted.
- The frequent irregularity of both verb stems and endings. Very common verbs, such as *tener* (to have), *poner* (to put), *poder* (to be able to), *hacer* (to do), etc., have up to 7 different stems e.g., *hac-er*, *hag-o*, *hic-e*, *ha-ré*, *hiz-o*, *haz*, *hech-o*.
- Gaps in some verb paradigms. In the defective verbs some forms are missing or simply not used. For instance, meteorological verbs such as *llover* (to rain), *nevar* (to snow), etc. are conjugated only in third person singular form.
- Duplicate past participles: a number of verbs have two alternative forms, both correct, like *impreso* (printed) and *imprimido* (printed). In such cases, the analysis has to treat both.

- There exist some highly irregular verbs that can be handled only by including many of their forms directly in the lexicon like *ir* (to go), *ser* (to be), etc.
- Nominal inflection can be of two major types: with grammatical gender i.e. concatenating the gender morpheme to the stem and some nouns and adjectives present alternative correct forms for plural.
- In contrast with verb morphology, nominal processes do not produce internal change in the stem caused by the addition of a gender or plural suffix, although there can be many allomorphs produced by spelling changes: *luz* (light), *luc-es* (lights).

Lexical and surface forms of Spanish words are represented in orthographic and structural way. Below is an example lexicon entry for Spanish verb *tener* (to have):

```
esp_morph_entry('FiilKök',[[t,e,n],[type(verb),sem(Time^Abl^Loc^Dat^Theme^Agent^have(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))]],0,2,10).
```

```
esp_morph_entry('FiilKök',[[t,i,e,n],[type(verb),sem(Time^Abl^Loc^Dat^Theme^Agent^have(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))]],1,2,10).
```

```
esp_morph_entry('FiilKök',[[t,u,v],[type(verb),sem(Time^Abl^Loc^Dat^Theme^Agent^have(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))]],3,2,10).
```

A Spanish morph entry contains the label of the morpheme, *FiilKök*, the morpheme as a character list, *[t,e,n]*. The stem type value 0 which is the first of the last 3 numbers represents that the verb is in infinitive form. In the second and third entries the character list is *[t,i,e,n]* and *[t,u,v]* and the stem types are 1 and 3 respectively.

The morph entry contains also semantic meaning of the morpheme. Here *Time*, *Abl*, *Loc*, *Dat*, *Theme*, *Agent* and *have* are parameters that will be used during syntactic and semantic analysis. These are explained in Section 3.3. The tense of the Spanish verb will be retrieved by traversing over the arcs of FSM and obtaining the suffix type which will

give the tense information. The type of the Spanish verb, type 2, which means verb is ending with –er, and group number, 10, which indicates irregular group 10. We have figured out 53 irregular verb types; they are listed in different categories in Appendix A.

The aim of a morphological analysis module in a machine translation system is to reduce the size of the lexicon, especially for languages with inflectional morphology. With morphological analysis, all words do not have to be kept in the lexicon, instead only the root forms of the possible words are stored. But as shown in the above example, for some of the irregular verb types we have to enter the other distinct irregular forms of the verbs. During affixations of suffixes letter changes and deletions may occur in the stem or in the suffix itself. Our Root Matching (Left-to-Right) technique limits us to handle each stem changes with different entries. The same applies to Turkish entries; words that are modifiable by a Turkish morphophonemic rule are represented as separate lexical entries. During the morphological analysis of Turkish vowel and consonant harmony rules, Çetinoğlu [8] has separated lexical entries where its dictionary form and its alternative form have the same semantics. Phoneme transformations of these rules are given as predicates and different functions are defined for morphophonemic rules, two of them are *vtoa* (Vowel harmony rule) and *ltod* (Consonant harmony rule). The last vowel and the last letter are given according to the rules.

```
tr_morph_entry('AdKök',[[ç,o,c,u,k],[type(noun),sem(X^child(X))]],_ ,_ ,_ ,u,k,specok
).
```

```
tr_morph_entry('AdKök',[[ç,o,c,u,ğ],[type(noun),sem(X^child(X))]],_ ,_ ,_ ,u,ğ,spec).
```

```
tr_morph_entry('DA',[[D,A],[case(loc)]],V,L,OK,A,A,ok):-
```

```
vtoa(V,A),
```

```
ltod(L,D),!
```

```
ok(OK).
```

Spanish verbs are grouped in 3 types based on the verb endings (-ar, -er, -ir). In the following figures the Finite State Machine diagram of the Spanish verb conjugation is illustrated. Available Spanish tenses are aorist, narrative past, definite past, present subjunctive and imperfect subjunctive, future, conditional, and imperative. In addition to those tenses, participles, present participles and past participles are also covered by this FSM.

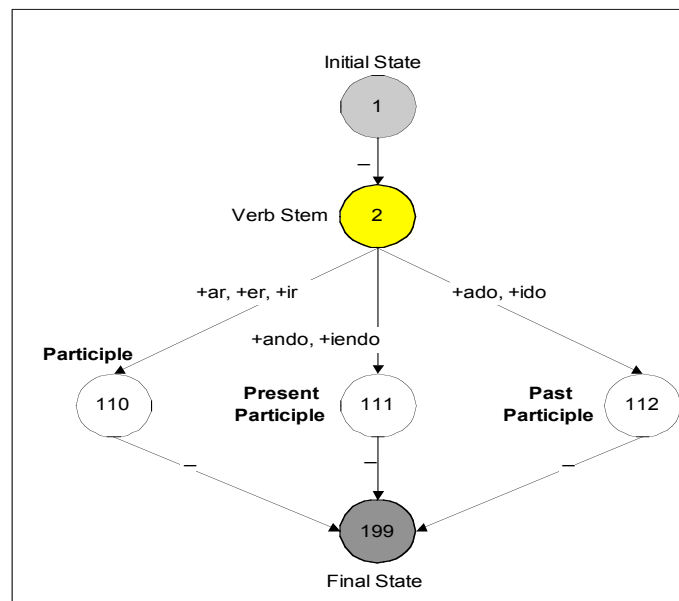


Figure 3.1. FSM for Spanish Participles

In Figure 3.1, Spanish participle part of the FSM is illustrated. The infinitive forms of Spanish verbs are handled by Participles. For example, Spanish verb *hablar* (to talk) is entered into the dictionary as [h,a,b,l] and when it takes *+ar* suffix it goes to Participle and with an epsilon transition it reaches the final state. Similar to *hablar* other verb ending types like *comer* (to eat) and *vivir* (to live) are entered as [c,o,m] and [v,i,v] and they get *+er* and *+ir* accordingly. When the verbs ending with *+ar* get *+ando* suffix they go to Present Participle and again with an epsilon transition to final state where other types follows the same arcs with the *+iendo* suffix. With the suffixes *+ado* and *+ido* these verbs reach the Past Participle state.

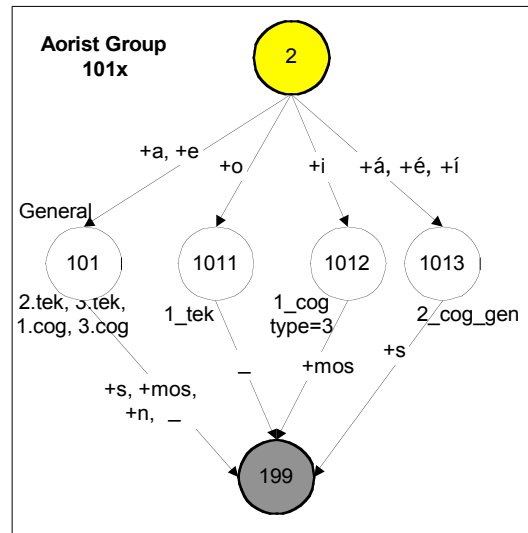


Figure 3.2. FSM for Aorist Tense

In Figure 3.2, the Aorist tense is handled. The verbs ending with *+ar* take *+a* as aorist tense suffix, except the first person singular and second person plural. In aorist tense, first person singular takes *+o* for all verb endings. Such as, *hablo* (I talk), *como* (I eat) and *vivo* (I live). So with *+o*, instead of going to general aorist state (100), it goes to 1011 and with an epsilon transition to final state.

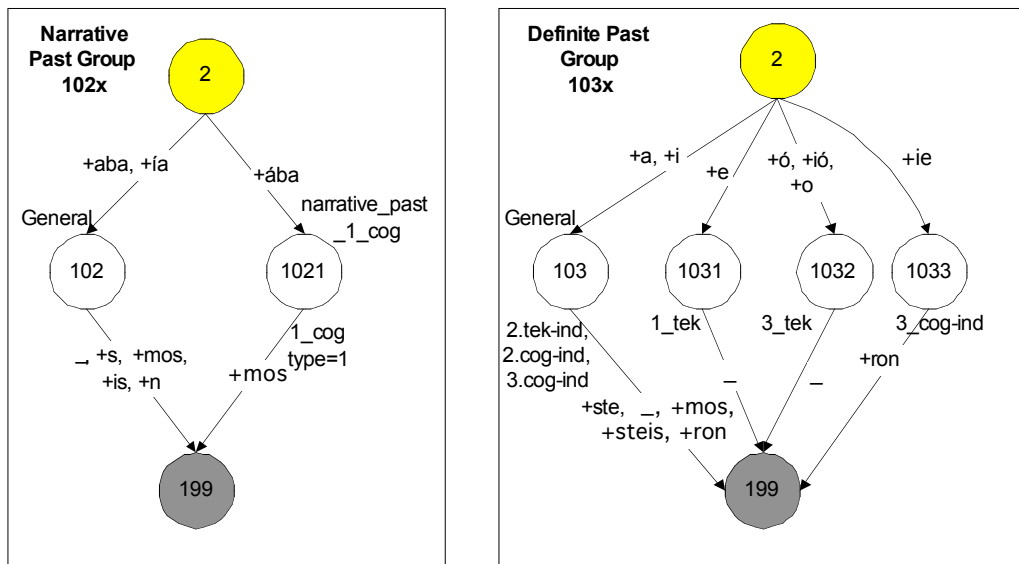


Figure 3.3. FSM for Past Tenses, Definite and Narrative

In Figure 3.3, some states of both definite past and narrative past tenses are shown. In Spanish, it is the past tense that has the most irregularities, so not all of the states are illustrated in Figure 3.3.

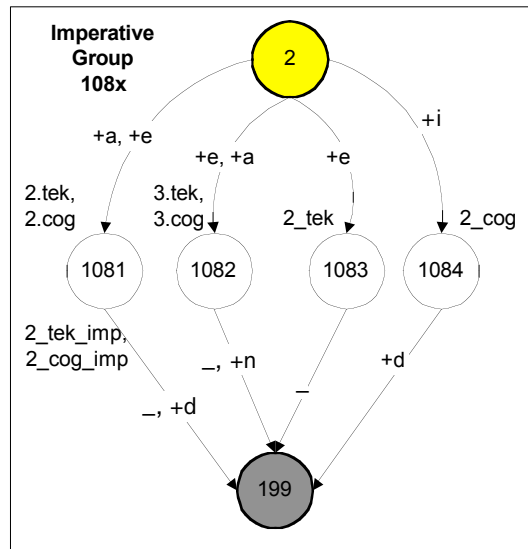


Figure 3.4. FSM for Imperative Tense

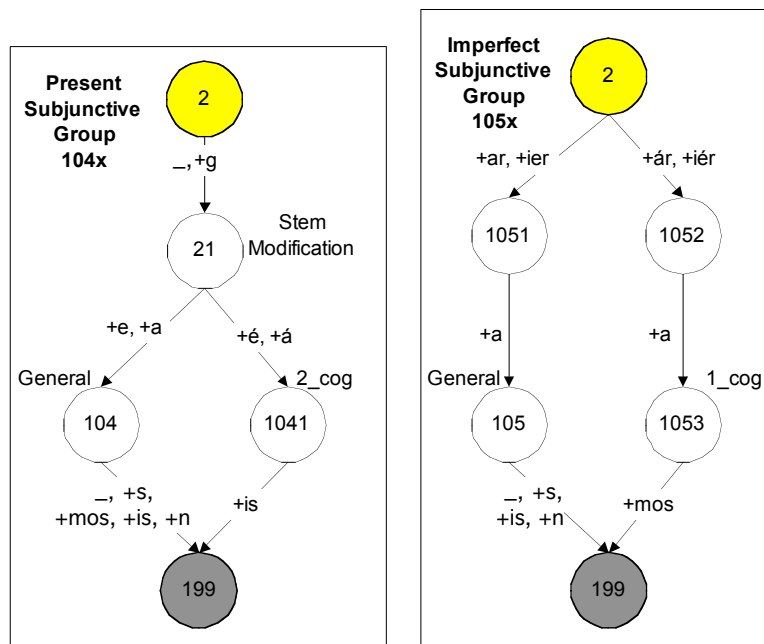


Figure 3.5. FSM for Subjunctive Tense, Present and Imperfect

In Figure 3.4, 3.5 and 3.6 other Spanish tenses, the Imperative, Present Subjunctive, Imperfect Subjunctive and Future and Conditional, are illustrated. The tense and personal suffixes are shown in all figures.

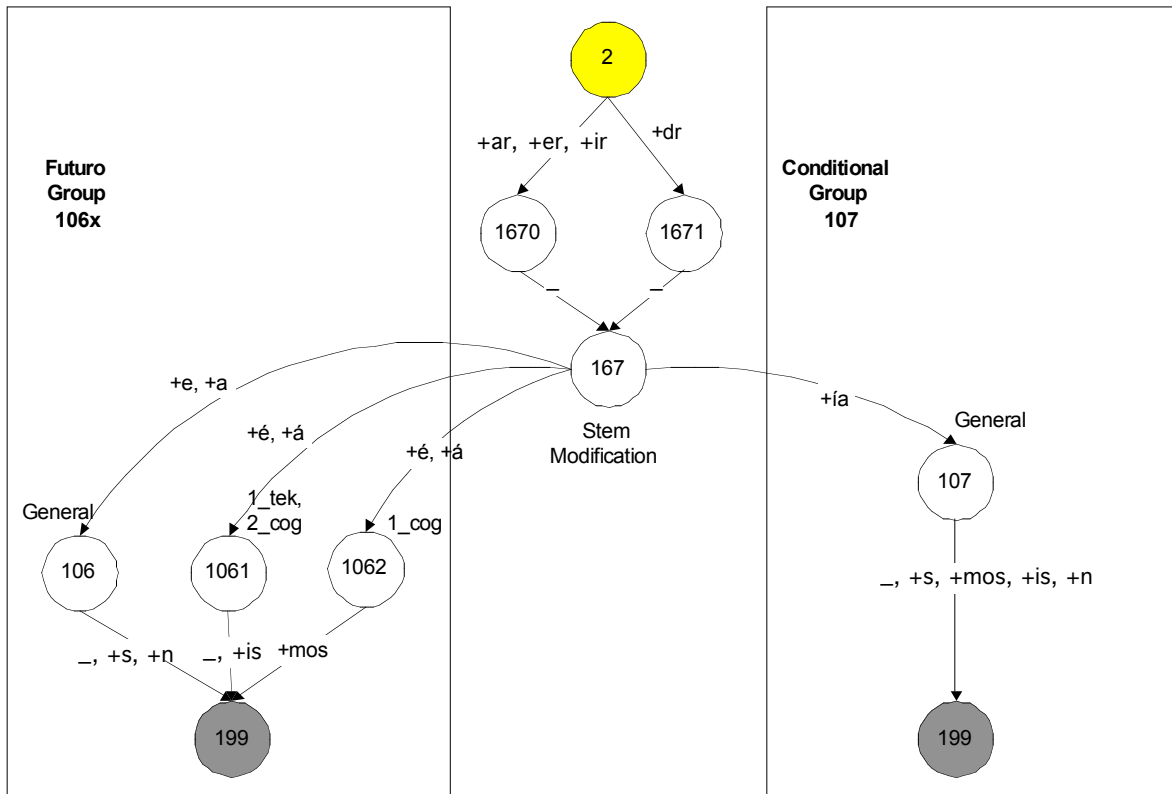


Figure 3.6. FSM for Future and Conditional Tenses

One of the well-defined areas of Artificial Intelligence is Knowledge Representation (KR) which deals with the formalisms and inference procedures needed to represent knowledge about a domain [15]. One of the most common KR formalisms in use is frames. Frames represent knowledge in terms of slots and values, where each slot expresses a specific piece of knowledge. We have used same formalism in Spanish and Turkish morpheme entries. For example, Spanish noun morpheme entries contain flags in addition to Turkish morphological noun parameters. Turkish nouns do not change according to sex information, masculine or feminine, but Spanish nouns differ according to sex. This information is entered not only for noun morpheme entries but also for adjective morpheme entries, too.

*esp_morph_entry('IsimKök',[[g,a,t,A],[type(noun),sem(X^cat(X)),mf(MF)]],_):-
change5(A,MF).*

Here mf (MF) represents the sex of the noun; possible values are m for masculine and f for feminine. In above sample, noun can be either masculine el gato (the male cat) or feminine la gata (the female cat). But there are nouns which are specific for only one sex, like la mesa (the table). Its entry is as follows:

esp_morph_entry('IsimKök',[[m,e,s,a],[type(noun),sem(X^table(X)),mf(f)]],_).

Similar to Turkish functions used for morphophonemic rules such as vtoa, ltod, etc., there are Spanish functions that are defined for Spanish morphophonemic rules; currently they are named as Change, Change5. They are used for masculine/feminine or singular/plural phoneme transformations.

3.3. Grammatical Categories in the Lexicon

The machine translation lexicon will give the information needed for syntactic and semantic processing: grammatical category (noun, verb, etc.), sub-categorization features (masculine or feminine noun), and semantic information.

Person (pronouns, verbs): Identifies speaker, addressee, third or fourth party. Sample values: first, second and third person.

Number (pronouns, nouns, verbs): Indicates the number of elements referred to. Sample values: singular, plural.

Gender (pronouns, nouns, verbs, adjectives): Relates to whether the object referred to is male, female or neutral (sexless).

Case (pronouns, nouns, adjectives): Indicates the role of a participant within a phrase; it distinguishes subjects, objects and various other roles. Sample values: nominative, accusative, dative, locative, ablative. The subject of the sentence is nominative; the object can be accusative, dative and other values.

Tense (verbs): Whether an action is performed in the future, present or past time.
Sample values: past, present, future tense.

4. SYNTAX LEVEL

In interlingua based MT application, the “Syntax Level” is the intermediate level between Morphological and Semantics levels. The sentences from the source language are broken into the constituents. Then every constituent is passed to the Semantics level where their semantic arguments are set. In the end, those arguments are combined together in order to build the interlingua representation of the whole sentence. In this Chapter the syntactic rules for Turkish and Spanish will be discussed.

4.1. The Main Syntax Rule

The most important difference between Turkish and Spanish syntax rules is the position of the verb in the sentence. Although, in an ordinary Turkish sentence the constituents are ordered in SOV (Subject – Object – Verb) format, like all other Latin based languages in Spanish they are ordered in SVO (Subject – Verb – Object) format. This major difference obviously affects the syntax rules to be written for these languages.

In both languages, the subject of the sentence comes first. In the syntax level, the ‘Number’ and ‘Person’ arguments of the Subject are set (see Chapter 3), so that they should match with the corresponding arguments of the ‘Verb’ of the sentence. Moreover, in both languages, the subject of the sentence should be in nominal case.

In our approach, the rest of the sentence (Object + Verb) is thought to be a “Verb Phrase” where the objects determine the verb in Time, Manner and Location aspects (they will be mentioned in Semantics Chapter). As a result, the distinction between Turkish and Spanish affects the syntax rule of the Verb Phrase instead of the main sentence rule.

Our main syntax rule is very simple for both of the languages: A sentence can contain a Subject, which is a noun phrase (will be discussed later) and a verb phrase which includes the objects and the verb of the sentence. And the Prolog representation is like the following:

$ss(Tense, PN, N, P, Sem) \rightarrow$
 $snp(_, nom, N, P, (X^{Sco})^{Sem}),$
 $svp(Tense, N, P, X^{Sco}, PN).$

In this representation, “nom” implies that the case argument of the Subject should be in nominal form. Also N (number) and P (person) arguments of noun and verb phrases should match. In general, this rule breaks the sentence into two as subject and the rest; then handles them separately.

Çetinoğlu [8] gives a general outline of Turkish grammar in terms of Phrase Structure Rules. In Table 4.1 main Turkish phrase structure rules are listed, most of the Turkish phrase structure rules are derived from previous studies [8, 9]. Spanish phrase structure rules are defined for syntax analysis. Table 4.2 lists the phrase structure rules for Spanish sentences.

Table 4.1 Turkish Phrase Structure Rules

TS	→	TNP	TVP
TNP	→	TD	TN
TNP	→	TADJP	TNP
TNP	→	TPN	
TNP	→	TCNP	
TNP	→	TNP _{genitive}	TNP
TVP	→	TV	
TVP	→	TOBJP	TV
TOBJP	→	TNP _{acc,dat,loc,abl}	TOBJP
TOBJP	→	TADV _{time}	TOBJP

Table 4.2 Spanish Phrase Structure Rules

S	→	NP _{subject}	VP _{verbal}	
S	→	NP _{subject}	VP _{nominal}	
S	→	VP _{verbal}		
S _{vocative}	→	NP _{vocative}	S	
NP _{subject}	→	NP _{nominative}		
NP _{subject}	→	PR		
NP _{subject}	→	NP _{complement}		
NP _{nominative}	→	n		
NP _{complement}	→	NP _{complement}	NP _{genitive}	
NP _{complement}	→	d	n	
NP _{genitive}	→	PREP _{de}	n	
NP _{accusative}	→	PREP _a	n	
NP _{dative}	→	PREP _a	n	
NP _{dative}	→	PREP _{para}	n	
NP _{ablative}	→	PREP	n	
NP _{vocative}	→	N _{comma}		
VP _{nominal}	→	VP _{auxiliary}	NP _{nominative}	
VP _{verbal}	→	VP _{transitive}	NP _{accusative}	
VP _{verbal}	→	VP _{transitive}	NP _{dative}	
VP _{verbal}	→	VP _{transitive}	NP _{accusative}	NP _{dative}
VP _{verbal}	→	v	NP _{complement}	
VP _{verbal}	→	v	NP _{ablative}	
VP _{verbal}	→	v		
VP _{verbal}	→	V _{imperative}	NP _{complement}	
VP _{transitive}	→	VP _{prep_a}		
VP _{transitive}	→	VP _{no_prep_a}		

Main elements of a Spanish sentence are subject and predicate. The subject is the element that does the action, which is stated by the predicate in the sentence.

4.2. Noun Phrases

4.2.1 Simple Noun Phrases

In Spanish, every noun should be used with a determiner like *el* (male the), *la* (female the), *un* (male a), *una* (female a), etc. except for proper nouns. So even a simple noun should be represented as a noun phrase, which consists of a determiner and a noun as follows:

$$\begin{aligned} \text{snp}(\text{Form}, \text{Case}, \text{N}, \text{P}, \text{Sem}) \text{ -->} \\ \text{sd}(\text{Form}, \text{N}, (\text{X}^{\wedge}\text{Res})^{\wedge}\text{Sem}, \text{MF}), \\ \text{sn}(\text{Form}, \text{Case}, \text{N}, \text{P}, \text{X}^{\wedge}\text{Res}, \text{MF}). \end{aligned}$$

Here, the N (Number) arguments of the determiner and the noun should match, so that if one of them is singular, the other one should be singular, too. This check prevents such cases like *uno perros* (a dogs) or *las mesa* (the table(s)).

In Spanish every determiner and noun has a “masculine” or “feminine” value. In our implementation, the MF (masculine/feminine) argument is also used for preventing a mismatch between the determiner and noun like *el mesa* (the (masculine)table) or *una perro* (a (feminine) dog).

In Turkish, the determiners and nouns do not have masculine or feminine values, so we do not need to use MF arguments for Turkish determiners and nouns.

It can also be observed that the ‘Case’ argument of the whole phrase is determined by the noun.

4.2.2 Prepositional Noun Phrases

Since Turkish is an agglutinative language, the case of a noun can be determined by morphological analysis of that word. On the other hand, Spanish does not have such a property. In Spanish, generally nouns are used with prepositions like *a* (to), *en* (in/on), *de* (from), etc. forming Prepositional Noun Phrases, like *a la ciudad* (to the city), *en la casa* (in the house) and *de la escuela* (from the school).

A simple Prepositional Noun Phrase (PNP) contains a preposition, the determiner of the noun and the noun itself. It is the preposition that determines the Case of the noun phrase. In implementation a PNP is represented as follows:

$$\begin{aligned} \text{snp}(\text{Form}, \text{Case}, \text{N}, \text{P}, \text{Sem}) \rightarrow & \\ & \text{sprep}(\text{Case}), \\ & \text{sd}(\text{Form}, \text{N}, (\text{X}^{\wedge}\text{Res})^{\wedge}\text{Sem}, \text{MF}), \\ & \text{sn}(\text{Form}, \text{Case}, \text{N}, \text{P}, \text{X}^{\wedge}\text{Res}, \text{MF}). \end{aligned}$$

In Spanish, the determiner “el” is merged with prepositions like *a* (to), *de* (from) forming single words *al* and *del* as in the sentence given below:

El hijo del (de + el) vecino irá al (a + el) cine mañana (The son of the neighbor will go to the cinema tomorrow).

The Spanish input sentences are preprocessed and these merged words are decomposed before the sentences are parsed. Similarly, when Spanish is used as the target language, the newly generated sentences are post-processed and corresponding the prepositions and determiners are merged.

4.2.3 Noun Phrases with Adjectives

Both Turkish and Spanish noun phrases can contain as many adjectives. Since the number of the adjectives in a noun phrase is not limited, these kinds of phrases are implemented in recursive manner.

But there is a basic distinction between Turkish and Spanish in this case. While in Turkish the adjectives are placed before the noun like *büyük siyah köpek* (the big black dog), in Spanish they come after the noun as, *el perro negro grande* (the big black dog). This minor difference requires a change in the order of the constituents in the syntax rules in our implementation.

4.2.4 Proper Noun Phrases

Proper nouns are special cases for both Turkish and Spanish languages. They do not have determiners, they are always singular and 3rd person; also their semantic

representations are different than common nouns. The syntactic rule for the proper nouns is given below:

$$\begin{aligned} \text{snp}(_, \text{nom}, \text{sing}, 3, \text{Sem}) \rightarrow & \\ & \text{sd}(\text{Form}, N, (X^{\wedge} \text{Res})^{\wedge} \text{Sem}, \text{MF}), \\ & \text{sprop}(\text{Form}, \text{Case}, N, P, \text{Sem}, \text{MF}). \end{aligned}$$

4.2.5 Definite Noun Phrases

In Turkish the definite noun phrases are formed of several numbers of ordinary noun phrases. Except for the last one, all the noun phrases in the definite noun phrase should be in “Genitive” case. The Case, number and person arguments of the last noun phrase determine the arguments of the whole definite noun phrase as in the example, *komşunun oğlunun arabasına* (to the car of the son of the neighbor).

On the other hand, in Spanish the syntax rule for the definite noun phrases is somewhat different. The “de” preposition should be located between the noun phrases forming the definite noun phrase. Also these noun phrases are placed in reverse order; so that it is the first phrase that decides the case, number and person quantities of the whole definite noun phrase. As an example, the phrase given above can be translated into Spanish as follows: *al coche del hijo del vecino* (to the car of the son of the neighbor)

Since the number of noun phrases in the definite noun phrase is unlimited, they are implemented in recursive manner:

$$\begin{aligned} \text{snpdef}(_, \text{Case}, N, P, (X2^{\wedge} \text{Pred2})^{\wedge} \text{Sem}) \rightarrow & \\ & \text{snp}(_, \text{nom}, _, P, (X2^{\wedge} \text{PredLast2})^{\wedge} \text{PredLast1}), \\ & [\text{de}], \\ & \text{snpdef}(_, \text{Case}, N, P, (X1^{\wedge} \text{Pred2})^{\wedge} \text{PredLast2}). \end{aligned}$$

4.3. Verb Phrases

Once the subject of the sentence is handled, the rest of the constituents are located in the verb phrase. These constituents can describe the verb in Time, Location and Manner aspects. Obviously the verb phrase also contains the verb itself besides these constituents.

In both languages, the constituents defining the verb (which are called Objects) are syntactically similar; but the main difference is the position of the verb among the constituents. In Turkish the words in a simple sentence are ordered as S-O-V format, while in Spanish they are ordered as S-V-O.

Besides the verb itself, the verb phrase contains at most four noun phrases. One of them is the Theme of the sentence if the verb is transitive. The Theme is usually in accusative case. The remaining three noun phrases define the verb in Location manners. They may have dative, locative and ablative cases. In a verb phrase, no two noun phrases can have the same case argument; so that if one of the noun phrases is in dative case, none of the others can have dative case argument.

Moreover in a verb phrase there can be adverbs defining the verb in Time and Manner aspects. In our implementation, we focused on the adverbs that have Time meaning (the Meaning will be discussed in Semantics Chapter); like *bugün* (today), *yarın* (tomorrow), *her zaman* (always), etc.

In Spanish the verb should be located before these constituents; but in Turkish, the verb is located in the end. Moreover, the constituents defining the verb can be in any order. For example, following verb phrases are all syntactically correct for Turkish:

Arabayı yarın şehirden alacak. (will pick the car tomorrow from the city)

Yarın şehirden arabayı alacak. (tomorrow will pick the car from the city)

Şehirden arabayı yarın alacak. (from the city will pick the car tomorrow)

The Spanish translations for these sentences are also syntactically correct. Therefore, in order to handle all kinds of orderings of the constituents, a recursive implementation technique is used. In our implementation, a verb phrase contains a verb and an “object phrase”:

```

svp(Tense,N,P,X^Pred,PN) -->
    sv(Tense,N,P,Sem2,PN),
    sobjp(X^Pred,Sem,Sco,Sem2).

```

Here the “object phrase” handles all of the constituents (noun phrases and adverbs) that describe the verb in several aspects. A simplified representation of the object phrase can be given as follows:

```

// The object can be a noun phrase with Location Information
sobjp(X^Pred,Sem,Sco,Semfinal)-->
    snp(_,Case,N,3,(Y^Scotemp)^Pred),
    sobjp(X^Pred2,Sem,Scotemp,Semfinal).

// The object can be an adverb with Time information
sobjp(X^Pred,Sem,Sco,Semfinal)-->
    sadvp(Semadvp),
    sobjp(X^Pred,Sem,Sco,Semfinal).

// Terminate the recursion
sobjp(X^Pred,Sem,Sco,Semfinal).

```

5. SEMANTICS LEVEL

Semantics is the top of the 3-level architecture used for the interlingua based MT technique we used in our implementation. After the semantics level is completed, the interlingua representation is available to be used in many applications like MT.

Semantics is also the hardest and the most complicated level in MT. The meaning of the sentence has to be represented correctly, in order to be used in further applications. Also many problems like ambiguities have to be handled and resolved in the semantics level.

5.1. First Order Predicate Calculus

In MT applications, First Order Predicate Calculus (FOPC) is most widely used technique. In FOPC, there are constants, variables, connectives and quantifiers. The connectives are logical connectives like and, or, if, iff, etc. There are two types of quantifiers used in FOPC, the existential and the universal quantifiers. An existentially quantified formula (\exists) is true iff at least one of the quantified variables makes the internal sub-formula true. For the universal quantified sentence (\forall) to be true, all the quantified variables makes the internal sub-formula must be true.

In FOPC definitions and relationships can be expressed. For example the corresponding expression for the sentence, 'X is cat' is $\text{cat}(X)$ in FOPC. There is no limit to this kind of definitions in FOPC. Also relationships can be defined by using n-ary predicates. As an example, the relation 'A loves B' can be expressed as $\text{loves}(A, B)$ by defining a binary "loves" predicate.

5.2. Lambda Calculus

Lambda Calculus is defined to represent properties of objects. According to Lambda calculus the previous definition, 'X is cat' can be written as:

$$(\lambda x)cat(x)$$

The relationships can also be expressed in Lambda calculus with n-ary predicates. As an example, the previous relationship can be defined as:

$$(\lambda b)(\lambda a)loves(a, b)$$

We used Prolog language in our MT implementations. The definitions and the relationships expressed in Lamda Calculus should be converted into Prolog format. But in Prolog, λ expression is not defined; instead, “ \wedge ” operator is used as follows:

$$B \wedge A \wedge loves(A, B)$$

5.3. Semantic Representation

Semantic formalization of Turkish sentences has been a subject matter for several studies done in universities in Turkey. In this thesis, in addition to these works, we defined several more clauses and phrases in Turkish and built semantic representations for them. In this chapter, only these additions will be mentioned.

On the other hand, the main aim of this thesis is bi-directional translation between Spanish and Turkish languages. In order to achieve this goal, syntactic rules for both languages are implemented. Additionally, semantic representation models are built for both parsing and generation of Spanish and Turkish sentences.

Furthermore, a major improvement done in this research is ambiguity resolution. Since there are several ambiguity and resolution types, we have chosen some of them and tried to resolve the ambiguity. The example cases and the resolution methods are discussed later in this chapter.

5.4. Parsing Stage

In order to be used in further applications, the interlingua representation should be formed correctly from the source sentence. Interlingua is created by the syntax rules of the source language, while the source sentence is being parsed. The Semantics arguments of the syntax rules are filled during the parse operation, and in the end, those arguments are merged to form the interlingua representation.

As discussed in the previous chapter, a simple Turkish sentence is in S-O-V format; so that a sentence starts with the subject and ends with the verb; and there are several objects describing the verb, between them. In this manner, the sentence can be divided into two major parts; the noun phrase (subject) and the verb phrase (objects + verb).

The verb phrase, which contains a number of objects and a verb, can be implemented as follows:

$$\begin{aligned}
 tvp(ST, Tense, PN, N, P, X^{\wedge}Pred) \rightarrow \\
 \quad tobjp(X^{\wedge}Pred, Sem, Sco, SemFinal), \\
 \quad tv(ST, Tense, PN, N, P, SemFinal).
 \end{aligned}$$

Here, during the parse operation, some part of the $X^{\wedge}Pred$ argument is filled by the subject of the sentence. This information is passed to the *object phrase* which will parse the objects in the sentence. The Object Phrase uses temporary *Sem* and *Sco* arguments which will be explained later. While the Object Phrase is parsing the objects of the sentence, it simultaneously fills the composite *SemFinal* argument which includes the *Pred* itself. If it returns correctly, this argument is passed to the Verb predicate, which parses the verb of the sentence, to set the Tense, Number and Person arguments. The verb predicate also fills the *SemFinal* argument and returns *Pred* as a result.

5.4.1 Verbs

In semantic representation of a sentence, the verbs play the most important role. It is the verb which contains the meaning of the sentence. The rest of the constituents are used as parameters for describing the verb.

In our work, the verbs in both Turkish and Spanish languages have the same representation. As an example, the verb ‘to talk’ is represented in the following way:

```
tr_morph_entry('FiilKök',[[k,o,n,u,ş],[type(verb),sem(Time^Abl^Loc^Dat^Theme^Agent^talk(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))]],_,_,_,u,ş,ok).
```

```
esp_morph_entry('FiilKök',[[h,a,b,l],[type(verb),sem(Time^Abl^Loc^Dat^Theme^Agent^talk(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))]],0,1,1).
```

```
...sem(Time^Abl^Loc^Dat^Theme^Agent^talk(EvNo,Agent,Theme,Dat,Loc,Abl,Time,[B,E,D,U],Tense,AuxTense,ST2))...
```

In this example, the verb ‘talk’ has 11 parameters and 6 of them will be filled by other constituents of the sentence during the parsing phase. These 6 parameters provide the verb, the information about the subject of the sentence (Agent), the Theme, the Location (Dat, Loc and Abl) and the Time. Obviously, any of these parameters can be empty, if they are not used in the input sentence.

The first parameter, EvNo keeps the event number, which will be used for inter sentence dependencies in future applications. The argument list [B,E,D,U] which is used for detailed time information, is also left blank for the same reason. The last 3 parameters are filled by the verb itself, with the help of the morphological analyzer. These are the Tense of the verb, the AuxTense (if any) and the Positive/Negative argument.

5.4.2 The Object Phrase

The objects are the constituents that define the verb in several aspects. In this work, the ones with Location and Time meanings are handled. Also because the number of objects in the sentence is unlimited, the object phrase predicate is implemented in recursive manner.

```

tobjp(X^Pred,Sem,Sco,Semfinal)-->
(
{ var(Pred)},
tnp(Case,(Y^Scotemp)^Pred,N,3),
{
( Case==acc,Sem=_ ^ _ ^ _ ^Y^X^Temp);
( Case==dat,Sem=_ ^ _ ^Y^ _ ^X^Temp);
( Case==loc,Sem=_ ^ _ ^Y^ _ ^ _ ^X^Temp);
( Case==abl,Sem=_ ^Y^ _ ^ _ ^ _ ^X^Temp)
},
{
Pred=[_ , _ , _ , Pred2]
},
tobjp(X^Pred2,Sem,Scotemp,Semfinal));

```

In this program code, the objects that define the verb in Location manner and the Theme of the sentence are handled. These objects are in fact noun phrases with their case arguments. The Location manner is divided into three groups: Dative, Locative and Ablative. The meaning of the sentence can be obtained from the Case value of a noun phrase. For example, if a noun phrase is in Dative case, the phrase exposes the destination of verb to be done, like the word *okula* (to school) in the sentence, *Ali okula gidiyor* (Ali is going to school).

In the Object Phrase predicate, firstly the noun phrase is parsed, and its Case argument is set. Then according to this Case value, that entity (Y) is set to be the corresponding argument of the overall semantics of the verb (Temp). Here X represents the subject of the sentence. After that allocation, the “Object Phrase” is called recursively with the new *Pred2* argument.

There may also be adverbs in the Object Phrase defining the verb in Time and Manner aspects. In this work, we are focused on the ones with Time meanings only. The following code handles such adverbs:

```
tobjp(X^Pred,Sem,Sco,Semfinal)-->
  (
    {var(Pred)},
    tadvp(Semadvp),
    {
      Sem=Semadvp^_^_^X^Temp
    },
    tobjp(X^Pred,Sem,Sco,Semfinal)
  );
```

If the next word to be parsed is not a noun phrase, the program will call this predicate. The *tadvp* predicate tries to parse the word as an adverb. If it succeeds, it fills the *Semadvp* argument, which holds the semantic information about the adverb. Then this information is passed to the overall semantic representation of the sentence. And in the end, the Object Phrase predicate is called recursively.

On the other hand, in Spanish, there is a SVO ordering of the words in a regular sentence. According to this ordering, the verb comes before the objects defining it. So in order to parse those sentences, the “Object Phrase” predicate should slightly be changed as follows;

```
svp(Tense,N,P,X^Pred,PN) -->
    sv(Tense,N,P,SemFinal,PN),
    sobjp(X^Pred,Sem,Sco,SemFinal).
```

In contrast to Turkish rule, in Spanish the verb predicate parses the verb of the sentence, fills the *SemFinal* argument and passes it to the “Object Phrase”. Then Object

Phrase fills the corresponding elements (Location, Time) of the SemFinal argument and returns the $X^{\wedge}Pred$; which is the final Semantic representation of the overall sentence.

5.5. Generation Phase

After the input sentence from the source language is parsed and the interlingua representation is formed, it will be used in generation phase for the target language. The syntax (and semantics) rules for the target language get the previously built interlingua representation as input and try to generate the sentence in the objective language.

During the generation phase, a very important problem is the ambiguity resolution. The interlingua built from the source language should be general and universal; so that it can be converted into real-life sentences in many target languages. For the moment, each target language has to have some limitation rules in the generation phase, in order to prevent incorrect translation and misunderstanding. The types of ambiguities and the methods to prevent such cases will be discussed later.

Turkish generation rules are slightly different than the ones for Spanish. This difference occurs again for the same reason; the word ordering. As it was discussed before, in Spanish, the verb comes before the objects defining it in an ordinary sentence. This situation makes life easier for the Spanish sentence generator. Because, the generator consumes the verb first, generates it, then it is faced with the objects. During the consumption of the objects, the generator reads the parameters of the verb one by one, and generates the corresponding objects accordingly.

Conversely, the Turkish generator does not have such an opportunity. It should generate the objects before the verb; since an ordinary and regular Turkish sentence has this kind of ordering. But the information about the objects to be generated is located in the semantic representation of the verb.

As a solution for this condition, the generator does a redundant (or fake) consumption of the verb. The verb is consumed, but in contrast, not generated. The

semantic information of the verb is acquired by the generator, to be used in generation of the objects. The verb is generated, after the generation of the objects defining it.

$$\begin{aligned}
 & tvp(ST, Tense, PN, N, P, X^{\wedge}Pred) \rightarrow \\
 & \quad \{ tv(ST, Tense, PN, N, P, Sem2, L1, L2) \}, \\
 & \quad tobjp(X^{\wedge}Pred, Sem, Sco, Sem2), \\
 & \quad tv(ST, Tense, PN, N, P, Sem2).
 \end{aligned}$$

As it was mentioned in syntax level, we used a recursive implementation technique. This approach is useful for handling unlimited number of objects. On the other hand, our recursive method does not prevent the word ordering among the objects. It forces the generator to generate the Theme first, then the objects about the Location, and the adverbs about Time in the end. Mainly, this bias does not affect the general meaning of the sentence, but it may violate the Mood of the verb.

In the example below, the Turkish sentence is translated into Spanish, but after this Spanish sentence is translated back into Turkish, the word ordering will be different from the original Turkish sentence

- Turkish Sentence: Turgut yarın şehirden arabayı getirecek. (Turgut will bring the car from the city tomorrow)
- Spanish Sentence: Turgut traerá el coche de la ciudad mañana.
- Regenerated Turkish Sentence: Turgut arabayı şehirden yarın getirecek.

5.6. Ambiguity Resolution

In semantic formalization of the sentences, one of the most challenging jobs to do is ‘Ambiguity Resolution’. There are several ambiguity types and some methods for resolving these ambiguities.

Disambiguation, the task of selecting the correct interpretation of a text, is one of the most important yet hardest problems in NLP and MT. For MT, it was already seen by Bar-Hillel [18] as a major stumbling block for unrestricted texts.

There are many sources of ambiguity in MT and their resolution can take place at various stages of processing. Disambiguation during analysis involves selecting the appropriate syntactic structure and semantic interpretation of the input text. This process requires monolingual information mainly. In NLP this stage may also involve scope resolution, anaphora and definite reference disambiguation and ellipsis resolution, as well as a host of other problems. However, they are not considered here mainly.

Ambiguities can be observed under two topics; Lexical and Structural Ambiguities.

5.6.1 Lexical Ambiguities

Some of the morphological problems discussed above involve ambiguity, that is to say, there are potentially two or more ways in which a word can be analyzed. There are lexical ambiguities, where one word can be interpreted in more than one way. Lexical ambiguities are of three basic types: category ambiguities, homographs and polysemes, and transfer (or translational) ambiguities.

5.6.1.1 Category Ambiguity

The most straightforward type of lexical ambiguity is that of category ambiguity: a given word may be assigned to more than one grammatical or syntactic category (e.g., verb or adjective) according to the context. These category ambiguities can often be resolved by morphological inflection.

For example, in Turkish there is a highly ambiguous word “*yüz*” It can be used as a verb (to swim, to skin), as a noun (face), or as an adjective (hundred). Our morphological analyzer can distinguish a verb, a noun and an adjective; and our syntactic rules can resolve these kinds of category ambiguity that are caused by the word “*yüz*”.

Yüz çocuk denizde yüzdü (A hundred children swam in the sea).

In this sentence the first *yüz* is used as an adjective in the noun phrase “*yüz çocuk*” (a hundred children) forming the subject of the sentence. This is because; it is forced to be the subject by the syntax rule that parses the sentence. Then the second “*yüz*” is used as the verb of the sentence, because it gets a suffix *-dü* which can only be added to the verbs. The morphological analyzer returns as the decision verb; and the corresponding syntax rule confirms it.

However, the problems increase when several categorically ambiguous words occur in the same sentence, each requiring to be resolved syntactically.

5.6.1.2 Homography and Polysemy

The second type of lexical ambiguity occurs when a word can have two or more different meanings. Linguists distinguish between homographs, homophones and polysemes. Homographs are two (or more) words with quite different meanings which have the same spelling: examples are club (weapon and social gathering), bank (riverside and financial institution), and light (not dark and not heavy). Homophones are words which are pronounced the same but spelled differently (e.g. hair and hare), but since MT is concerned essentially with written texts, they need not concern us here. Polysemes are words which exhibit a range of meaning related in some way to each other, e.g. (mouth of a river, channel of communication, etc.).

In this thesis, we focused on resolving ambiguities caused by homographs. There are a number of methods to cope with the homographs such as building semantic networks, using probabilistic approaches, etc. We tried to resolve the true meaning of the homographs with morphological and syntactic analysis.

For instance, in the previous example, the word “*yüz*” has two different meanings even if it is used as a verb (to swim and to skin). If the word is used in “*to skin*” meaning, then the verb should be transitive; so that there can be a Theme in the sentence. Conversely, if it is used in “*to swim*” meaning, then the verb becomes intransitive, so there can not be any Themes in the sentence. As a result, if the verb of the sentence is the word

yüz and there is a Theme in the sentence, it can be concluded that the meaning of the verb is to skin, otherwise its meaning is to swim. Following are two example sentences:

Adam koyunu yüzdü (The man skinned the sheep).

Adam denizde yüzdü (The man swam in the sea).

In the first sentence, there is a Theme, koyun (the sheep), so the verb yüzdü is recognized as to skin, and in the second sentence, since there are not any Themes, the verb is recognized as to swim. Once the correct interlingua representation for the verb is created, it can properly be translated to Spanish.

In fact, in the previous example, the argument, “since there are not any Themes, the verb is recognized as to swim” is somewhat defective; because without any Themes, the verb may also have the meaning to skin, as in the sentence:

Adam bahçede yüzdü (The man skinned (something) in the garden).

But this kind of ambiguity resolution needs more sophisticated techniques like semantic networks. The action of “swimming” can be done in “swimmable” places such as sea, lake, pool, etc. But in this example, the action is done in the garden which is not a “swimmable” place; so the verb should be recognized as “to skin”.

Moreover, the word “çal” also causes a similar kind of ambiguity with meanings to play and to steal. The difference from the previous example is that, with both meanings the verb is transitive. Therefore, searching for a Theme will not be enough for resolving the ambiguity. The method to be used here can be clarified by examining the following sentences:

Ali piyano çaldı (Ali played the piano).

Ali piyanoyu çaldı (Ali stole the piano).

If the Themes of the sentences are examined carefully, it can be observed that the Theme of the first sentence is in nominative case, while the Theme of the second sentence

is in accusative case. So this type of ambiguity is resolved by examining the case of the Theme.

Although the methods for resolving ambiguity used in this work look like case specific; they can easily be generalized for other homographic verbs in Turkish.

5.6.1.3 Transfer Ambiguity

Category ambiguities, homography and polysemy are all examples of lexical ambiguities which cause problems primarily in the analysis of the source language. In MT there are also transfer ambiguities which arise when a single source language word can potentially be translated by a number of different target language words or expressions. The source language word itself is not ambiguous, or rather it is not perceived by native speakers of the language to be ambiguous; it is 'ambiguous' only from the perspective of another language. It is, therefore, a problem of translation (certainly a difficult one) but not a problem of linguistic analysis.

5.6.2 Structural Ambiguity

Whereas lexical ambiguities involve problems of analyzing individual words and transferring their meanings, structural ambiguity involves problems with the syntactic structures and representations of sentences. Ambiguity arises when there is more than one way of analyzing the underlying structure of a sentence according to the grammar used in the system. The qualification is a reminder that no parser can go beyond the limitations of the grammar being implemented. If the grammar does not make distinctions which a human reader would make, then the parser will not be able to decide between alternative analyses. It is consequently valid to distinguish between real ambiguities, for which a human might find several interpretations, and system ambiguities which human readers would not necessarily recognize.

6. APPLICATION and RESULTS

The main application developed for this thesis translates input sentences either in Spanish or Turkish to the target language. The program asks the user to indicate the file, reads it, and outputs the translation of the sentence respectively.

Some sample outputs for several Turkish and Spanish sentences are given here. The first sentence, given here is the input for the program, and the second sentence is the intermediate representation of the sentence and the last one is the translated output for that input.

- Turkish Sentence: Turgut yarın şehirden arabayı getirecek (Turgut will bring the car from the city tomorrow).
- Interlingua Representation: proper(turgut, turgut, singthe(abl, city(abl), singthe(acc, car(acc), bring(_G1499, turgut, acc, _G887, _G884, abl, tomorrow, [_G1511, _G1514, _G1517, _G1520], future, none, pos))))
- Spanish Sentence: Turgut traerá el coche de la ciudad mañana.

This sentence contains 3 different objects, *yarın* (tomorrow), *şehirden* (from the city), *arabayı* (the car). As it was mentioned in Chapter 5, in Generation phase, our recursive approach generates the Theme first, then the Location arguments and the adverb in the end. Most of the time, this approach does not cause any loss of meaning (especially in generating the corresponding Spanish sentence). But if we try to get the original Turkish sentence, by using that Spanish sentence as input, we will get the same sentence with a different word ordering.

- Spanish Sentence: Turgut traerá el coche de la ciudad mañana.
- Interlingua Representation: proper(turgut, turgut, singthe(abl, city(abl), singthe(acc, car(acc), bring(_G1499, turgut, acc, _G887, _G884, abl, tomorrow, [_G1511, _G1514, _G1517, _G1520], future, none, pos))))
- Turkish Sentence: Turgut arabayı şehirden yarın getirecek.

In the next example, the subject of the sentence is a definite noun phrase:

- Turkish Sentence: Komşunun oğlu yakında askere gidecek (The neighbor's son will join the army soon).
- Interlingua Representation: singthe(_G493, neighbor(_G493), singthe(_G487, (son(_G487), owns(_G493, _G487))), singthe(dat, military(dat), go(_G1364, _G487, _G1022, dat, _G1016, _G1013, soon, [_G1376, _G1379, _G1382, _G1385], future, none, pos))))
- Spanish Sentence: El hijo del vecino irá a la mili pronto.

The sentences can contain noun phrases with conjunctions:

- Turkish Sentence: Kediler ve köpekler evden bahçeye koşular (The dogs and the cats ran from the house to the garden).
- Interlingua Representation: andconj(([_G669, _G672], (plurthe(_G669, cat(_G669), singthe(abl, house(abl), singthe(dat, garden(dat), run(_G1669, [_G669, _G672], _G1294, dat, _G1288, abl, _G1282, [_G1681, _G1684, _G1687, _G1690], definite_past, none, pos))))), plurthe(_G672, dog(_G672), singthe(abl, house(abl), singthe(dat, garden(dat), run(_G1669, [_G669, _G672], _G1294, dat, _G1288, abl, _G1282, [_G1681, _G1684, _G1687, _G1690], definite_past, none, pos))))), singthe(abl, house(abl), singthe(dat, garden(dat), run(_G1669, [_G669, _G672], _G1294, dat, _G1288, abl, _G1282, [_G1681, _G1684, _G1687, _G1690], definite_past, none, pos))))
- Spanish Sentence: Los perros y los gatos corrieron al jardín de la casa.

A Spanish sentence can also be used as input for the translator:

- Spanish Sentence: Ali habla la verdad siempre (Ali always tells the truth).
- Interlingua Representation: proper(ali, ali, singthe(nom, truth(nom), talk(_G659, ali, nom, _G650, _G647, _G644, always, [_G671, _G674, _G677, _G680], _G667, _G668, _G669)))
- Turkish Sentence: Ali doğruyu her zaman söyler (Ali tells the truth always).
- Spanish Sentence: Todos estudiantas hablan Ingles (All of the students speak English).

- Interlingua Representation. plurall(_G499, student(_G499), proper(english, english, talk(_G721, _G499, nom, _G712, _G709, _G706, _G703, [_G733, _G736, _G739, _G742], _G729, _G730, _G731)))
- Turkish Sentence: Bütün öğrenciler İngilizce konuşur.

6.1. Ambiguity Resolution

As it was mentioned in Chapter 5, the following Turkish sentences involve several types of ambiguities; the corresponding Spanish sentences are also given below;

- Turkish Sentence: Yüz çocuk denizde yüzdü (A hundred children swam in the sea).
- Interlingua Representation: plurthe(_G421, [hundred(_G421), child(_G421)], singthe(loc, sea(loc), swim(_G987, _G421, _G831, _G828, loc, _G822, _G819, [_G999, _G1002, _G1005, _G1008], definite_past, none, pos)))
- Spanish Sentence: Los ciento niños nadaron en el mar.
- Turkish Sentence: Adam koyunu yüzdü (The man skinned the sheep).
- Interlingua Representation: singthe(_G361, man(_G361), singthe(acc, sheep(acc), skin(_G819, _G361, acc, _G687, _G684, _G681, _G678, [_G831, _G834, _G837, _G840], definite_past, none, pos)))
- Spanish Sentence: El hombre desolló el carnero.
- Turkish Sentence: Adam denizde yüzdü (The man swam in the sea).
- Interlingua Representation: singthe(_G367, man(_G367), singthe(loc, sea(loc), swim(_G837, _G367, _G708, _G705, loc, _G699, _G696, [_G849, _G852, _G855, _G858], definite_past, none, pos)))
- Spanish Sentence: El hombre nadó en el mar.
- Turkish Sentence: Adam bahçede yüzdü (The man skinned (something) in the garden).
- Interlingua Representation 1: singthe(_G367, man(_G367), singthe(loc, garden(loc), skin(_G837, _G367, _G708, _G705, loc, _G699, _G696, [_G849, _G852, _G855, _G858], definite_past, none, pos)))

- Spanish Sentence: El hombre desolló en el jardin.
- Interlingua Representation 2: singthe(_G364, man(_G364), singthe(loc, garden(loc), swim(_G834, _G364, _G705, _G702, loc, _G696, _G693, [_G846, _G849, _G852, _G855], definite_past, none, pos)))
- Spanish Sentence: El hombre nadó en el jardin.

- Turkish Sentence: Ali piyano çaldı (Ali played the piano).
- Interlingua Representation: special(ali, ali, singthe(nom, piano(nom), play(_G795, ali, nom, _G663, _G660, _G657, _G654, [_G807, _G810, _G813, _G816], definite_past, none, pos)))
- Spanish Sentence: Ali tocó el piano.

- Turkish Sentence: Ali piyanoyu çaldı (Ali stole the piano).
- Interlingua Representation: proper(ali, ali, singthe(acc, piano(acc), steal(_G848, ali, acc, _G716, _G713, _G710, _G707, [_G860, _G863, _G866, _G869], definite_past, none, pos)))
- Spanish Sentence: Ali robó el piano.

There are also several ambiguity categories that cannot be resolved in this study. For example, the following sentence has a homograph that can not be resolved according to our approach:

- Turkish Sentence: Çocukların yüzü bahçede oynuyor (A hundred of the children are playing in the garden).

In this sentence the Turkish word *yüz* can have two different meanings even in noun form; as *face* and *a hundred*. Because the *faces* of the children can not play; the word *yüz* should be translated as *a hundred*.

Moreover, transfer ambiguities are not resolved in this thesis. Such a sentence in Turkish is incorrectly translated into Spanish:

- Turkish Sentence: Seni düşünüyorum. (I am thinking about you)

- Spanish Sentence: Pienso te.

But the correct translation should be;

- Spanish Sentence: Pienso en ti.

Where the verb *pensar* (to think) in Spanish must be used with the preposition *en* (in) in order to give the correct meaning. However, in Turkish there is not such a requirement.

7. CONCLUSION

In this thesis, we have presented an interlingua based bidirectional machine translation system between Turkish and Spanish. In order to achieve this goal, Spanish verb conjugation with a great variety of morphological combinations, syntax rules for both languages are examined. Additionally, semantic representation models are built for both parsing and generation of Spanish and Turkish sentences. We take the advantage of bidirectional property of Prolog during analysis and generation phases of the program. The intermediate representation for the source language can easily be used for other Latin languages structurally similar to Spanish.

This thesis is built on top of several previous studies done in morphology and syntax levels, in Boğaziçi University. Especially, the syntactic and the semantic approaches of Çetinoğlu's [8] and Kardeş's [9] researches are used.

After morphological analysis, the words that are grouped in noun, adjective, and verb are investigated according to their functionality in the sentences. Spanish phrase structure rules are defined for syntax analysis. Furthermore, a major improvement was achieved in ambiguity resolution.

Demand for MT has grown and will continue to grow steadily. The huge Internet usage, great number of electronic communication and the use of computers have increased the demand for automatic translation of sufficient quality for determining the content of a Web page. Browsers with built-in MT capabilities will become commonplace in near future. Users can search remote databases containing abstracts in foreign languages and request translations of the abstracts or the full documents into their own language.

7.1. Future Work

7.1.1 Complex Sentences

The coverage and accuracy of this machine translation system can be extended. At the moment, our bidirectional MT system can translate about 50 Turkish and Spanish sentences which is a small number for a professional application. The word order variations can be controlled in a better way, because currently the sentences are generated in a default order in Turkish. Also, the structure of the source arguments needs to be modified for passive sentences. For example, the objects in the passive sentences should be mapped to the subject position during transfer. This can be achieved by adding Voice information to the lexicon. Voice expresses the relation of the subject or other participants of the verb to the action expressed. Typical values are active, passive, middle, causative.

The current Spanish dictionary has 48 nouns, 27 verbs, and 16 adjectives. This study can also be used for Spanish derivative and compounds as well, but it has not been done yet, since further linguistic analysis must be done to specify the features needed.

7.1.2 Incorrect Input Translation

All and only correct forms are analyzed and generated. When an incorrect source text is received, the system should be modified to generate partial translations, leaving some parts of the output untouched to be translated by the user later.

Not only post-editing, but also interactive system can be implemented. By using *Human-Aided Machine Translation* (HAMT) techniques, the system user can choose to interactively guide the system during the analysis. Either the most plausible representation for the source sentence is translated, or all alternative parses are translated enabling the user to choose among them. The translations are then post edited to correct any errors in word order, choice of lexical item or inflection.

7.1.3 Specific Domain

The argument of the need to incorporate real world and contextual knowledge into a machine translation system for the true understanding of input does not hold true for restricted technical domains since the anaphors and ambiguous words/phrases are rare in technical writing.

A transfer system tends to be less costly to build for a pair of languages, but in multilingual environments the costs quickly multiply as transfer modules need to be updated. Interlingua systems would be particularly useful for translation into multiple languages for restricted domains since unrestricted domains are too difficult to treat with this approach. Recently, a combination of different strategies within a single system is a growing area of research and development.

Machine Translation is one of the most challenging research activities, involving the application of complex theoretical knowledge. The main problems of machine translation are not related with computational technology but with language, meaning, understanding, and the social and cultural differences of human communication.

REFERENCES

1. Nirenburg, S., J. Carbonell, M. Tomita and K. Goodman. Machine Translation: A Knowledge-based Approach, Morgan Kaufmann Publishers, San Mateo, California, 1992
2. Hutchins, W. John and Harold L. Somers. An Introduction to Machine Translation, Academic Press, London, 1992.
3. Buchmann, Beat, "Early History of Machine Translation." In Margaret King, editor, Machine Translation Today: The State of the Art, Edinburgh University Press, Edinburgh, 1987, pp. 3-21.
4. Turhan, Çiğdem K., Structural Transfer in An English To Turkish Machine Translation System. Ph.D. Thesis, The Middle East Technical University, 1998
5. Hamzaoğlu, I., Machine Translation from Turkish to Other Turkish Languages and an Implementation for the Azeri Language, M.S. Thesis, Boğaziçi University, 1993.
6. Hakkani, Dilek Z., Gökhan Tür, Teruko Mitamura, Eric H. Nyberg, 3rd, and Kemal Oflazer. Issues in Generating Turkish from Interlingua, CMU-CMT-97-152, Sept. 1997.
7. Nyberg, E. H., 3rd and T. Mitamura, The KANT System; Fast, Accurate, High-Quality Translation in Practical Domainsö In Proceedings of COLING'92, Nantes, France, July, 1992.
8. Çetinoğlu, Ö., A Prolog Based Natural Language Processing Infrastructure for Turkish, M.S. Thesis, Boğaziçi University, 2001.
9. Kardeş, O., Forming Semantics of Turkish Texts In An Application Area, M.S. Thesis, Boğaziçi University, 2002. Goñi, J.M. and González, J.C. A framework for lexical representation. Proceedings of AI'95: Fifteenth International Conference. Language Engineering `95, pp. 243--252. Montpellier, June 27-30, 1995.

10. González, J.C., J.M. Goñi and A.F. Nieto, ARIES: a ready for use platform for engineering Spanish-processing tools, Language Engineering Convention, London, October, 1995.
11. Villena, J., B. González, J.C. González, M. Muriel, STILUS: Sistema de Revisión Lingüística de Textos en Castellano, Unpublished Manuscript, Iberamia, 2002.
12. Oflazer, K., "Two-level Description of Turkish Morphology", The Second Turkish Symposium on Artificial Intelligence and Neural Networks, Ankara, pp 86-93, 1993.
13. Güngör, T., Computer Processing of Turkish: Morphological and Lexical Investigation, Ph. D. Thesis, Boğaziçi University, 1995
14. Moreno, A. and Goñi, J.M., GRAMPAL: A morphological model and processor for Spanish implemented in Prolog, 1995 Joint Conference on Declarative Programming (GULP-PRODE'95), Marina di Vietri (Salerno, Italy), September, 1995.
15. Trujillo, A., Translation Engines: Techniques for Machine Translation, Springer-Verlag London Limited, London, 1999.
16. Verbix Inc., Verbix conjugate verbs in 30+ languages,
<http://www.verbix.com/languages/spanish.shtml>, 2002
17. Wanadoo Inc, Verbos Irregulares,
<http://perso.wanadoo.es/verbos/irregulares/01.htm>, 2002
18. Bar-Hillel, Y., *The present status of automatic translation of languages*, In F.L. Alt (Ed.), *Advances in Computers*, Volume I, pp.91-163, New York: Academic Press, 1960.

REFERENCES NOT CITED

Banguođlu, T., Türkçenin Grameri, Türk Tarih Kurumu Basım Evi, Ankara, 1990

Bratko, I., Prolog Programming for Artificial Intelligence, Addison Wesley, 2001

Castro, F., Uso de la Gramática Española-nivel intermedio-, Edelsa Grupo Didascalia, S.A., Madrid, 2001

Covington, M. A., Natural Language Programming for Prolog Programmers, Prentice Hall, New Jersey, 1994.

Hermoso, A. G., J.R. Cuenol, M. S. Alfaro, Gramática de Español Lengua Extranjera, Edelsa Grupo Didascalia, S.A., Madrid, 1995

Kut, İ., İspanyol Dili ve Grameri, İnkilap Kitabevi, İstanbul, 1996

APPENDIX A: SPANISH VERB TYPES

Spanish verbs are divided into regular and irregular verbs. Most of the verbs are regular. The irregular verbs are showing some similarities, such as vowel and consonant changes in stem in different tenses, changes in orthographic rules, etc. All the regular and irregular verbs are also divided into conjugation groups according to the ending in the infinitive.

- 1st conjugation (I), verbs ending in -ar : hablar (to talk).
- 2nd conjugation (II), verbs ending in -er : comer (to eat).
- 3rd conjugation (III), verbs ending in -ir : vivir (to live).

The verb irregularities below are obtained from Verbix [16] and Wanadoo [17] irregular verb lists.

A.1. Irregularities in Present Tense

A.1.1. Verbs with diphthongue

- E > IE: calentar (I), perder (II), discernir (III)
- I > IE: adquirir (III)
- O > UE: contar (I), volver (II)
- U > UE: jugar (I)
- E > IE/I: sentir (III)
- O > UE/U: dormir (III)

A.1.2. Change in the stem vocale

- E > I: pedir (III), erguir (III)
- O > U: podrir (III)

A.1.3. Consonants are added in the stem

- C > ZC: conocer (II)
- N > NG: tener (II)
- L > LG: valer (II), salir (III)
- S > SG: asir (III)
- U > UY: huir (III)
- G added: caer (II), oír (III)

A.1.4. Other irregularities

- B > Y in subjunctive: haber (II)
- E > Y in subjunctive: roer (II)
- C > G: hacer (II)
- AB > EP: caber (II), saber (II)
- EC > IG: decir (III)
- C > Z/ZG: yacer (II)

A.2. Irregularities in Preterite Tense

A.2.1. Change in stem vocale

- E > I: gemir (III)
- O > U: dormir (III)

A.2.1. Strong preterites

- 1st conjugation: andar, dar, estar

- 2nd conjugation: caber, haber, hacer, placer, poder, poner, querer, responder, saber, ser, tener, traer, ver , absolver
- 3rd conjugation: conducir, decir, ir, venir, abrir, cubrir, escribir, imprimir, morir

A.3. Irregularities in Future and Conditional Tenses

A.3.1. Loss of the protonic vocal

- saber (II)

A.3.2. Loss of vocal and consonant

- 2nd conjugation: hacer
- 3rd conjugation: decir

A.3.3. Loss of vocal and addition of consonant

- 2nd conjugation: poner
- 3rd conjugation: salir

A.4. Orthographic Modifications for Preserving the Pronunciations

- C > QU: buscar (I)
- G > GU: jugar (I)
- GU > GÜ: averiguar (I)
- Z > C: cruzar (I)
- C > Z: vencer (II), esparcir (III)
- G > J: coger (II), dirigir (III)
- GU > G: seguir (III)
- QU > C: delinquir (III)

A.5. Changes due to Orthographic Rules

A.5.1. Loss of atonic i

- bullir (III), bruñir (III)

A.5.2. Atonic i changes to y

- caer (II)
- oír (III)
- huir (III)

A.6. Change of the Orthographic Accent in Verbs Ending in -iar and -uar

- confiar (I)
- continuar (I)

A.7. Other changes of the orthographic accent

- U > Ú : aullar (I)
- I > Í : airar (I)