SYNTACTIC ANALYSIS IN MACHINE TRANSLATION

Petr Homola

ÚSTAV FORMÁLNÍ
A APLIKOVANÉ LINGVISTIKY

STUDIES IN COMPUTATIONAL AND THEORETICAL LINGUISTICS

Petr Homola

SYNTACTIC ANALYSIS IN MACHINE TRANSLATION

Published by Institute of Formal and Applied Linguistics
as the 6th publication in the series
Studies in Computational and Theoretical Linguistics.

This book has been printed with the support of the project MSM0021620838 of The Ministry of Education of the Czech Republic.

ISBN 978-80-904175-7-1

Contents

List of Figures

List of Tables

Acknowledgement

I am indebted to my supervisor Vladislav Kuboň for his guidance and support. I am also indebted to many reviewers—native speakers of the languages I have researched—for their feedback and help with evaluation. Finally, I would like to thank to everyone who helped me during my Ph.D. studies.

1
Introduction

Natural language processing (NLP) is a comparatively new and rapidly growing discipline in the borderland of theoretical linguistics on the one side and applied mathematics, especially graph theory and statistics, on the other. Machine translation (MT) is a kind of king's discipline of NLP and there has been long and extensive research in the area of rule-based formalisms as well as of statistical approaches to MT. One subcategory of MT is the translation between related languages which is being researched since the late 1980's of the 20th century. This thesis focuses on MT among Balto-Slavic languages.

1.1 The Significance of Machine Translation

The goal of machine translation is to automatically transfer a discourse (in MT usually in written form) from a source language to a target language while preserving its meaning and stylistic characteristics. When building an MT system, a natural requirement is to develop it with as little effort as possible. As the complexity of an MT system depends on the similarity of the source and the target language, the knowledge of different strategies for various degrees of language similarity can minimize the effort and guarantee an acceptable quality.

We mainly focus on Baltic and Slavic languages although most of the discussed aspects are valid in general. The mentioned language family has been chosen since it is an ideal 'playground' due to its typology and different degrees of similarity which allows to investigate MT among related languages in detail. Moreover, for many of these languages linguistic resources (such as morphological analyzers, synthesizers, taggers, corpora etc.) are available, thus it is comparatively easy to perform practical experiments to approve or falsify theoretical hypotheses. Also, the typology of these languages, mainly the extremely free word order at the level of actants, is very interesting from the viewpoint of formal theories as it cannot be directly processed by means of formalisms based on context-free rules. Last but not least, the importance of MT among these languages has grown since the accession of several Baltic and Slavic nations to the European Union.

It is obvious that MT between related languages is generally easier than between, for example, Guaraní and Georgian, but what is still unclear is what we have to focus on in the complex MT process so that we can effectively maximize the translation quality. This thesis attempts to explore the contribution of syntactic analysis to the MT in the context of the Balto-Slavic language family, and our additional experiments

with another language group, Romance, show that most of the conclusions are valid not only for Baltic and Slavic.

1.2 Structure of the Thesis

The thesis can be roughly split into three parts. Chapters 2, 3 and 4 define basic notions, give an overview of Baltic and Slavic languages and review older MT systems for related languages. Chapters 5, 6, 7 and 8 focus on the properties of the researched languages and on the implementation of an MT framework for them. Finally, Chapter 9 is dedicated to the statistical part of the framework, the ranker, and to the evaluation of our experiments.

There are many approaches to rule-based NLP such as the categorial grammar, HPSG, LFG etc. Our framework is loosely based on the Lexical Functional Grammar and on the theory behind the Prague linguistic school, which is described, along with the other used theoretical background, in Chapter 2. In Chapter 3, we give an overview of Baltic and Slavic languages and present the most notable facts about them. Chapter 4 gives a brief overview of MT systems for related languages that have been developed in the last decades.

In Chapter 5, we focus on the relationship between Balto-Slavic languages and we identify the various free-rides as well as substantial differences among them which are crucial for MT and NLP in general. Chapter 6 focuses on the most important syntactic features of the Balto-Slavic languages at the shallow and deep level. Chapters 7 and 8 describe the implementation of the partial parser and shallow transfer, respectively.

Chapter 9 is dedicated to the statistical ranker which is crucial to the framework since it is the only module that deals with the non-determinism of all other modules of the framework. Furthermore, we use the most notable methods of automatic evaluation of translation quality to evaluate our framework and to compare it to the shallow-transfer based MT system Apertium.

The concluding chapter provides a broader perspective on the problematics of MT between related languages and summarizes the contribution of the thesis to this particular area of NLP.

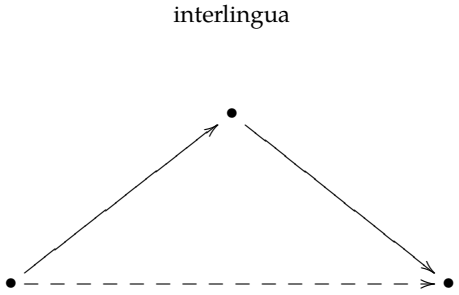
2

Basic Notions and Notation

This introductory chapter defines some basic notions used within the thesis. The concepts and terminology are loosely based on the Lexical Functional Grammar (LFG) and on the formalism used in the Prague Dependency Treebank (PDT) which is described in detail by Hajič et al. (2001) which in turn builds on the Functional Generative Description (FGD) proposed by Sgall et al. (1986), with affinities to the naturalness theory at the level of syntax and morphosyntax as defined by Mayerthaler et al. (1998).

2.1 Typical Scheme of Machine Translation

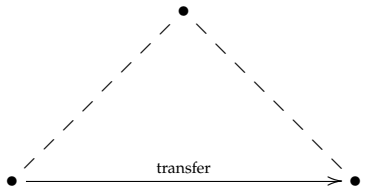
Most MT systems consist of three subsequent phases: analysis, transfer and synthesis. In the first phase, the input is analyzed and an abstract representation of it is produced. The concrete shape of the representation can vary. In the transfer phase, the abstract representation is adapted to the target language and finally, the translation is generated (synthesized) out of the abstract representation. The MT architecture with a hypothetical interlingua can be schematized by the so-called Vauquois' triangle:
(2.1)



The vertical axis represents the abstractness of the intermediate representation with the interlingua being the most abstract language independent representation.

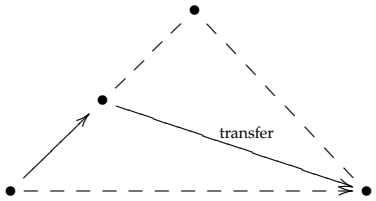
The original system *Česílko* which has neither parser nor transfer (except for the lexical one) could be schematized as follows:

(2.2)



In our system, we use a less abstract representation (at the language specific shallow syntactic level). Moreover, the transfer is recursively combined with synthesis, which can be schematized as follows:

(2.3)



The recursivity of the synthesis is given by the recursive character of the abstract representation—the feature structures. The transfer phase is described in detail in Chapter 8.

2.2 Linguistic Levels

The FGD in its original form is a stratificational formalism with five levels of linguistic description. At each level, there are two types of elements: elementary and complex; the complex elements consist of the elementary ones (the relation of composition). Between each two adjacent levels, there is the relation of realization. An element at a level, representing a function, is realized by one or more elements at the inferior level and vice versa, each element at a level, being a form, corresponds to one or more elements at the superior level. There are five levels:

tectogrammatical deep syntax, widely language independent, expressing the grammatical meaning of sentences

analytical surface syntax, language specific, reflecting the linearized representation of sentences

morphemic level of (complex) morphemes and (elementary) morphonemes

morphonological level of (complex) morphonemes and (elementary) phonemes

phonetical level of (complex) phonemes and (elementary) distinctive features

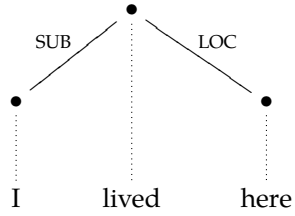
The relation of realization is a relationship between form and function at all levels of linguistic description (Panevová (1980) gives a detailed description and examples for various linguistic levels). For machine translation, the two highest levels—analytical and tectogrammatical—are of special interest.

2.2.1 Analytical Level (surface syntax)

At the analytical level, the sentence is represented by a syntactic tree where each node corresponds to exactly one word in the sentence. The edges of the tree connect head nodes with their dependants and are labelled with grammatical functions. Hence at this level, the complex element is a syntactic tree.

For example, for the English sentence *I lived here*, the analytical tree (with simplified labels) looks as follows:

(2.4)



Analytical trees are totally ordered and they can be non-projective. The order of the nodes reflects the order of the corresponding words in the underlying sentence.

Besides dependency, edges in a tree can also represent other relations, such as coordination, apposition or coreferences.

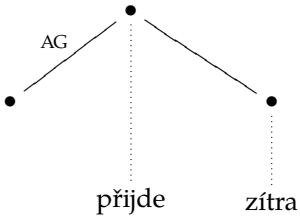
2.2.2 Tectogrammatical Level (deep syntax)

The goal of the tectogrammatical level is to abstract from language specific phenomena. Only autosemantic words correspond to nodes in a tectogrammatical tree, the synsemantic words are encoded in node or edge labels. On the other hand, tectogrammatical structures can contain nodes that are not lexicalized in the linear representation of a sentence (and hence they do not occur in its analytical tree either). For example, the unexpressed subject in the following sentence has its own node in the corresponding tectogrammatical structure:

(2.5) *Přijde* *zítra.*
come-3SG,FUT tomorrow

"He/she will come tomorrow." (Cze)

(2.6)



Elements that depend on a verb can be either actants or free modifiers. Actants differ from free modifiers in that there can be, at most, one actant of a particular type for a verb (coordination phrases are considered to be one unit). There are the following actants:

agent the role of the active participant on a process, usually realized by subject in active sentences

patient the role of the passive participant on a process, usually realized by direct object in active sentences and subject in passive sentences

addressee the beneficiary of a process, often realized by indirect object or an equivalent prepositional phrase.

source the origin of a process, either local or conceptual

effect the result of a process

Whether an actant can (or must) occur as a dependant of a verb is determined in the valence frame of the verb. Actants can be obligatory or facultative; free modifiers can be obligatory within a valency frame.

It is noteworthy that tectogrammatical trees are always, by definition, projective. The order of the nodes in a tectogrammatical tree reflects the topic-focus articulation.

2.3 Equivalence of Linguistic Expressions

Equivalence is a relation which is reflexive, symmetrical and transitive. Two linguistic expressions are equivalent if they have the same meaning, and they are strictly equivalent if they have the same meaning in any context they may occur in (Panevová, 1980). Hence the following sentences are equivalent, the first one being active and the second one passive:

(2.7) *Anię* *namalował* *Tomasz.*
Anna-ACC draw-LPART,MASC,SG Thomas-NOM

"Thomas painted Anna." (Pol)

(2.8) *Ania* *została* *namalowana* *przez* *Tomasza.*
Anna-NOM become-LPART,FEM,SG draw-PART,PASS,FEM,SG,NOM by Thomas-GEN

"Anna was painted by Thomas." (Pol)

Nevertheless, these expressions are not strictly equivalent because if we add the free modifier *z radością* "with pleasure" to them they gain a different meaning (the modifier depends on the subject which is different in both sentences).

We say that two syntactic structures are structurally equivalent if they are represented by isomorphic trees (regardless of the order of nodes).

2.4 Topic-Focus Articulation

An essential component of the linguistic description, namely of the tectogrammatical level, is the topic-focus articulation (Sgall et al. (1980) give a detailed description of the problematics). It expresses the grade of context-boundness and it may influence the meaning of a proposition; two structurally equivalent propositions may have different meanings if they differ in the topic-focus articulation, i.e., in the order of nodes in the tectogrammatical tree, as in the following example (if the intonation is unmarked):

(2.9) *Na* *Moravě* *se* *mluví* *česky.*
in Moravia-LOC,SG REFL speak-3SG,PRES Czech

"In Moravia, Czech is spoken." (Cze)

(2.10) *Česky* *se* *mluví* *na* *Moravě.*
Czech REFL speak-3SG,PRES in Moravia-LOC,SG

"Czech is spoken in Moravia." (Cze)

In an MT system between languages that express the topic-focus articulation mainly by word order, it is widely possible to use a free-ride, i.e., not to consider the word order at the verbal level in the transfer phase. Of course, local word order (such as that of elements of a noun phrase) may require rearrangement.

2.5 Markedness and Underspecification

2.5.1 Markedness

The concept of markedness was developed within the Prague linguistic school, initially for the phonological level. Later, it was generalized for other linguistic levels as well. For syntax and morphosyntax, a detailed formalization in context of the naturalness theory offer Mayerthaler et al. (1998). The markedness of a linguistic sign is complementary to its naturalness. Marked elements or constructions usually are more complex then unmarked ones, they occur less often in propositions and they can be observed in more languages around the world.

As our main focus lies on the syntactic and morphosyntactic level, we constrain ourselves to syntactic and morphosyntactic markedness. According to Mayerthaler et al. (1998), the markedness of a construction grows with its complexity, i.e., the number of nodes it consists of. Furthermore, a construction is more marked than another one if it contains more empty (null) elements.

In the area of machine translation, it is a significant problem if a construction in the target language has a marked counterpart which is its combinatoric variant and there is no marked element in the source language for it.

As an example, let us consider the following Czech sentence which is ambiguous (the tense is underspecified) because the verb is in conditional mood:

(2.11) *přišel* *bych*
come-LPART,MASC,SG would-1SG

"I would come/I would have come." (Cze)

If translating into a language where the two meanings are realized by combinatoric variants, we have to know the formally underspecified tense in order to translate the sentence correctly. In Lithuanian, for example, the following translations are possible and exclude each other depending on the context:

(2.12) *ateičiau*
would-come-1SG,PRES

"I would come." (Lit)

(2.13) *būčiau* *atėjęs*
would-be-1SG,PRES come-PART,ACT,PAST,MASC,SG

"I would have come." (Lit)

This problem also affects MT systems that aim to deeply parse whole sentences since the information that is necessary to decide which combinatoric variant to choose, may only be obtained from the intersententional context.

2.5.2 Underspecification

The concept of underspecification concerns linguistic features that are associated with word forms and phrases. A feature bundle is underspecified if it does not include all relevant features or if a feature's value is underspecified in itself (if an underspecified feature's value is recursively embedded, we call this situation inherited underspecification). For example, the Czech form *ženě* "woman-DAT/LOC" is underspecified since it is morphologically ambiguous with respect to case. On the other hand, the sentence *Přijde* is syntactically underspecified with respect to gender since the subject is not realized, e.g., by a personal pronoun *on/ona/ono* "he/she/it". While the morphological underspecification is inherent for many word forms and gets resolved (at least partially) during the parsing, the syntactic underspecification may be caused by the lack of context (if no intersententional dependencies are considered) and, in the case of partial parsing, the missing dependencies can be seen as (fully) underspecified (Federici et al., 1996). Underspecification can be strict, potential or obligatory. If a form or construction is strictly underspecified, it has to be resolved by the surrounding context; otherwise, the sentence would be ill-formed. In the case of potential underspecification, there is a default value which applies for the underspecified feature if the context does not resolve the uncertainty; otherwise, the default value gets overwritten by the context. An obligatory underspecification must not be resolved.

Let us show a couple of examples. The Polish impersonal participles with *-no/-to* are obligatorily underspecified with respect to subject (while fixing the tense), e.g., *Nie chciano wrócić* "One did not want to return." In Lithuanian, for example, there are

sentences with a partitive actant in genitive (e.g., *Įbėgdavo čia jaunų merginų* "Young girls used to come here"). These can be analyzed, according to Ambrazas et al. (1999), in the way that the genitival noun phrase depends on a null element. Nouns are potentially underspecified with respect to person. If they depend on finite verbs they usually are in the third person. Nevertheless, the feature of person can be different if a noun phrase with a noun as its head is specified by a pronoun, e.g., *My studenci nie mamy pieniędzy* "We students have no money"; in Slovenian, no pronoun is necessary in such a construction, e.g., *Slovenci volimo…* "we Slovenes vote for…" Generally, the underspecification gets resolved through syntactic relations with other elements of the sentence, often through agreement. For example, in the Czech sentence *Přijdu za tebou* "I will come to you", the underspecified gender and number of the general subject can be (fully or partially) resolved by adding a transgressive. Adding a masculine transgressive resolves the aforementioned features completely (*Dokonče práci přijdu* "After having finished the work, I will come"), whereas the transgressival form *dokončíc* only reduces the underspecification of the gender in that it excludes the masculine value while fixing the number to singular.

Interesting examples can be found in dialects. The Russian transgressive, for example, can be used dialectically to express the perfect tense (Trubinskij, 1984) and in this function, it is potentially underspecified with respect to tense. The unmarked use would be, e.g., *Лена приехавши* "Lena has come", hence the default value of the tense is present. Through an auxiliary, a different tense can be expressed: *Лена была/будет приехавши* "Lenna had/will have come." In the analogous Lithuanian construction, the infinite verb form is strictly underspecified, as it always requires an auxiliary to specify the person: *Lena buvo/yra/bus atvažiavusi*.

In Lower Sorbian, there is practically only one past tense nowadays which is built with the *l*-participle. This verb form is strictly underspecified with respect to person since it always has to be resolved by an auxiliary. In other words, *Pśišeł* "came" is not a well-formed sentence, the verb always has to be accompanied by an auxiliary: *(ja) som pśišeł, (ty) sy pśišeł* or *(wón) jo pśišeł* "I came, you came, he come."

It is noteworthy that from the diachronic point of view, strictly underspecified constructions tend to become potentially underspecified. The tendency is related to the principle of markedness reduction and to the fact that syntactically complex constructions are mostly more marked.

2.6 Notations of Data Structures and Rules

This section explains essential notations regarding linguistic data structures and grammar rules.

2.6.1 Feature Structures

In our framework, the basic data structure for representation of linguistic data are feature structures. A feature structure is an attribute-value-matrix (AVM) whereby the values of the attributes are atoms, strings or complex values (lists, sets, embedded feature structures). Feature structures are usually typed, i.e., there is a global type hierarchy and each feature structure is assigned a type. Here is an example of a simple feature structure:

(2.14)
$$\begin{bmatrix} \textit{adv} & \\ \text{LEMMA} & \text{'quickly'} \\ \text{POS} & \text{adv} \end{bmatrix}$$

Each linguistically significant entity has a set of relevant features. The value of a feature may be underspecified, i.e., its value might not be fully specified. Ambiguous feature values may be resolvable from the context.

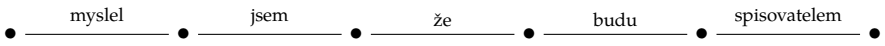
The most typical operation on feature structures is unification which is a combination of mutually compatible attribute values. What is often used in rules is partial unification, i.e., only specified attributes are unified (for example: case, gender, number etc.).

2.6.2 Charts

As an auxiliary data structure, a chart, is used for parsing in our framework. Formally, it is a multigraph that represents all parsing hypotheses that are valid up to a certain point in the parsing process. At the end of the process, the remaining valid hypotheses build up the result of the parser. One possible implementation of a chart parser describes Colmerauer (1969).

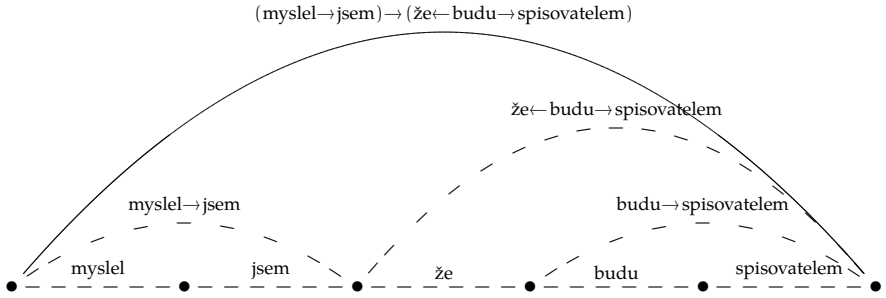
Here is an example of a chart at the beginning of the parsing process:

(2.15)



And here is the the chart for the same sentence after the parsing process:

(2.16)



2.6.3 Grammar Rules

Rule-based systems contain grammars for syntactic analysis and/or generation. These grammars consist of declarative rules that prescribe how to combine words and constituents into complex structures. The most common type used in linguistic formalisms are context-free grammars which operate on adjacent constituents. However, some kinds of non-projective dependencies can be recognized within context-free grammars by means of rule templates—the so called functional uncertainty.

A rule consists of a left-hand side which is matched against a part of the parsed input, and a right-hand side which is the result of the rule's application. In most formalisms, rules can be associated with conditions to restrict their application.

We use the following schematic rule notation

(2.17) $[A] + [B] == [C]$

where A, B and C are feature structures. A rule can be applied if its left-hand side (A and B) unifies with a path of the chain graph. In such a case, a new edge is added to the chain graph which spans the edges that are covered by the left-hand side of the rule and is labelled with its right-hand side. The feature structure, which the new edge is labelled with, is defined on the right-hand side of the rule.

The mechanism of rule interpretation which we are using is described in detail in Chapter 7. See also Appendix A for the list of rules which are used in our grammar for Czech.

3

Basic Facts about Baltic and Slavic Languages

In this chapter, we briefly describe the family of Baltic and Slavic languages and point out some linguistic (mostly syntactic) facts that are of certain importance for the MT.

Some basic information about the discussed languages (such as the number of speakers) is taken from (Bußmann, 2002).

3.1 Baltic languages

Baltic languages are a small group in Eastern Europe at the Baltic Sea. They have rich declension and conjugation and have preserved many features of the older stage of Indo-European. From the two living Baltic languages, Latvian and Lithuanian, we have used Lithuanian as target language in our system.

3.1.1 Extinct Baltic languages

The most known extinct Baltic languages are Selian, Curonian, Sudovian and Old Prussian. We will only give some information on Old Prussian here.

Old Prussian

Only for Old Prussian, there are longer documents left to us. Except of two glossaries, we have three Lutherian catechisms that have been translated from German. Old Prussian, a West Baltic language, was spoken in the region between Vistula and Neman until the 17th or 18th century when its speakers assimilated to Lithuanian, German and Polish. The examinated stage of the languages (based on the preserved texts) had probably 4–5 cases and a number of analytical tenses. The use of determinative pronouns and the numeral *one* as articles may have been an influence of German which was present in the Prussian territory from the 11th century on (when the crusaders exterminated most of the Prussian population). In recent years, a grammar and several dictionaries of Old Prussian have been published, some of them propagating a revived form of the language.

3.1.2 Living Baltic languages

Latvian

Latvian is spoken in Latvia by approx. 1.5 million people. It is more innovative that Lithuanian, its grammar is comparatively simpler, it has an initial accent (as an effect of the Finno-ugric substrate) but its vocabulary is Baltic. It has phonological length.

There are analytical tenses similar to the Lithuanian ones as well as *modus relativus* (e.g., *es smejoties* "I reportedly laugh"). It also has a specific verbal category called *debitive* which is used to express *to have* (e.g., *viņam jābūt mājās* "he has to be at home"). See (Forssman, 2001) for a detailed description of the Latvian grammar.

Lithuanian

Lithuanian is spoken by approx. 4 million people in Lithuania and by national minorities in Poland and Belarus. It has rich declension and conjugation and preserves many features of the old Indo-European. Due to this fact it is very important for the examination of Indo-European in general. It has developed a complicated system of moods and tenses in which participles have an important role. Some of these constructions are more or less similar to sentence patterns in Slavic languages.

Let us have a look at some examples. The following two sentences show how passive participles can be used to express evidentiality:

(3.1) *Darbininkų* *nešama* *plytos.*
workers-MASC,PL,GEN carry-PART,PASS,PRES,NEUT,NOM plate-FEM,PL,NOM

"The workers are evidently carrying plates."

(3.2) *Čia* *kuršių* *gyventa.*
here Curonian-MASC,PL,GEN lived-PART,PASS,PAST,NEUT,NOM

"Curonians evidently lived here."

This construction expresses the narrative, a grammatical category absent from Slavic languages except for Bulgarian and Macedonian.

In the following sentence, the use of a participle of necessity is shown:

(3.3) *Dar* *minėtina,* *kad…*
still mention-PARTNEC,NEUT that

"Moreover it should be mentioned that…"

Lithuanian also has special periphrastic progressive tenses:

(3.4) *Aš* *buvau* *bevalgąs,* *kai* *Gintarė* *atėjo.*
I-NOM was eating-PARTBE,MASC,SG,NOM when Gintarė-NOM came-3SG,PAST

"I was eating when Gintarė came."

More examples of these and similar sentence patterns in Baltic and Slavic languages are given in Section 6.2.2.

3.2 Slavic languages

Slavic languages are a large languages family in Central and Eastern Europe and in the Balcans as well as in part of Asia. The largest Slavic language is Russian, followed by Ukrainian. The following Slavic languages are official languages of the European Union:

- Bulgarian
- Czech
- Polish
- Slovak
- Slovenian

Regional Slavic languages on the EU's territory are Lower and Upper Sorbian (in the German provinces Brandenburg and Saxonia, respectively), Russian in Estonia, Latvia and Lithuania, Macedonian in Greece and Slovenian in Austria and Italy.

Slavic languages have rich conjugation and most of them (except for Bulgarian and Macedonian) rich nominal declension.

3.2.1 Extinct Slavic languages

Old Church Slavic

Old Church Slavic is the language in which the oldest Slavic texts have been written. It is based on the medieval dialect of the Macedonian metropolis Solun (today's Thessaloniki) and was the religious language of Great Moravia. It is well documented, there are dictionaries and grammars of this language.

Polabian

Polabian is an extinct West Slavic language which was spoken in today's North-East Germany and on the Baltic island Rügen. It became extinct in the 18th century, last speakers were the inhabitants of the Lüneburger Wendland in Lower Saxonia. Polabian was closely related to Kashubian and to the Sorbian languages.

3.2.2 Living Slavic languages

Belarussian

Belarussian is spoken in Belarus by approx. 7 million people. It as an East Slavic language closely related to Russian and Ukrainian.

Bosnian, Croatian, Montenegrin, and Serbian (BCS)

These languages, spoken on the territory of former Yugoslavia, are part of the South Slavic dialect continuum. In the past, a collective term Serbo-Croatian has been used.

They are closely related to Slovenian in the North-West and Bulgarian and Macedonian in the South-East. All these languages have together approx. 16 million speakers.

Bulgarian

Bulgarian is spoken by approx. 7.5 million people in Bulgaria. It is a bit specific among Slavic languages since it has lost the declension of substantives. Furthermore, it has developed a postponed definite article and a specific mood—the narrative, which roughly corresponds to the East Baltic modus relativus.

Czech

This West Slavic language is spoken by approx. 10.5 million people in the Czech Republic. Due to historical circumstances, it has two variants—a literary one which is quite distinct from the colloquial variant, thus there is a diglossy. Besides Russian, it is the Slavic language with most computational linguistic resources and tools.

Kashubian

Kashubian (also *Cassubian, Pomeranian*) is spoken in Northern Poland by approx. 50.000 people. All Kashubians are bilingual (Polish). This language is—besides South-West Macedonian and the North-West Russian dialects—the only Slavic language which has a fully grammaticalized possessive perfect[1], for example:

(3.5) *mam* *bëté*
have-1SG,PRES been-PART,PAST,NEUT

"I have been"

The closely related Slovincian language (also in Northern Poland) became extinct at the beginning of the 20th century.

Lower Sorbian

Lower Sorbian is spoken by approx. 15,000 people in the German region of Lower Lusatia. As a peripheral dialect, it has preserved many old linguistic features, such as the dual, supine and concise past tenses (aorist and imperfect). On the other hand, it has been strongly influenced by the surrounding German language so that there are are many German calques, for example (cf. Ger "es gibt hier viele Flüsse"):

(3.6) *How* *dajo* *wjele* *rěkow.*
here gives-3SG,PRES many rivers-FEM,PL,GEN

[1] The term *possessive perfect* is used for example in (Trubinskij, 1984) to denote a perfective sentence pattern built by the expression *to have* and a formally passive participle, e.g., Rus (dial.) *у меня корова подоено* "I have milked a/the cow", Mac *имам дојдено* "I have come" etc. A typical feature of this analytical 'tense' is the non-agreeing past passive participle in the neuter form.

"There are many rivers here."

As in some other West Slavic idioms, past passive participles can have an active meaning, for example:

(3.7) *Som stanjony.*
am stand-up-PART,PAST,MASC,SG,NOM

"I did stand up."

There is also an (although not fully grammaticalized) possessive perfect:

(3.8) *Mam dom natwarjony.*
have-1SG,PRES house-SG,SG,ACC built-PART,PAST,MASC,SG,ACC

"I have built a house."

Macedonian

Macedonian is a South Slavic language spoken by approx. 1.5 million people in Macedonia and by national minorities in Albania, Bulgaria and Aegean Macedonia (today's Greece).

Similarly to Bulgarian, it has lost substantival declension and developed postponed definite articles (with three deictic degrees, e.g., *куќава* "this house", *куќата* "the house", *куќана* "that house"). Furthermore, there is an object doubling which compensates in a certain sense the loss of cases, e.g., *Јас ја гледам Марија* "I see Mary".

The dialects in the South-West of the Macedonian language territory have developed a possessive perfect:

(3.9) *Јас ја имам вчера видено Марија.*
I-NOM her-DAT have-1SG,PRES yesterday seen-PART,PAST,NEUT,SG Mary

"I have seen Mary."

The past passive participle can also have an active meaning and build a sort of past tense with the auxiliary *to be* in constructions like the following ones:

(3.10) *Сум дојден.*
am come-PART,PAST,MASC,SG

"I have come."

(3.11) *Сум јаден.*
am eaten-PART,PAST,MASC,SG

"I have eaten."

Polish

Polish is a West Slavic language spoken in Poland (approx. 38 million people) and by national minorities in Belarus, Czech Republic, Lithuania and Ukraine. There is

also a large Polish speaking community in the United States. From the viewpoint of the comparative Slavic linguistics, the predicative use of past passive participles is an interesting feature, for example:

(3.12) *Rozmawiano* *pijąc* *herbatę.*
talk-PART,PAST,NEUT,SG drink-TRG,PRES FEM,SG,ACC

"It has been talked while drinking tea."

(3.13) *Już* *się* *nie* *śmiano.*
already REFL NEG lough-PART,PAST,NEUT,SG

"One did not laugh any more."

Another specific feature are the auxiliary agglutinants, e.g., *miał**am*** "I had".

Russian

Russian is an East Slavic language spoken by approx. 150 million people in Russia and the former Soviet republics. It is the Slavic language with the most speakers.

One of the interesting properties of Russian is its lack of the auxiliary verb in the past tense, for example:

(3.14) *Я* *пришёл.*
I-NOM come-LPART,MASC,SG

"I have come'

Furthermore, the use of the verb *быть* "to be" is very rare, for example:

(3.15) *Он* *хороший* *человек.*
he-NOM good-MASC,SG,NOM man-MASC,SG,NOM

"He is a good man.'

Slovak

Slovak is a West Slavic language with approx. 4.5 million speakers which is part of the Czech-Slovak dialect continuum (Townsend and Janda, 2003). It is closely related to Czech, the differences are mainly of phonetical nature.

Slovenian

Slovenian is spoken in Slovenia and by national minorities in Carinthia (Austria) and Friaul (Italy) by approx. 1.8 million speakers. It has preserved some old features such as the dual.

Ukrainian

Ukrainian is an East Slavic language spoken by approx. 37 million people in the Ukraine. Similarly to Polish, there is a predicative use of past passive participles.

There is also a specific future tense, built with suffices, for example:

(3.16) *читатиму*
read-1SG,FUT

"I will read'

Upper Sorbian

Upper Sorbian is spoken by approx. 35,000 people in the German region of Upper Lusatia. As a peripheral dialect, it has preserved many old linguistic features, such as the dual and concise past tenses (aorist and imperfect). On the other hand, it has been strongly influenced by the surrounding German language so that there are are many German calques.

4

An Overview of MT Systems between Related Languages

MT between closely related languages has a long tradition and it has experienced a rebirth in the last decade. The first experiments were done for Slavic and Scandinavian languages. The shallow-transfer approach has been shown to give viable results for related languages with very rich inflection as well as for analytical and agglutinative languages. We give a brief overview of several systems in the following sections.

4.1 Slavic Languages

4.1.1 RUSLAN

The first MT system for closely related Slavic languages was RUSLAN (Hajič, 1987; Bémová et al., 1988), translating from Czech into Russian. The system used a deep syntactic analysis and a full-fledged transfer. Its core modules were implemented in Q-systems (Colmerauer, 1969).

4.1.2 Česílko

An MT system from Czech into Slovak was implemented by Hajič et al. (2000). As there are almost no syntactic or semantic differences between the two languages, the system uses a direct lemma-to-lemma lexical transfer with a one-to-one dictionary.

Later, the system was adapted to the language pair Czech-Polish (Dębowski et al., 2002) and finally, the shallow-transfer approach has been suggested and implemented by Hajič et al. (2003) after experiments with translation from Czech into Lithuanian.

The MT system *Česílko* originally was an experimental system for automatic translation as a supporting module for pre-filled translation memories. Since the source and target language of the system are closely related, the system did not perform any syntactic analysis but it translated the input text on a lemma-to-lemma and tag-to-tag basis. The system consisted solely of the following modules (we have reused some of them in our experiments):

1. morphological tagger for Czech
2. bilingual glossaries
3. morphological synthesis for Slovak/Polish.

Czech is a language with rich inflection, i.e., a word usually has many different endings that express various morphological categories. The morphological analyzer

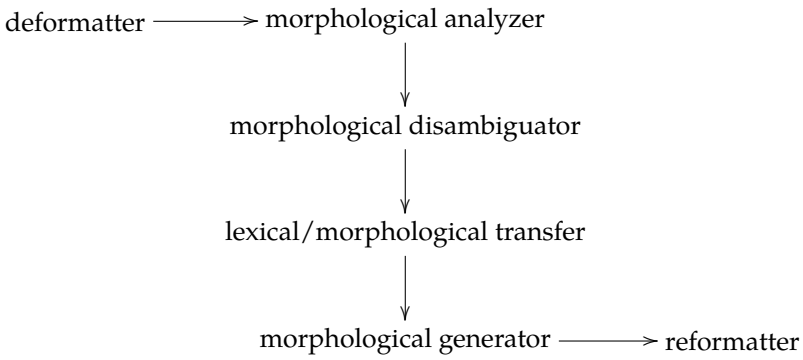


Figure 4.1: Architecture of the first version of the system Česílko

assigns a set of lemmas and tags to each word. As it was necessary to have only one tag for each word determined by the context of the sentence, a statistical tagger was used with an accuracy of approx. 94% (Hajič and Kuboň, 2003). The use of the tagger was necessary since the input of the lexical transfer (which was the immediately following module) was expected to be disambiguated.

The bilingual glossaries contained lemmas of the source language and their counterparts in the target language. It is an inherent problem of dictionaries that a source lemma often corresponds to several lemmas in the target language and the correct translation depends on the semantic context, the style of the text etc. Even for very closely related languages such as Czech and Slovak, there may occur discrepancies relevant for the meaning. This problem has been partially solved by the division of the glossary into a domain-specific part and a general part. During the lexical transfer, the domain-specific glossary is used first and the general glossary is used only if no translation has been found.

It may happen that no translation is found even in the general glossary since no dictionary can contain all the words of a language. In such a case, the original lemma is left untranslated in the text which may help a human post-editor to correct the translation.

The final phase generates word forms in the target language which is comparatively simple. It may happen that a lemma is unknown in the morphological module of the target language because it has not been translated at all or simply because the module does not contain it. In such a case, the lemma is left unchanged in the target sentence.

The system was evaluated using the Trados Translator's Workbench (TTW). The result of the automatic translation was post-edited manually to be grammatically and semantically correct. Afterwards, the TTW calculated the similarity of each automatically translated sentence with its manually corrected version. The accuracy for a set of sentences has been expressed as a weighted mean of sentence accuracies weighted by length (number of words). The accuracy for the language pair Czech-Slovak was around 90% while for Polish as target language it reached, according to Dębowski et al. (2002), 71.4%. The Trados metric was believed to reflect the effort a post-editor would have to put into making the translation grammatically and semantically correct. Unfortunately, the algorithm used by Trados is not public so it is not exactly known how the evaluation proceeds. However, the numbers can be used to compare different methods (given a language pair and a text for evaluation) or two language pairs (if the same method is used).

4.1.3 GUAT

An MT system from Slovenian into Serbian, based on Apertium, has been experimentally implemented by Vičič (2008) (the architecture of the framework is described in Section 4.5). The system utilizes the available Slovenian morphological analyzer. The other linguistic resources were built automatically by exploiting available corpora for both languages. Even transfer rules are intended to be induced automatically in the future versions of the system. Currently, there are only a few hand-written rules.

In the last version of GUAT, our ranker has been used for the language pair Slovenian-Serbian with a significant improvement in translation quality (Jernej Vičič, personal communication).

4.2 Scandinavian Languages

4.2.1 PONS

There has been extensive research in MT between various Scandinavian languages. The first extensive experiment was the PONS (Partiell Oversettelse mellom Nærstående Språk = Partial translation between closely related languages) system (Dyvik, 1995) that translated from Norwegian into Swedish. The authors argue that if two languages are close enough, it is mostly not necessary to "waste time finding a lot of redundant grammatical and semantic information about the expressions". They suggest that for closely related languages, one should choose a different strategy than for distant languages. Concretely for Scandinavian languages, "formal equivalence will often imply denotational and stylistic equivalence". The general principle is to use as much of the structure of the source sentence as possible "within the limits imposed by idiomacity". In particular, semantic and stylistic properties of translated sentences are not taken into account, relying on the closeness of both languages at the corresponding

levels, since "in closely related languages, similar effect can be achieved with similar means". The source sentence serves as a template for the encoding of the target sentence.

The core of the system is based on the D-PATR unification-based formalism (Kartunnen, 1986). An interesting property of this system is that no morphological analyzer was used, all word forms were stored in the lexicon. Each entry is a set of equations which define a feature structure. As a convenient method of adding handwritten entries, there are templates for defining recurring sets of equations.

Before parsing, the source text is divided into substrings at certain punctuation marks. The substrings are then parsed by a bottom-up unification-based chart parser. The grammar is not designed to fully cover the source sentence—the result of the parser is typically a set of partial analyses. At the end of the parsing process, the parser chooses the edge sequence(s) with the lowest number of edges which correspond(s) to the maximal analyses of the substring. Subsequently, each edge is translated separately and the results are concatenated. The system is robust in the sense that "as long as the words are known, some output is guaranteed".

The transfer uses three operating modes. Modes 1 and 2 are "shortcut modes", i.e., the structural similarity between source and target language is exploited. The third mode generates the structure of the target substring from scratch. The 'shortcut'-modes perform a kind of word-to-word translation by substituting target words for source words at the terminal nodes of the parse tree. The transfer is generally nondeterministic. For example, when translating from a language without tense (such as Chinese) into English, a set of English strings is generated with all possible tense values (in other words, underspecification expands in ambiguous output).

Besides Norwegian-to-Swedish, the system has also been tested for English and Norwegian.

4.2.2 Norwegian-Danish

A similar approach was used in the MT system from Norwegian (bokmål) into English that used Danish as an interlingua (Bick and Nygaard, 2007). As there are almost no syntactic differences between these two Scandinavian languages, and there is a widely corresponding polysemy, they generate the Danish translation from the output of a Norwegian tagger by substituting lemmas using a one-to-one dictionary. The output of a newly constructed Norwegian-to-Danish MT system is piped into an existing Danish parser and further processed. This approach exploits the fact that "the polysemy spectrum of many Bokmål words closely matches the semantics of the corresponding Danish word, so different English translation equivalents can be chosen using Danish context-based discriminators".

The first step in the system is disambiguation of lemmas and PoS tagging. The subsequently used Norwegian-Danish one-to-one lexicon was built, mostly automatically, by creating a monolingual automatically lemmatized Norwegian corpus and

regarding Norwegian as 'mis-spelled Danish', using a Danish spell checker on the lemma candidates. Furthermore, phonetic transmutations for Norwegian and Danish were produced to generate hypothetical Danish words from Norwegian words. The presented approach resulted in a list of 226,000 lemmas with Danish translation candidates.

After the tagger, Norwegian lemmas are substituted by Danish ones. Additionally, there is a special handling of compound nouns based on partial translation of words. The morphology of the two languages is not completely isomorphic and there are also some structural differences that are handled by a CG grammar (for example, double definiteness in Norwegian which is solved by substitution rules).

4.2.3 T4F

An English-to-Swedish MT system is presented by Ahrenberg and Holmqvist (2005). The authors claim that even English and Swedish are close enough for what they call a 'direct' model.

The system has been designed to support quick development of domain-restricted machine translation. It is named T4F which is an abbreviation of "Tokenization, Tagging, Transfer, Transposition and Filtering". The system uses a dictionary with a greedy algorithm, i.e., the longest match is used. Word order is handled by transposition rules with the source word order being the "point of departure". Again, the authors claim for a structurally similar language, "the case for abstract syntactic analysis seems less convincing". In MT systems, they distinguish *concrete objects* (sentences) and *abstract objects* (structural representation of sentences) and argue that it is an "unnecessary roundabout" to introduce an abstract representation for the purpose of creating another concrete objects which is more or less isomorphic to the first one. To sum up, translation units in English and Swedish correspond and the rare structural differences are tied to lexical entries. Furthermore, grammatical morphemes correspond "fairly well" in numbers and use (a morphological variant in English corresponds only to a small number of morphological variants for any Swedish translation).

There are three phases: analysis, transfer and selection. The analysis consists of tokenization and tagging. Besides inherent features, contextual information is assigned to the tokens too, such as the definiteness of English nouns. For analysis, the FDG parser of Connextor is used.

In the transfer phase, the English tokens are considered one by one. For an English token e, all Swedish tokens are retrieved that are defined as possible translation of e and that match the inherent and contextual information of e. As usual, the English token is used if no Swedish translation can be found. To reduce the size of the set of possible translations, a filtering module is applied. After filtering, target sentences are derived by combining all remaining tokens and the alternative translation is ranked according to a bigram model.

BLEU has been used for evaluation. As the authors claim, if system modules (lexical entries, rules) are obtained automatically and not revised carefully, the filtering and reordering rules are less applicable and as a result, "the burden of selection of a translation falls on the probabilistic ranking procedure".

Let us give an example of the difference between inherent and contextual feature, consider the following English phrase:

(4.1) *the Employees table*

The noun *employees* is contextually definite which is given through the article *the* in front of it. In the Swedish translation, the definiteness is an inherent feature of the noun which is expressed by an appended morpheme:

(4.2) *tabellen Anställda*

4.3 Turkic Languages

For Turkic languages, an experimental MT system from Turkish into Crimean Tatar has been implemented by Altintas and Cicekli (2002). They claim that for languages with shared historical background and similar culture, there is no need for a semantic analyzer. As most parts of the grammar are common in both languages, the system focuses on differences at the morphemic level, thus translation from Turkish into Crimean Tatar is basically "disambiguated word-for-word translation".

For the implemented language pair, there are several categories of transfer rules:

No change of roots or morphemes; no translation rules are applied.
Root change — only the root is changed (using the bilingual dictionary).
Morpheme change — the root remains the same.
Root and morpheme change is the combination of the previous two categories.
Verbs that effect its object — changing the case of the object.
Structures effecting previous and following words — for example, if a morpheme is added to a verb in Turkish instead to its dependent noun in Crimean Tatar.
More than one word map to one word — a typical case of multiword expressions.
One word maps to more than one word — a typical case of multiword expressions.

The rules can generally be applied in any order, except for the rules that change the root. The system is implemented using finite-state tools with an interface written in Java. The system outputs all possible results of rule application and lexical ambiguities.

4.4 Celtic Languages

A machine translation system between Irish and Scottish Gaelic (both Insular Celtic/-Goidelic languages) is presented by Scannell (2006). Both languages are not mutually intelligible, at least in their spoken variant, but their grammars are very close since they have a common ancestor—Middle Irish, and a shared literary tradition written

in the so-called Classical Gaelic (Gaeilge Chlasaiceach) up through the 18th century. Historically, there was a geographic continuum of dialects from the far southwest of Ireland to the northernmost parts of Scotland. The aim of the system is information retrieval for all Goidelic languages.

There are the following modules in the system:

1. Irish standardization,
2. POS tagging, stemming, and chunking,
3. Word sense disambiguation,
4. Syntactic transfer,
5. Lexical transfer,
6. Scottish post-processing.

It is noteworthy that the input is normalized before being translated since the orthography of processed texts may differ. It is obvious that one cannot use statistical MT methods for these languages since there are no suitable corpora available. However, the differences between the two languages are comparatively small, thus chunking is believed to be sufficient in most cases. Formally, the result of the chunker may be seen as a parse tree of depth one. Due to the syntactic closeness of both languages, the biggest translation problem occurs at the semantic level; therefore, a word sense disambiguation is an integral part of the system.

A specific feature of the Insular Celtic languages is the initial mutation of consonants which mostly has grammatical meaning. For example, the Irish word *céad* can mean "first" or "one hundred" and precedes the noun it modifies in each case. However, when it means "first" then it causes lenition of the modified noun. This kind of grammatical change is very important for the disambiguation.

Syntactic transfer is a necessary part of the system due to periphrastic constructions which are present only in one language. For example, there is no structurally equal analogue of the present Irish verb in Scottish. So the phrase *(bh)feiceann tú* "you see" is translated as *tha thu a'faicinn* "you are a seeing" in Scottish. In a case like this, the chunker has to identify the subject noun phrase.

The rules are transformed into a finite state recognizer which can be compiled for fast matching against the tagged and chunked input stream. In the current version, there are less than 100 transfer rules. Their number is expected to grow rapidly as new rules for handling additional multiword expressions will be added.

The prevalent part (90%) of the lexicon has been extracted automatically from two electronic dictionaries—Irish-English and Scottish-English.

Finally, there is a post-processing phase performing local corrections (such as incorrect initial mutation) which is based on the Gramadóir grammar checker.

4.5 Romance Languages

For the Romance languages of Spain, the *Apertium* system has been implemented (Corbi-Bellot et al., 2005). The system is largely based on the older MT systems in-

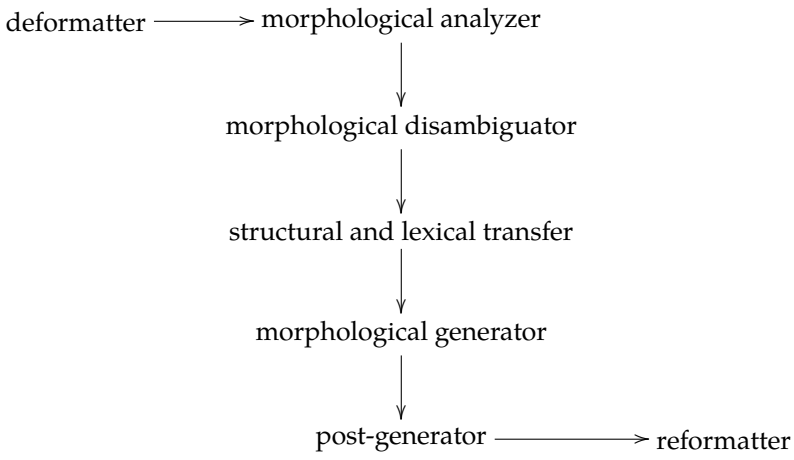


Figure 4.2: Architecture of the shallow-transfer MT system Apertium

terNOSTRUM (Forcada et al., 2001) and Tradutor Universia[1]. The authors claim that a word-to-word translation may give an adequate translation of 75% of the text. The system uses the shallow-transfer approach. Open source data are available for a number of language pairs.

The system consists of the following modules:

1. The de-formatter converts the source text from a format such as HTML or RTF to an internal format with tags.
2. The morphological analyzer delivers lemmas and morphological tags for source word forms.
3. The output of the morphological analyzer is disambiguated by the subsequent tagger (reportedly, about 30% of word forms are morphologically ambiguous in Romance languages).
4. The lexical transfer module is used from within the structural transfer module. The dictionary contains one translation for each entry which is a source lemma or a multiword expression.
5. The structural transfer module uses finite-state pattern matching to detect fixed-length patterns of lemmas to handle grammatical divergences between both languages (the matching strategy is left-to-right, longest match).

[1] http://tradutor.universia.net

6. The morphological generator produces inflected forms for target lemmas and tags.
7. The post-generator adapts the surface representation of the translation, e.g., *me* "to me" and *o* "it/him" in Portuguese is contracted to *mo* etc.
8. Finally, the re-formatter restores the original input format (HTML, RTF etc.).

It is also claimed that this architecture be suitable even for pairs of distant languages, such as Spanish-Basque, which is a language pair intended to be implemented within Apertium. For this language pair, a deeper-transfer architecture is being designed.

Because of the morphological ambiguity, a tagger has been prepended before the transfer. The dictionaries contain single equivalents as well as multiword expressions. Transfer rules, which handle, for example, the rearrangement of clitic pronouns, have the form pattern-action, and there are approx. 90 of them. The system is able to process about 5,000 words per second.

Machine translation from Portuguese into Spanish within Apertium was implemented by Armentano-Oller et al. (2006). The system is able to recognize 9,700 Protuguese lemmas and to generate the same amount of Spanish lemmas. The bilingual dictionary contains 9,100 lemma-to-lemma pairs.

5

Free-rides in Baltic and Slavic Languages

The experience from the field of MT between closely related languages presented in the previous sections shows that it is useful to classify the language similarity in several categories. We distinguish typological, morphological, syntactic, and lexical similarity. In the following, we discuss these categories from the viewpoint of machine translation.

5.1 Typological Similarity

The first type of similarity is probably, for our purposes, the most significant one. If both the source and target language are of different language types, it is more difficult to obtain good translation quality. Features like word order, the existence or non-existence of articles, different temporal system and similar discrepancies have direct consequences for translation quality.

Let us take Czech and Macedonian as an example of a pair of languages which belong to one language family but differ typologically. Both languages have rich verbal inflection and a high degree of word order freedom, thus it is mostly not necessary to change the word order at the verbal level. On the other hand, Macedonian has virtually no nominal declension.

For example, both (5.1) and (5.3) mean approximately "*My brother read a/the book*".

(5.1) *Můj* *bratr* *četl* *knihu*
my-MASC,SG,NOM brother-MASC,SG,NOM read-LPART,MASC,SG book-FEM,SG,ACC

"My brother read a book." (Cze)

(5.2) *Брат* *ми* *читаше* *книга*
brother-MASC,SG me-DAT read-3SG,PAST book-FEM,SG

"My brother read a book." (Mac)

(5.3) *Knihu* *četl* *můj* *bratr*
book-FEM,SG,ACC read-LPART,MASC,SG my-MASC,SG,NOM brother-MASC,SG,NOM

"The book has been read by my brother." (Cze)

(5.4) *Книгата* *ја* *читаше* *брат* *ми*
book-FEM,SG her-ACC read-3SG,PAST brother-MASC,SG me-DAT

"The book has been read by my brother." (Mac)

What these sentences differ in is the information structure. (5.1) should be translated as "*My brother read a book*", whereas (5.3) means in fact "*The book has been read by my brother*". The category of voice differs in both sentences because of the strict word order in English, although in both Czech equivalents, active voice is used. We see that in the Macedonian translation, the word order is exactly the same.

5.2 Syntactic Similarity

Syntactic similarity is also very important, in particular at the verbal level. The differences in verbal valency have negative influence on the quality of translation due to the fact that the transfer requires a large scale valence lexicon for both languages which is extremely costly to produce. The syntactic structure of smaller constituents, such as noun and prepositional phrases, is not that important because it is much easier to analyze those constituents syntactically using a shallow syntactic analysis and thus it is simpler to adapt the syntactic structure of a target sentence locally.

For related languages, the word order of the source sentence is usually preserved, although sometimes it is necessary to change the local word order. For example, in Lithuanian, noun phrases with a genitive attribute:

(5.5) *bratr* *otce*
∗brolis tėvo

"father's brother" (correctly: *tėvo brolis*)

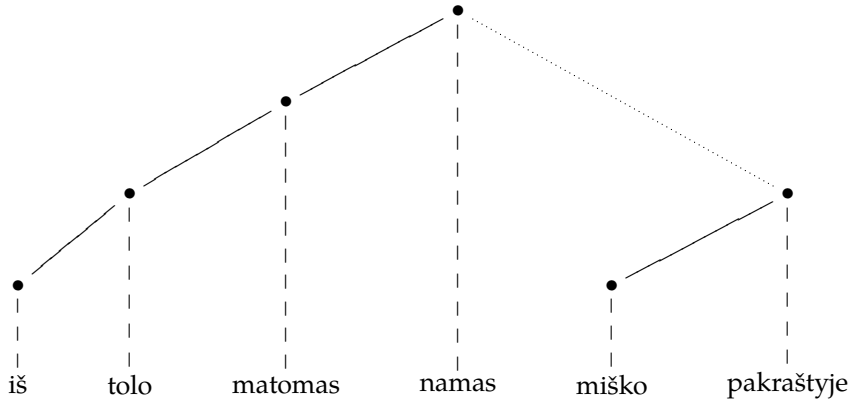
5.2.1 Syntactic Underspecification

In shallow syntactic analysis, only some dependencies in the sentence are analyzed, mostly those in smaller constituents, such as noun and prepositional phrases. Such dependencies should be sufficient in most cases in translation between closely related languages as one can rely on free-rides at the verbal level, although the valence remains a huge problem.

Let us have a look at example (5.6). Dependencies analyzed by the shallow parser are expressed by the solid line, not recognized 'deeper' dependencies by the dotted line.

(5.6) *Iš* *tolo* *matomas* *namas* *miško*
from far visible-MASC,SG,NOM house-MASC,SG,NOM forest-MASC,SG,GEN
pakraštyje.
border-MASC,SG,LOC

"a/the from far visible house at the border of the wood" (Lit)



In the Czech source sentence, the word order of constituents is very similar to (5.6). The only difference is in the translation of *iš tolo* (in Czech *zdaleka*) and the word order in the NP *miško pakraštyje* (genitive attributes follow the governing noun in Czech).

Omitted dependencies (dotted lines in (5.6)) can be considered to be syntactically underspecified. The syntactic structure of a sentence built by a shallow parser is incomplete and could be optionally extended by a subsequent module.

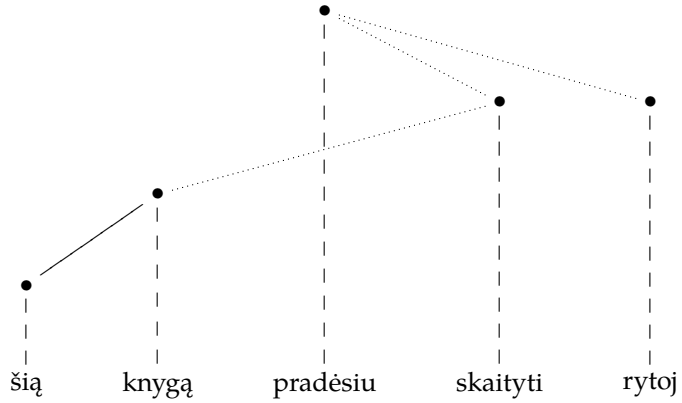
A serious problem for NLP of languages with rich inflection represents the so-called non-projectivity. In these languages, non-projective sentences are still understandable because the word order (at the level of actants) has almost no grammatical meaning. For example, approx. 23% sentences in the Prague Dependency Treebank (Hajič et al., 2001) are non-projective, as reported by Zeman (2004). In the implementation of our system, we do not consider non-projectivity since both languages in our language pairs use the similar types of non-projective dependencies.

For example, the syntactic structure in (5.7), a non-projective Lithuanian sentence, is the same as the structure of its Czech translation.

(5.7)

Šią	*knygą*	*pradėsiu*	*skaityti*	*rytoj.*
this-FEM,SG,ACC	book-FEM,SG,ACC	start-1SG,FUT	read-INF	tomorrow

"I will start to read this book tomorrow." (Lit)



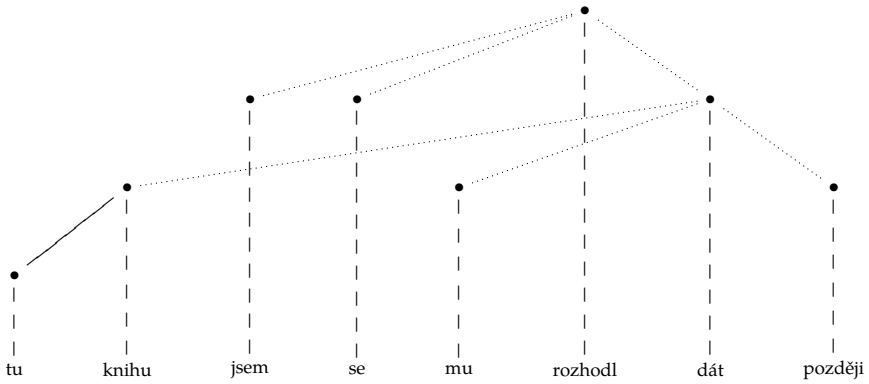
In (5.7), only one gap (discontinuity) occurs. The Czech translation has exactly the same syntactic structure. Nevertheless there are sentences with more gaps and the amount of gaps is theoretically unrestricted (Kuboň, 2001). In such sentences, the high degree of non-projectivity is often caused by two or more verbs (e.g., a finite verb and its infinitival complement) with rich valence frames and contextually affected order of actants. In (5.8) (a slightly modified version of an example from (Kuboň, 2001)), for example, three gaps occur (see the corresponding syntactic tree (5.9)).

(5.8) *Tu* *knihu* *jsem* *se* *mu*
this-FEM,SG,ACC book-FEM,SG,ACC am refl-ACC him-DAT
rozhodl *dát* *později.*
decided-LPART,MASC,SG give-INF later

"I decided to give him the book later." (Cze)

Two gaps are built by auxiliary (synsemantic) words, in particular *jsem* "I-am" and *se*, which is a reflexive pronoun. In the Lithuanian translation, there would be only one gap, containing the finite form *rozhodl* "decided", because both past tense and reflexivity are expressed synthetically in Lithuanian. We see that from the viewpoint of shallow parsing, synthetic languages are easier to analyze, as more linguistic categories are expressed at the level of morphology.

(5.9)



5.3 Morphological Similarity

Morphological similarity means similar structure of morphological hierarchy and paradigms such as case system, verbal system etc. In our understanding, Baltic and Slavic languages (except for Bulgarian and Macedonian) have a similar case system and their verbal systems are quite similar as well. Some problems are caused by synthetic forms which have to be expressed by analytical constructions in other languages (e.g., future tense or conjunctive in Czech and Lithuanian) and vice versa. The differences in morphology can be relatively easily overcome by the exploitation of a full-fledged morphological module for both languages of the language pair.

Similar morphological systems simplify the transfer. For example, Slavic languages (except for Bulgarian and Macedonian) have 6–7 cases. The case system of Baltic languages is very similar although it has been formally reduced in Latvian. Ambrazas (1996) gives seven cases for Lithuanian but there are in fact at least eight cases in the language (Vladarskienė, 2003). Nevertheless, the case systems of Slavic and Baltic languages are very similar which makes the languages closely related even across the border of different language groups.

Significant differences occur only in the verbal system, Baltic languages have a huge amount of participles and half-participles that have no direct counterpart in Czech. For example, the Lithuanian translation of an example by Gamut (1991) is given in (5.10):

(5.10) *Gimė* *vaikas,* *valdysiantis*
was-born-3SG child-MASC,SG,NOM rule-PART,ACT,FUT,MASC,SG,NOM
pasaulį
world-MASC,SG,ACC

"A child was born which would/will rule the world." (Lit)

The participle *valdysiantis* "which will rule" is used instead of an embedded sentence because Lithuanian has future participles. These participles have to be expressed by an embedded sentence in the contemporary Slavic languages.

5.4 Lexical Similarity

Lexical similarity does not mean that the vocabulary has to have the same origin, i.e., that words have to be created from the same (proto-)stem. What is important for shallow MT (and for MT in general) is semantic correspondence (preferably a one-to-one relation).

Lexical similarity is the least significant one from the viewpoint of MT since lexical differences are comparatively easily solved in the glossaries and general dictionaries.

Nonetheless there may be a need to extend the dictionaries by morphological information. Even for the language pair Czech-Slovak, there are some nouns that have different gender in both languages. For example, the Slovak translation of the Czech word *požadavek*-MASC "requirement" is *požiadavka*-FEM. This difference can be handled in the dictionary during the lexical transfer. In this phase, the target lemma is added to the corresponding feature structure and its gender is changed to the correct one. However, it is obvious that such changes of morphological properties can break agreement within a phrase if there is an agreement in the changed attribute between a head and its dependant, as in the following example (the correct translation—with the correct agreement—is given in brackets):

(5.11) *nový*-MASC *požadavek*-MASC
*nový-MASC požiadavka-FEM

"a/the new request" (nová požiadavka)

This is why another task of the transfer module is to modify morphological categories of dependants of translated items to preserve agreement. Another example (from Polish) is the agreement in case between prepositions and their governing nouns (this case is in opposition with the previous one, as during the lexical transfer, the case is changed in the feature structure of the head while in the previous example, a dependant was changed):

(5.12) *pro*-ACC *Joannu*-ACC
*dla-GEN Joannę-ACC

"for Joanna" (dla Joanny)

So we see that the lexical transfer also includes adapting morphological features gaining the necessary information for the dictionary. On the other hand, the structural transfer only operates at the level of (morpho)syntax.

6

Syntactic Relationships in Baltic and Slavic Languages

The previous chapter sketched the most important similarities between Baltic and Slavic languages at various levels of linguistic description. In this chapter, we attempt to present the most significant differences in the structure of noun and prepositional phrases and in verbal phrases.

6.1 The Morphosyntax of Baltic and Slavic Noun Phrases

Noun phrases (NPs) are basic building blocks of complements and adjuncts of predicates. The core of a prototypical NP is a noun, possibly extended, modified or restricted with complements and/or adjuncts. Of course, the core of an NP can be any language unit with nominal properties, such as certain kinds of pronouns, an adjective, infinite verb forms (infinitive, participle, quasi-participle etc.), an embedded sentence or a coordination of these. The structure of NPs is generally recursive, i.e., NPs may consist of simpler NPs or phrases that involve other NPs. According to Mayerthaler et al. (1998), nouns are universal with respect to the universal grammar, however the internal structure of NPs is language specific. In some languages, NPs may even be non-projective, especially in questions with an interrogative pronoun or in the case of topicalisation or dislocation, as in the following Polish and German examples, respectively:

(6.1) *Jaką* *kupiłeś* *książkę?*
which-FEM,SG,ACC buy-LPART,MASC,SG,2SG book-FEM,SG,ACC

"Which book did you buy?" (Pol)

(6.2) *Hirsche* *habe* *ich* *keine* *gesehen.*
deer-MASC,PL,ACC have-1SG,PRES I-NOM none-PL,ACC seen-PART,PAST

"I have seen no deers." (Ger)

NPs may also be predicative, for example in Russian, although this is diachronically only an effect of the ellipsis of the copula (e.g., Rus *Маша—его сестра* "Maša is his sister" or *У Маши прекрасные глаза* "Maša has beautiful eyes"). Moreover, NPs can build nominal sentences, such as headlines of newspaper articles or shortened answers to an wh-question.

NPs may be modified by prepositions to build prepositional phrases (PP). Such a modification traditionally changes the category of the phrase, although a simple

cross-linguistic comparative analysis shows that NPs are often represented by PPs in another language and vice versa. In some cases, this correspondence between NPs and PPs may be observed within one language. For example, the Lithuanian illative can be expressed by a PP with the preposition *į* "into". Sometimes, the corespondence between form and function is not straightforward, there may be splits and joins. For example, from the diachronic point of view, the Lithuanian allative, i.e., a bare case, can be expressed by a PP with the preposition *prie* or *pas*, depending on the animacity of the NP, as presented in the following examples:

(6.3) *miškan* → *į* *mišką*
forest-MASC,SG,ILL into forest-MASC,SG,ACC

"into the forest" (Lit)

(6.4) *miškop* → *prie* *miško*
forest-MASC,SG,ALL towards forest-MASC,SG,GEN

"towards the forest" (Lit)

(6.5) *tėvop* → *pas* *tėvą*
father-MASC,SG,ALL to father-MASC,SG,ACC

"to the father" (Lit)

Moreover, the same case of ambiguity can be observed for adessive. Thus the bare cases leave animacity underspecified whereas the semantically equivalent PPs leave directionality underspecified.

It has already been shown by Kuryłowicz (1949) that prepositions which modify an NP show an affinity to the category of case. There is also other evidence that a PP often acts exactly in the same way as an NP. In Lower Sorbian, for example, which lacks phonological length of vowels, accusative and instrumental of the noun *mama* "mother" collapsed in the form *mamu*. Nevertheless, in the context of a sentence there is usually no ambiguity since the instrumental is always used with a preposition (mostly *z* "with") and on the other hand, the preposition *z* cannot be used with an accusative. In our parser, prepositions depend on the NP without changing the category of the resulting constituent, i.e., the noun/prepositional phrase in the following sentences gets the same categorial status (namely NP) after having been processed by the parser:

(6.6) *Bydlím* *v* *centru.*
live-1SG,PRES in center-NEUT,SG,LOC

"I live in the center." (Cze)

(6.7) *Gyvenu* *centre.*
live-1SG,PRES center-MASC,SG,LOC

"I live in the center." (Lit)

6.1.1 Morphosyntactic Categories of Noun Phrases

In general, syntactic theories distinguish morphological and structural (abstract) morphosyntactic categories. In the generative grammar (for configurational languages such as English or French), all NPs get assigned a case. This assignment depends on the surrounding context, i.e., the grammatical function of the NP in its governor's phrase. Many languages, on the other hand, have declension with an inherent category of case, i.e., the case is expressed by a specific bound morpheme or in a similar way.

The Category of Case

All Baltic and Slavic languages, except for Bulgarian and Macedonian, have inherent cases of nouns, adjectives and some pronouns and numerals. Bulgarian and Macedonian have lost the nominal inflection as a result of their membership to the Balkan language union, i.e., through the influence of adjacent non-Slavic languages.

As for the morphosyntactic alignment, the languages we are examining belong to the nominative-accusative group. Nevertheless, Lithuanian shows an affinity to the antiergative system (cf. (Mayerthaler et al., 1998) for a more detailed explanation) which is thought to be a Finno-Ugric influence. Thus a typical impersonal sentence looks as follows:

(6.8) *Šiąnakt* *matoma* *mėnulis.*
tonight is-visible-NEUT moon-MASC,SG,NOM

"The moon is visible tonight." (Lit)

In the example above, *mėnulis* "moon" is the patient of the verb *matyti* "to see". The use of nominative is obligatory, the accusative (which would be used in Slavic language) appears only in some dialects (Zinkevičius, 1994, 1998).

Slavic languages with nominal inflection have 6–7 cases, Latvian has five cases (Forssman, 2001), Lithuanian eight (incl. illative which is contemporarily productive (Vladarskienė, 2003)). Macedonian and Bulgarian use analytical constructions to express grammatical functions of NPs in VPs, mainly prepositions and/or clitical head-marking pronouns. The following example illustrates how a direct object can be expressed in Macedonian:

(6.9) *јас* *ја* *гледам* *Марија*
I-NOM her-ACC see-1SG,PRES Mary

"I am seeing Mary." (Mac)

This example shows how the Macedonian NP *Марија* gets assigned a structural case, namely the accusative which is expressing that it is a direct object. It is noteworthy that only definite direct objects are marked at the verb. In the above example, *Марија* is a proper noun and therefore definite although no definite article is used.

Similarly, indirect objects are marked by a clitical pronoun (if definite or specific) and a PP:

(6.10) *му* *велам* *на* *Стојан…*
him-DAT say-1SG,PRES on Stojan

"I am saying to Stojan…" (Mac)

If both the direct and indirect object occur in the sentence, both clitical pronouns precede the verb:

(6.11) *му* *ја* *дадов* *книгата* *на* *брат*
him-DAT her-ACC gave-1SG,PAST book-FEM,SG,DEF on brother-MASC,SG
ми
me-DAT

"I gave the book to my brother." (Mac)

Thus we see that the "case" marker (a personal pronoun with an inherent case) is attached proclitically to the verb whereas the noun which has the function of object has no inherent case. This configuration allows for preserving free word order in some cases.

In Bulgarian, the assignment of structural cases is very similar. Nevertheless, whereas in Macedonian, the object doubling is obligatory, Bulgarian uses the pronominal marker to indicate a marked word order, e.g., topicalization of the object, as the following examples show:

(6.12) *Иван* *обича* *Мария*
Ivan love-3SG,PRES Mary

"Ivan loves Mary." (Bul)

(6.13) *Мария* *я* *обича* *Иван*
Mary her-ACC love-3SG,PRES Ivan

"Mary, Ivan loves." (Bul)

This difference between the two languages causes some sentences that are structurally ambiguous in Macedonian to be clearly expressed in Bulgarian.

6.1.2 The Category of Definiteness

The category of definiteness, according to Mayerthaler et al. (1998), belongs to the universal grammar. However, there are different means how to express this category. Baltic and Slavic languages have no articles, except for Bulgarian and Macedonian which have a definite article.

We can identify the following values of the category of definiteness:

definite Definite nouns are known to both the speaker and the listener.

(6.14) *Kreml* *je* *sídlem* *ruského*
Kreml-MASC,SG,NOM is residence-NEUT,SG,INS Russian-MASC,SG,GEN
prezidenta.
president-MASC,SG,GEN

"The Kreml is the residence of the Russian president." (Cze)

indefinite (specific) Indefinite specific objects are known to the speaker.

(6.15) *Včera* *jsem* *potkal* *jednoho*
Yesterday am meet-LPART,MASC,SG one-MASC,SG,ACC
kamaráda.
friend-MASC,SG,ACC

"Yesterday, I met a friend of mine." (Cze)

indefinite (non-specific) Indefinite non-specific objects are unknown and introduced at speech time.

(6.16) *Včera* *jsem* *našel* *na* *zemi*
Yesterday am found-LPART,MASC,SG on ground-FEM,SG,LOC
prstýnek.
ring-MASC,SG,ACC

"Yesterday, I found a ring on the ground." (Cze)

Specificity is usually not expressed explicitly except, for example, for Macedonian as in the following example from (Friedman, 2001):

(6.17) *Барав* *една* *марка,* *но* *не* *најдов.*
look-for-1SG,PAST one-FEM stamp-FEM,SG but NEG found-1SG,PAST

"I was looking for a stamp but I did not find any." (Mac)

(6.18) *Барав* *една* *марка,* *но* *не* *ја*
look-for-1SG,PAST one-FEM stamp-FEM,SG but NEG her-ACC
најдов.
found-1SG,PAST

"I was looking for a stamp but I did not find it." (Mac)

In Baltic and Slavic languages, the definiteness is reflected morphologically at the adjective. The so called short (nominal) forms are indefinite whereas the long (pronominal) forms express definiteness. Thus in Lithuanian, for example, there is a semantic difference between *mažas* "a small" and *mažasis* "the small".

6.1.3 Adjectival Agreeing Attributes

An NP can be modified by an adjective. In languages with adjectival and nominal case inflection, the adjective has to agree with its governor in gender, number and case. Among the languages we are investigating, only Bulgarian and Macedonian do

not have cases, however the adjectives still agree with their governors in gender and number.

In unmarked phrases, adjectives precede their governors, except for Polish where, for example, collocations (e.g., *szkoła handlowa* "business school") may have postponed adjectives. Otherwise, adjectives may be postponed to express emphasis. Given such a word order, the determiner (or a preposition) may be repeated to express the dependency relation (e.g., LSor *twój dom twój wóśćojski* "your father's house", Rus *у брата у старшего* "at the elder brother"). In the case of prepositional phrases we observe a parallel relation between the inherent case of a noun and the preposition as described by Kuryłowicz (1949).

Adjectives may also carry additional information, for example, the category of definitness as in Latvian and Lithuanian (cf. *baltas* vs. *baltasis* "a white/the white").

The rules for handling adjectives as agreeing attributes of a noun are schematically defined as follows:

N′ → A N′, ↑ CASE =↓ CASE, ↑ GENDER =↓ GENDER, ↑ NUMBER =↓ NUMBER, ↑ ADJ ∋↓

N′ → N′ A, ↑ CASE =↓ CASE, ↑ GENDER =↓ GENDER, ↑ NUMBER =↓ NUMBER, ↑ ADJ ∋↓

6.1.4 Non-agreeing Genitive Attributes

In Slavic languages, genitive attributes follow its governor in unmarked cases, whereas in Baltic languages, they precede the governing noun. Genitive possessive attributes have to be distinguished from partitive attributes that follow its governor, for example:

(6.19) *stiklinė* *pieno*
glass-FEM,SG,NOM milk-MASC,SG,GEN

"a glass of milk" (Lit)

Sometimes, a prepositional phrase can be used to express possessivity, for example:

(6.20) *žeńska* *wót* *mójogo* *bratša*
wife-FEM,SG,NOM from my-MASC,SG,GEN brother-MASC,SG,GEN

"my brother's wife" (LSor)

Macedonian mostly uses prepositional phrases to express possessivity, for example:

(6.21) *председателот* *на* *Македонија*
president-MASC,SG,DEF on Macedonia-FEM

"the president of Macedonia" (Mac)

However, partitivity is expressed without a preposition, i.e., only by means of word order (in combination with the semantic characteristics):

(6.22) *чаша* *вода*
glass-FEM,SG water-FEM,SG

"a glass of water" (Mac)

Nouns usually govern only one genitive attribute (which however, can be modified by another genitive attribute recursively). However, there can be more of them in marked cases, as in the following example:

(6.23) *královna* *krásy* *České*
queen-FEM,SG,NOM beauty-FEM,SG,GEN Czech-FEM,SG,GEN
republiky
republic-FEM,SG,GEN

"Miss of the Czech Republic" (Cze)

This example can be explained by the fact that the noun phrase *královna krásy* is a semantically tight word group—a collocation, which acts in the NP as an atomic unit.

Deverbal nouns may use noun phrases in genitive to express the subject or the object of the underlying process. Such constructions are often ambiguous, for example:

(6.24) *podpora* *otce/dětí*
support-FEM,SG,NOM father-MASC,SG,GEN/children-NEUT,PL,GEN

"support of the father/children" (Cze)

The rules for handling genitive attributes of a noun are schematically defined as follows:

N′ → N N′, ↓ CASE = genitive, ↑ GEN-ATTR ∋↓ (Baltic)
N′ → N′ N, ↓ CASE = genitive, ↑ GEN-ATTR ∋↓ (Slavic)

6.1.5 Prepositional Phrases as Attributes

Prepositional phrases can generally modify nouns as well as verbs. In this subsection we only consider prepositional phrases as attributes of nouns.

A special use of prepositions can be observed in Bulgarian and Macedonian in connection with verb phrases introduced by the particle *да*. These constructions are a solution of the fact that these two languages do not have infinitive as a distinct verb form.

(6.25) *без* *да* *дојдеш*
without that come-2SG,PRES

"without you coming" (Mac)

The rules for handling prepositional phrases as attributes of a noun are defined as follows:

N′ → PP N′, ↑ PREP-ADJ ∋↓
N′ → N′ PP, ↑ PREP-ADJ ∋↓

6.1.6 Appositions

There are two types of appositions, tight (e.g., (6.26)) and loose (e.g., (6.27)):

(6.26) *teta* *Jana*
aunt-FEM,SG,NOM Jane-NOM

"aunt Jane" (Cze)

(6.27) *můj* *soused,* *ředitel*
my-MASC,SG,NOM neighbour-MASC,SG,NOM director-MASC,SG,NOM
základní *školy*
elementary-FEM,SG,GEN school-FEM,SG,GEN

"my neighbour, the director of the elementary school" (Cze)

According to Eroms (2000), tight appositions are a special attribute type and the apposition may be prenominal or postnominal. Both the apposition and its head are inflected and agree in case. Moreover, Latvian and Lithuanian use non-agreeing genitive prenominal appositions, for example:

(6.28) *Lietuvos* *respublika*
Lithuania-FEM,SG,GEN republic-FEM,SG,NOM

"Republic of Lithuania" (Lit)

In the syntactic representation, this kind of apposition is equal to genitive attributes. Slavic languages use an agreeing tight apposition or an adjective instead, i.e., *Litevská republika* or *Republika Litva* "Republic of Lithuania".

According to Eroms (2000), loose appositions are comparatively independent modifiers of nouns or pronouns. They are similar to a parenthesis but they do not include a finite verb. Loose appositions are always postnominal. They can be also introduced by a conjunction, e.g., *jako* "as" in Czech, for example:

(6.29) *Já* *jako* *vedoucí* *rozhoduji* *o* *všem.*
I-NOM as director-MASC,SG,NOM decide-1SG,PRES about everything-LOC

"As a director, I decide about everything." (Cze)

Like 'normal' appositions, such constructions are syntactically loose with regard to the sentence context they occur in.

6.2 The Morphosyntax of Baltic and Slavic Verb Phrases

Verb phrases represent a higher level of syntax and there is also greater difference between surface and deep syntax as compared to nominal and prepositional phrases. What is extremely important is the different realization across Baltic and Slavic languages.

The following sections describe the verbal phrases in Baltic and Slavic languages with focus on their language specific realization.

6.2.1 Morphosyntactic Properties of Verb Phrases

The Category of Tense

Each process expressed by a verb contains a time factor. The temporal classification gets expressed by the grammatical category of tense. There are three temporal dimensions. All events that happened before the speech time are past events, all events that are happening during the speech time are present events and events that will happen after the speech time are future events. These three stages, called absolute, are always relative to the speech time.

Matrix sentences are usually formulated relatively to the speech time whereas processes in embedded sentences are temporally relative to the matrix sentence or to the superordinated embedded sentence. In such a case, we speak about relative temporal stages that express anteriority, contemporaneity and posteriority.

To express the complex relationship between temporality (at the semantic level) and the grammatical category of tense, one has to consider, besides the absolute and relative stages, the viewpoint of the speaker. Depending on the division of the time axis, provided by the speaker, one distinguishes between speech time, reference time and event time. The event time is the time point or interval the reported process happens at. The reference time is the time point which is being reported about. For example, in the following Lower Sorbian sentence, one can distinguish, from the viewpoint of the speaker, three temporal dimensions:

(6.30) *Tam* *se* *lětosa* *kulki* *raźili*
there REFL this-year potatoes-FEM,PL,NOM succeed-LPART,PL
njejsu.
are-not-3PL,PRES

"Potatoes did not grow well there this year." (Lsor)

- Speech time: now,
- Reference time: a past process, specified through the temporal adverb *lětosa*,
- Event time: the growing of the potatoes took place before looking at the process.

In the following, we give an overview of the most common tenses in Baltic and Slavic languages (the classification is based on (Starosta, 1992) and generalized to other researched languages).

Actual present Event, reference and speech time collapse at the present time point. Optionally, the actual present may be specified by adjuncts such as *now, even* etc. This tense may also describe processes that have begun in the past and continue to be active after the speech time. It can be built only with imperfective verbs and is not substitutable by any other tense.

Future present Both event and reference time are equal and follow the speech time. If the verb is imperfective, temporal adjuncts have to modify the event time.

Past present If speech and reference time are equal and follow the event time, and if the event time is close to the speech time or the result of the process is still

relevant in the speech time, then one can use a present form instead of the perfect. If event and reference time are equal and followed by the speech time, then one can use a present instead of past forms. Such use is called historical present and it is a part of a functional style. If a past matrix sentence is followed by an embedded sentence with a distinct process, then the tense of the embedded sentence depends on the temporal relationship between both processes. If they are parallel, present or past forms may be used (of imperfective verbs). If the embedded process is general as for tense, one has to use present forms (of both aspects). If the embedded sentence expresses an expectation, a wish, a command etc., present and future forms may be used.

General (atemporal) present Speech and reference time are equal and they are integrated in the event time, i.e., the event time is unspecified.

Perfect Tense

In Slavic languages, the most common past tense pattern is the one with an *l*-participle and an auxiliary *be*. In Common Slavic, it was the only compound past tense with a resultative (perfect) meaning, nevertheless it developed to a universal past expression in most of the languages (after the loss of simple past tenses). Bulgarian and Macedonian have reanalyzed this pattern to a new verbal category, the so-called re-narrative.

A couple of examples:

(6.31) *Mojca* *je* *prišla*
Mojca-NOM be-3SG,PRES come-LPART,FEM,SG

"Mojca has come." (Slo)

(6.32) *Наташа* *пришла*
Nataša-NOM come-LPART,FEM,SG

"Nataša has come." (Rus)

In example (6.32), the auxiliary is omitted, as it is usual in Russian. The absence of a finite verb in the Russian construction is the reason for the obligatory presence of the subject if it is expressed by a pronoun, as in examples (6.33) and (6.34):

(6.33) *prišel* *sem*
come-LPART,MASC,SG am-1SG,PRES

"I have come." (Slo)

(6.34) *я* *пришел*
I-NOM come-LPART,MASC,SG

"I have come." (Rus)

In Polish, the pattern is, in principle, the same, but the auxiliary is attached to an accented word (usually to the participle itself):

(6.35) *przyszła* *-m*
come-LPART,FEM,SG am-1SG,PRES

"I have come." (Pol)

There are more possibilities if the subject is present at the surface level: *ja przyszła-m* vs. *ja-m przyszła*.

In some Slavic languages, the auxiliary is omitted in the third person, for example Czech *já jsem přišel* "I came" but *on přišel* "he came" vs. Lower Sorbian *wón jo pśišeł*. In BCS[1], the auxiliary is omitted if the verb is reflexive: *on je video* "he has seen" vs. *on se šetao* "he has walked". Moreover, the order of clitics in the third person may differ, e.g., Slovenian *sem ga videl* "I have seen him" vs. *ga je videl* "he has seen him".[2] In Bulgarian, the absence of the auxiliary has a semantic impact.

In Lithuanian, there are two patterns of compound past tenses with *be* and an active participle: perfect and progressive.

(6.36) *esu* *atvažiavęs*
am-1SG,PRES come-PART,ACT,PAST,MASC,SG,NOM

"I have come." (Lit)

(6.37) *buvau* *bemiegąs* *kai…*
was-1SG,PAST sleep-PART,PROGR,MASC,SG,NOM when

"I was sleeping when…" (Lit)

In most West European languages, the so called possessive perfect is very frequent; it usually consists of the pattern expressing *to have* and a passive participle. The participle was originally governed by a noun and it has been reanalyzed according to the following scheme: *I have a seen car* → *I have seen a car* (i.e., the governing noun became the object of the participle). It is obvious that this construction could develop only for transitive verbs (in the early stage). This dichotomy can be observed, for example, in German, where two auxiliaries are used: *haben* "to have" for transitive verbs and *sein* "to be" for intransitive verbs. Occasionally, other possessive constructions can be used to express the agent, e.g., the adessive in the Belorussian Lithuanian (*manip jau visa padaryta* "I have already done everything")[3], the preposition *у* in Rus (*у нее в больнице лежано* "she was down in the hospital").

In Slavic languages, this pattern has developed especially in Macedonian, Cashubian and some Russian dialects (in the North-West). In Macedonian, we have, for example:

(6.38) *ја* *имам* *завршено* *таа* *работа*
her-ACC have-1SG,PRES finish-PART,PASS,PAST,NEUT,SG this-FEM,SG work-FEM,SG

"I have finished this work." (Mac)

[1] Bosnian, Croatian, and Serbian, formerly denoted as Serbo-Croatian

[2] with a marked word order here

[3] Since one says, for example, *manip du broliai* instead of standard Lithuanian *aš turiu du brolius* "I have two brothers" etc. (Vidugiris, 2004)

The participle has the impersonal (neuter) form. However, in some Bulgarian dialects, it agrees with the object, and this is what the Macedonian pattern developed from, cf. *ја имам завршена таа работа* (Koneski, 1965).

This construction started to develop in other Slavic languages as well. By Janaš (1976), for example, it is interpreted as a specific Lower Sorbian voice. In Polish, one can find a syntactic pattern with *mieć* "to have", nevertheless it did not develop into a new tense (yet) (Weydt and Kaźmierczak, 1999).

This pattern can also build whole paradigms (i.e., pluperfect, future perfect etc.), in Macedonian, for example, *имам/имав/ќе имам речено* "I have/had/will have said" etc.

In Lower Sorbian, one can say, e.g., *wón jo stanjony* "he is up", in many Polish dialects, the same pattern occurs too (e.g. *śniyg je už sleźóny* "the snow has already come down"). It has an active meaning (although formally passive) and it is in competition with active sentences (*wón jo stanuł*).

In Baltic languages, patterns with passive participles are only used to build the passive (except for specific impersonal constructions). On the border of these two patterns are modal expressions, for example:

(6.39) *Šis* *darbs* *bijo*
this-MASC,SG,NOM work-MASC,SG,NOM be-3SG,PAST
padarāms
do-PART,PASS,PRES,MASC,SG,NOM

"This work had to be done." (Lat)

The perfect tense has four functions:

1. Speech and reference time are equal and follow the event time. It can describe processes that are active up to the present time. Such a process is often a base for an immediately following present process. In this meaning, it is not substitutable with any other tense.
2. Event and reference time are equal and they are followed by the speech time. It denotes processes that happened in the past. In these sentences, the perfect competes with the imperfect tense.
3. The event time precedes the reference time, which is followed by the speech time. This configuration occurs if one describes a process in the past and wants to express a process that was already completed by then. The perfect competes here with the pluperfect.
4. The speech time precedes the event time that precedes the reference time. This is the perfect future.

Simple Past Tense

Originally, there were two simple past tenses in Common Slavic—aorist and imperfect. These have disappeared except in Bulgarian, Macedonian, Sorbian and literary BCS (cf., e.g., Macedonian *јас имав* "I had").

In Common Slavic, the aorist has been used to 'push a story forward' whereas the imperfect has been used to 'describe circumstances' (Trunte, 2005). This characteristic is somewhat simplified but it roughly describes the function of these verb forms. On the contrary, the compound past tense with an *l*-participle has been used as a resultative.

Baltic languages have preserved the functional opposition and the simple past tense is by far the most frequent one. For example, the sentence *Ieva atvažiavo* "Eve came" stands in oppostion to the sentence in (6.40):

(6.40) *Ieva* *yra* *atvažiavusi*
Eve-NOM be-3SG,PRES come-PART,ACT,PAST,FEM,SG,NOM

"Eve has come." (Lit)

However, this functional opposition has been lost in most Slavic languages that still use simple past tenses. In Lower Sorbian, for example, *mějach* "I had" and *som měł* "I have had" have identical meaning.

A specific pattern exists in some of the considered languages. It consists of a passive participle which governs a patient (in most cases) whereas there is no agent. The highest degree of grammaticality (among Slavic languages) can be observed in some Russian dialects. There are two basic patterns:

(6.41) *корова* *подоена*
COW-FEM,SG,NOM milked-PART,PASS,PAST,FEM,SG

"The cow has been milked." (Rus)

(6.42) *корову* *подоено*
COW-FEM,SG,ACC milked-PART,PASS,PAST,NEUT,SG

In (6.41), there is an agreement between the patient and the participle, i.e., this pattern is close to the 'regular' passive; in (6.42), there is no agreement, and the passive status of this pattern is not clear as discussed by Lavine (1999) (the patient is realized by the accusative).

Furthermore, there is a 'mixed' form:

(6.43) *корова* *подоено*
COW-FEM,SG,NOM milked-PART,PASS,PAST,NEUT,SG

Some other examples:

(6.44) *tutaj* *wybudowano* *most*
here build-PART,PASS,IMPERS bridge-MASC,SG,ACC

"A bridge has been built here." (Pol)

(6.45) *matyt* *jų* *neturėta*
evidently they-PL,GEN not-have-PART,PASS,IMPERS

"Evidently, there have been none of them." (Lit)

(6.46) *Jomu* *bu* *pomagane*
him be-3SG,PASS help-PART,PASS,NEUT,SG

"One has helped him." (LSor)

Whereas this pattern is rather dialectal in Russian, it is well established in Polish, Ukrainian and Lithuanian, although one has to bear in mind that the surface realization yields to rigid constraints: in Polish, there must be no agent, in Ukrainian, there must be a patient etc. (Lavine, 2005).

Event and reference time are equal and followed by the speech time. Therefore the simple past denotes completed processes, thus it is typically used in stories etc.

(6.47) *Anka* *źěšo* *pó* *wódu.*
Anka-NOM went-3SG,PAST for water-FEM,SG,ACC

"Anka went for water." (LSor)

Pluperfect

Some languages also have the pluperfect (past perfect). There are basically three patterns:

- perfect of *to be*+*l*-participle, for example, Polish *jam był przyszedł* "I had come",
- simple past tense of *to be*+*l*-participle, for example, Lower Sorbian *běch pšišeł* "I had come",
- perfect of *to be*+past participle (active or passive with an active meaning), for example, Lithuanian *buvau atėjęs* "I had come".

The event time is followed by the reference time that is in turn followed by the speech time. This tense occurs frequently in embedded sentences; the reference time of the embedded sentence is the event time of the matrix sentence.

The pluperfect can usually be substituted with perfect, if the temporal order of the described processes (*consecutio temporum*) can be derived from the context.

The Category of Aspect

The aspect is a typical category in Baltic[4] and Slavic languages.

The interplay between the aspect and other verbal categories is very complicated and cannot be explained here in detail. The following examples from (Levinson, 2005) show one of the semantic differences:

[4]The use of aspect in Baltic languages is slightly different from that in Slavic which leads some linguists to deny the existence of the aspect there, cf. for example (Račienė, 1999) for Lithuanian.

(6.48) *Он* *построил* *дом.*
he-NOM build-LPART,MASC,SG,PERF house-MASC,SG,ACC

"He has built a house." (Rus)

(6.49) *Он* *строил* *дом.*
he-NOM build-LPART,MASC,SG,IMP house-MASC,SG,ACC

"He was building a house." (Rus)

There is an important difference between Baltic and Slavic languages concerning the future tense. In Slavic languages, the future tense is periphrastic for imperfect verbs whereas it is synthetic in Baltic languages. The following two sentences, Czech and Lithuanian, have the same meaning:

(6.50) *Budu* *psát* *knihu.*
be-1SG,FUT write-INF book-FEM,SG,ACC

"I will write a book." (Cze)

(6.51) *Rašysiu* *knygą.*
write-1SG,FUT book-FEM,SG,ACC

"I will write a book." (Lit)

For the perfect aspect, the structure of the sentences is identical in both languages:

(6.52) *Napíšu* *knihu.*
write-1SG,FUT book-FEM,SG,ACC

"I will write a book (completely)." (Cze)

(6.53) *Parašysiu* *knygą.*
write-1SG,FUT book-FEM,SG,ACC

"I will write a book (completely)." (Lit)

The Predicativity of Verbal Phrases

Predicativity is the structural property of a verb phrase carrying the syntactic function of a predicate. Usually, the core of such a verb phrase is a final verb. From the syntactic point of view, a verb can be used predicatively, attributively or semi-predicatively.

The most common semi-predicative constructions are listed below:

Appositive participles (active or passive) have the same meaning as transgressives and are usually a combinatoric variant of them, for example:

(6.54) *Ratownik,* *gwałtownie* *obudzony…*
rescuer-MASC,SG,NOM suddenly wake-up-PART,PASS,PAST,MASC,SG,NOM

"The rescuer, woken up suddenly…" (Pol)

Transgressives, half-participles or quasi-participles express a secondary process and are usually equal to an embedded sentence.

The following example shows an absolute use of a transgressive (in this case, it is the title of a book):

(6.55) *Jadąc* *do* *Babadag*
going-TRG,PRES to Babadag

"Going to Babadag" (Pol)

Passive can be expressed by a periphrastic transgressive phrase:

(6.56) *Odpověz* *jsa* *tázán.*
answer-2SG,IMP be-TRG,PRES,MASC,SG ask-PART,PASS,PAST,MASC,SG,NOM

"Answer if you are asked." (Cze)

6.2.2 Non-canonical Cases of Morpho-syntactic Linking

This section briefly describes several constructions that link the two most important actants, actor and patient, differently across Baltic and Slavic languages and hence constitute a problem for MT.

Genitive of Negation

In some Baltic and Slavic languages, the patient is expressed by the genitive case when the verb is negated. This phenomenon does not occur only for finite verbs but also for the infinitive and transgressive (however, only for the active voice), e.g., Polish *nie znając języka* "without knowing the language", Lithuanian *nepirkti vaisių* "not to buy fruits" etc. The case shift acc → gen can also occur when the verb itself is not negated but the sentence contains a negative predicative adverb, e.g., Lithuanian *čia negalima pirkti knygų* "it is not possible to buy books here".

This phenomenon does not occur in languages which have been significantly influenced by German, such as Czech, Lower Sorbian or the former Lithuanian dialect in East Prussia (Zinkevičius, 1998).

Oblique Agents in Valence Frames

There is a group of verbs where the actor is expressed by an oblique or prepositional case even in the active voice. The patient is then usually expressed by the nominative case. Typically, the dative case is used, e.g., *mně se líbí toto město*; *мне нравится этот город*; *man patinka šis miestas* "I like this town" (in Czech, Russian and Lithuanian, respectively).

Russian, Latvian and some Lithuanian dialects lack the verb *to have* and possession has to be expressed by the verb *to be* with a special valence frame. In Russian, the prepositional case *у + genitive* is used to express the actor, e.g., *у меня есть дом* "I have a house". The verb is omitted if the possession is inalienable, e.g., *у нее синие глаза* "she has blue eyes". Latvian uses the dative, e.g., *man ir grāmata* "I have a book".

Some Lithuanian dialects use the adessive to express the possession, e.g., *broliep(i) trys vaikai* "the brother has three children". The patient is expressed by the nominative.

Passive

The passive is one of the voices used with transitive verbs. In Baltic and Slavic languages, it is expressed by periphrastic syntactic constructions. It is used quite often in analytical languages (such as English or French), but its usage in Baltic and Slavic languages is comparatively rare because the sentence perspective which is the main reason of its use in the mentioned West European languages, can be expressed by the word order. The passive is used mainly if the actor is expressed marginally or not at all. In passive sentences, the actor is expressed by an oblique or prepositional case (usually instrumental in Slavic languages, genitive in Lithuanian) and the subject mostly expresses the patient. In passive sentences, it is not possible to express the actor in Latvian (Forssman, 2001). Lithuanian can build passive forms also for intransitive verbs, e.g., *tėvo seniai sergama* "the father is sick for a long time" (cf. the active sentence *tėvas seniai serga*).

Usually, the auxiliary verb *to be* is used in passive constructions, e.g., Czech *kniha je čtena* "the book is being read". In Lithuanian, the auxiliary verb is often omitted: *laiškas (yra) rašomas* "the letter is being written". Polish uses the auxiliary verb *zostać*, e.g., *zamek został zniszczony* "the castle has been destroyed".

A special case of the passive is the so called *statal passive*, e.g., Czech *dům je postavený* "the house is built", Lithuanian *prekė yra užsakyta* "the goods is ordered". Another special case of passive, the so called *mediopassive*, is described in the next subsection.

Mediopassive

The mediopassive (reflexive passive) is present only in Slavic languages and usually expresses a process without an actor (more precisely, with a general actor), e.g., Russian *ети машины производятся в Москве* "these cars are produced in Moscow". Baltic languages use normal passive (Lithuanian *šios mašinos gaminamos Maskvoje* "these cars are produced in Moscow") or a completely different construction (Russian *ети книги хорошо читаются* "these books are easy to read" vs. Lithuanian *man gerai skaityti šias knygas* "these books are easy to read").

Participles

Participles are verb forms that act as nouns (mostly adjectives, sometimes substantives). They behave morphologically as nouns but they have their own valence frame (which depends on the voice). Moreover, participles distinguish the tense (present and perfect in Slavic languages, up to four tenses in Baltic languages). The existence of a concrete participle form also depends on the aspect. The noun which gov-

erns the participle is linked to its actor (active participle) or patient (passive participle; see above for exceptions), e.g., Russian *читающий мальчик* "the reading boy" but *читаемая книга* "book being read". Tenses are distinguished morphologically: *читающий* "who is reading now" vs. *читавший* "who was reading", analogically for passive participles: *читаемый* "what is being read now" vs. *читанный* "what has been read". The linking of the remaining participants is analogical to the linking of finite verbs (of the same voice).

Participial phrases can be usually expressed by embedded sentences while preserving the meaning (e.g., Russian *читающий мальчик/мальчик, который читает* "the boy who is reading", Lower Sorbian *wuknjacy student/student, ako wuknjo* "a student who is learning"); the choice depends on the type of the text and other stylistic criteria. In BCS, there are no active present participles, thus only embedded sentences can be used (e.g., *muškarac, koji radi* "working man"). Macedonian has no participles any more except those used in periphrastic tenses which cannot be used as an attribute (e.g., *сум јаден/имам јадено* "I have eaten"; cf. Bulg. *падаща звезда* vs. Mac. *ѕвезда што паѓа* "a falling star").

Except the common participles, there are also modal participles in some languages, e.g., the participle of possibility:[5] Czech *vyslovitelný* "pronounceable", Polish *wymawialny* "pronounceable".[6] Lithuanian has the participle of necessity, e.g., *mokėtinas* "which has to be paid".[7]

Nominalization

Many nouns derived from verbs (e.g., verbal substantives) have their own valence frame. There are no precise rules how to assign the actor or patient, the linking depends on the inherent meaning of the verb the noun is derived from. Let us have a look at an example. The Czech phrase *ošetření lékaře*[8] can mean either "examination of the physician" or "treatment by the physician". On the other hand, the phrase *ošetření pacienta lékařem* "investigation of the patient by the physician" is not ambiguous as the linking is specified clearly by a different case. Common knowledge can help to disambiguate the meaning even if the verb is transitive, e.g., Lithuanian *miesto užkariavimas* "the conquest of the town". This type of ambiguity does not occur for intransitive verbs, of course, because they have no direct object, e.g., Lithuanian *ugnies užgesimas* "extinction of the fire".

Nominalized constructions often do not have a strictly equivalent verbal expression (such as an embedded sentence) because they usually lack some morphological

[5] Some languages which do not have this special participle can express the possibility by common participles.

[6] cf. common participles, e.g., Russian *выговариваемый* "pronounceable", Lithuanian *ištariamas* "pronounceable"

[7] cf. the German gerundive, e.g., *die zu bezahlende Rechnung*

[8] Only one actant is expressed (by the genitive case).

categories (e.g., tense or gender) which can cause an ambiguity. The Czech phrase *po příchodu otce* "after the father's arrival" can be equivalent to embedded sentences *poté, co otec přišel* "when the father came" or *poté, co otec přijde* "when the father will come", thus the temporal relation is underspecified in the nominalized phrase. Similarly, the phrase *po jejich příjezdu* "after their arrival" does not clarify the gender whereas the embedded sentence with the (almost) same meaning does: *poté, co přišli/-y/-a* "when they (masc./fem./neut.) came". The choice depends on the context.

Transgressive

Transgressive is a special verb form which distinguishes the tense and, in some languages, also gender and number. In Slavic languages, two morphologically different transgressive forms express the temporal relationship of the process represented by the verb. Baltic languages even have a future form and an iterative form (e.g., Lithuanian gerund[9] forms of the verb *kalbėti* "to speak" are *kalbėjus, kalbant, kalbėsiant, kalbėdavus*). Although it is usually possible to express the transgressive phrase by an embedded sentence, its exact meaning depends on the context (the meaning of the transgressive itself is vague). The contemporary transgressive represents mostly a secondary process, the anterior transgressive expresses a process which has finished before the beginning of the process of the main verb (cf. German embedded sentences with *nachdem*), e.g., Russian *он едет на автобусе читая газету* "he is going by bus while reading the newspaper", *ушед попрощавшись* "he left after the farewell", Upper Sorbian *hólc dźěše spěwajo domoj* "the boy went home while singing", BCS *imajući u vidu* "having in mind". Negated transgressive phrases are typically similar to embedded sentences with *although* or *without*, e.g., Russian *поехал в Америку не зная английского* "he went to America, although he does not speak English", *исчезнул, ничего не сказав* "he left without saying anything".

Baltic gerunds can have their own actor. In such a case, the actor is expressed by the dative case, e.g., *mes išvažiavome saulei tekant* "we left by sunrise" (literally: "when the sun was rising").

NB: Some transgressives (mostly in idiomatic expressions) are used without agreement, as the following examples show: Czech *chtě nechtě* "necessarily" vs. *chtíc* "wanting (fem./neut.)", Lithuanian *tiesą sakant* "to tell the truth" vs. *sakydamas tiesą* "telling the truth".

Accusativus *cum participio*

This construction is used in Baltic languages with a small class of verbs, such as *to see, to hear* etc. It consists of a verb from this class and a participle which depends on its patient, e.g., Lithuanian *aš girdėjau tave kalbantį per radiją* "I heard you speaking in the

[9] Baltic languages distinguish *gerunds* (which represent a process with a general or distinct actor) and *half-participles* (their actor is the subject of the main verb).

radio". Instead of the participle, an infinitive is used in Slavic languages. Because the participle has to agree with its governor (i.e., the patient of the main verb) which is the actor of the participle, we can conclude that it depends on it (syntactically). The participle has its own valence frame (see above).

Parasitic Infinitival Complements

Infinitival complements are used with autosemantic verbs in Baltic languages. Such infinitives can be used with many verbs (if it is semantically acceptable) and share an actant with it, either the actor or the patient. The shared actant is usually the addressee of the main verb, e.g., Lithuanian *motina įpylė pieno vaikams gėrti* "the mother gave milk to the children to drink". In such sentences, the addressee *vaikams* "to the children" (dative) is the actor of the infinitive *gėrti* "to drink". The other possibility is to express the patient of the infinitive by the dative case, e.g., Lithuanian *žodis 'gudas' vartojamas baltarusiams vadinti* "the word 'gudas' is used to denote Belorussians"; the actor of the infinitive *vadinti* "to denote" is in dative. The choice whether the shared actant is an actor or patient, depends on the inherent meaning of the verb and on the context. Usually, it is the actor if the patient (direct object) is shared as well, as *pieno* "milk" in the first example, otherwise it is the patient[10] (as in the latter example). In Slavic languages, this construction has to be expressed by an embedded sentence or nominalization.

Debitive

Latvian has a special verb form called debitive which is used to express the alethic (objective) necessity, cf. the example from (Forssman, 2001) *man jāpērk maize* "I have to buy bread". The actor is expressed by the dative case and the patient by the nominative case (except for pronouns of the first and second person which have the accusative form). The debitive of the perfect and future tense is built periphrastically, using the auxiliary verb *to be*, e.g., *viņai bija jāstrādā* "she had to work".

The debitive is semantically identical with infinitival constructions of obligation in some languages, for example, Polish *Trzeba mu iść do domu* "he has to go home", which is described in the next subsection.

Obligative

The obligative infinitival construction is used in some Baltic and Slavic languages to express the modality similar to English *shall, should*. The infinitive form of the autosemantic verb is used and the actor is expressed by the dative case, e.g., Russian *что мне тепер делать* "what should I do now", Lithuanian *ką man dabar daryti* "what should I do now", Polish *gdzie nam ojca szukać* "where should we look for the father".

[10] The actor is general in this case.

The other participants of the verb are linked as usual (depending on the voice etc.).

Some languages express the obligative by a modal verb, i.e., in the same way as English or German, e.g., Czech *mám přijít včas* "I have to come in time". Lithuanian can use both variants, e.g., *man eiti namo* vs. *aš turiu eiti namo* "I have to go home".

Supinum

This special form occurs in Baltic languages, especially in some dialects (Zinkevičius, 1994), as well as in Lower Sorbian and Slovenian. Supinum expresses a goal, e.g., *jis išėjo grybautų* "he went to pick mushrooms". The patient of transitive verbs is expressed by the genitive case[11] (Ambrazas, 1996), e.g., *Jonas siunčia čigoną malkų atneštų* "Jonas sends the gipsy to bring wood". The main verb can even be omitted (Ambrazas, 1996), e.g., *Algirdai, vėžimų krautų!* "Algirdas, (go to) load the wagons!". In Lithuanian, he supinum form can be expressed by the infinitive using the same syntactic structure.

In this chapter, we have given a selective overview of the syntax of Baltic and Slavic languages. As one can see, the differences, especially at the verbal level, are sometimes comparatively big, thus, unfortunately, not all of them can be handled by a shallow parser. In the following chapter, we present a parser which is capable of dealing with some of them, in particular at the lower level of NPs and PPs, which helps to improve the automatic translation at least in a local context.

[11] Is seems that not only for supinum but also if an infinitive is used to express a goal, e.g., Lithuanian *atvažiavau tavęs pasitikti* "I came to pick you up".

7

Partial Parser for Baltic and Slavic Languages

In this chapter, we describe the parser module and grammar architecture for shallow processing of Baltic and Slavic languages.[1]

7.1 Tasks of the Parser

There was no syntactic parser in the original system *Česílko*. This module has been added to the translation process to deliver information about the sentence structure to the transfer module so that language specific structural properties could be handled and translated properly. Without the parser, morphological differences have only been considered, which is, of course, not sufficient in general. Hence, the parser provides an add-on value which is supposed to improve the translation. If the source sentence is left untouched by the parser (because it is too short or too complex), the system translates it as if there was no syntactic parsing.

The parser uses a hand-written grammar which consists of a set of context-free rules that are written in a declarative form. The output of the parser is a set of c-forests.[2] It is important to mention that a c-forest does not represent the structure of the sentence as such but a concrete rule application sequence. Before being passed to the transfer module, c-forests are automatically converted to d-forests.[3] Thus the final result of the parser is a d-forest or a set of d-forests if the parsed sentence is ambiguous.

The parser is not supposed to parse whole sentences. Of course, if the syntactic structure of the sentence is quite simple, the result will be one tree (or set of trees) covering the whole sentence. Nevertheless, in most cases, the result is a set of trees which only represent fragments of the sentence. One reason for such behavior may be the non-projectivity which occurs quite often in languages with free word order. But projective sentences also may only be parsed partially since the grammar focuses on the level of noun and prepositional phrases. The coverage of verbal phrases is rather

[1] Of course, the parser can be used for other language families as well, with appropriate grammar rules.

[2] By a forest, we mean a set of constituent trees which represent fragments of the parsed sentence and span it completely.

[3] A d-forest is a set of dependency trees which have been created by contracting the vertical edges of a c-tree.

small, the rules on this level are only meant to capture syntactic constructions which may cause serious problems in the target sequence.

7.1.1 The Computational Formalism

We use a transformational formalism which is based on a chart parser similar to Q-Systems, designed and first implemented by Colmerauer (1969). What is very important is the fact that the derivational process is context-free (in the sense of Chomsky's hierarchy) which has the crucial consequence for Slavic languages that it is not capable of dealing with non-projective constructions (at least not directly).

The input of the parser can be morphologically ambiguous. In such a case, the parser tries to use all provided data to construct a complete tree. If it succeeds, all complete trees comprise the result set whereas all input items which are not contained in a complete tree, are discarded.

Theoretically, it would be necessary to parse the whole sentence in order to disambiguate it morphologically. Even then, some words may keep more than one morphological tag (due to case syncretism). In case of shallow parsing only, the morphological ambiguity seems to be one of the most serious problems. The best case scenario would be to get a disambiguated input. Unfortunately, at the moment the only possibility is to use a stochastic tagger which introduces errors and makes it impossible for the parser to recognize some dependencies. As has been shown by Žáčková (2002), it is not possible to disambiguate Czech texts by means of shallow rules only.

7.2 Main Principles of Parsing Rules

As usual in unification-based grammars, each rule is associated with a condition (constraint) on feature structures and the rule applies only if this condition is satisfied.

A typical example of a linguistically motivated condition is the agreement of morphological categories between the governor and its dependant. For example, an adjective which depends on a noun has to agree with it in gender, case and number. We understand the term *agreement* in broader sense, i.e., a dependant agrees with its governor if a set of conditions, which are defined for the particular type of syntactic construcion, is satisfied. In most cases, the conditions are simply equivalences of category values, as in the following phrase:

(7.1) *mladší* *sestře*
younger-FEM,SG,DAT sister-FEM,SG,DAT

"to the younger sister" (Cze)

Nevertheless, the condition may be more complicated sometimes, for instance, in Polish noun phrases if the governor is in dual form:

(7.2) *czarnymi* *oczyma*
black-NEUT,PL,INS eye-NEUT,DUAL,INS

"with black eyes" (Pol)

(7.3) *w* *swoim* *ręku*
in his-FEM,SG,LOC hand-FEM,DUAL,LOC

"in own hands" (Pol)

Another example can be found in Russian:

(7.4) *два* *больших* *города*
two-MASC,NOM big-MASC,PL,GEN town-MASC,SG,GEN

"two big cities" (Rus)

Another example concerning non-trivial agreement between subject and verb (possible, for example, in Slovenian):

(7.5) *Slovenci* *volimo…*
Slovenians-MASC,PL,NOM vote-1PL,PRES

"we Slovenians vote for…" (Slo)

Apart from rules used to build syntactic trees, we use some tricks in our parser. The aim of these tricks is to modify the chain graph or to control the parsing process. Two such rules are described in the following subsections.

7.2.1 Chain Link (shackle)

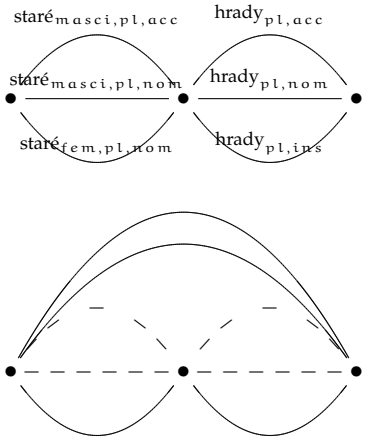


Figure 7.1: Example of NP analysis without a shackle

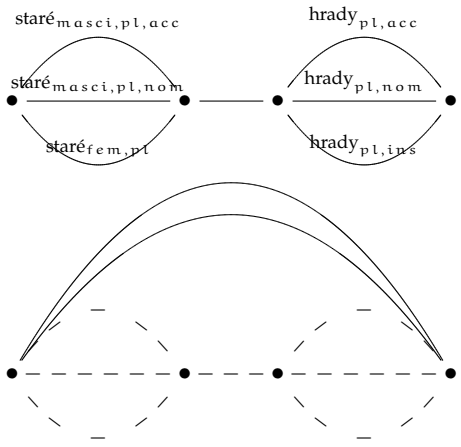


Figure 7.2: Example of NP analysis with a shackle

As has been already mentioned, the input of the parser is often morphologically highly ambiguous. One of the tasks of the parser is to disambiguate the sentence (or at least to lower the ambiguity). Let us consider the sentence *Starý hrad se tyčí nad řekou* "The old castle towers over the river". The phrase *starý hrad* is morphologically ambiguous (nominative and accusative). If this phrase has been recognized as the subject of the main verb, we know that the case is nominative in this context. And since there is no other reading where it would be accusative, we want to remove this wrong reading. In fact, it is removed automatically by the algorithm of the parser. But what would have happened if we had the bare phrase *staré hrady*? There are two possible readings (nominative and accusative) which cannot be resolved due to lack of context. Nevertheless, there are still other meanings for each of the words independently (disregarding the dependence between them). In this case, these edges will not be removed although the parser has analyzed the phrase. This is one negative property of the parser framework which has to be solved explicitely. We use a simple workaround: between edges which represent one word of the input sentence, we insert a new edge (*shackle*) that links bunches of edges. If there is at least one analysis which connects two words, the parser marks the shackle as used, i.e., it will be removed during the cleaning phrase (see Section 7.3). As an effect of this, the 'wrong' edges do not lie on a valid path in the multigraph any more and will be deleted as well,

as can be seen in Figure 7.2 (the adjective would have more morphological meanings; for the sake of simplicity, the multigraph contains only one edge with different gender).

It is obvious that if we modify the multigraph by adding 'shackles' between edges labelled with feature structures, we also have to modify all rules accordingly.

7.2.2 Elimination of Identical Results

The application of rules to the multigraph is non-deterministic. As a result, the application of several different sequences of rules may lead to identical results, as illustrated in the following example:

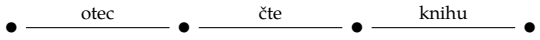


Figure 7.3: Example of a sentence with duplicate parses

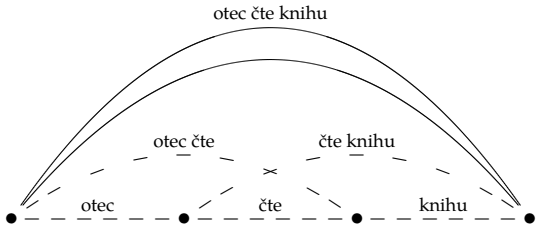


Figure 7.4: Chain graph with new edges

There are two possible parses:

1. The rule identifying direct objects is applied first, the rule identifying subjects is applied afterwards.
2. The rule identifying subjects is applied first, the rule identifying direct objects is applied afterwards.

Theoretically, we would get two edges spanning the whole sentence and labelled with identical syntactic structures (see Figure 7.4). In our implementation of the parser, this kind of duplicity is recognized automatically to avoid exponential explosion.

7.3 Multigraph Clean-up and Further Optimization

As long as a rule can be applied to the multigraph, edges are added to it but no existing edge is removed. The new edges represent (are labelled with) intermediary feature structures that may be used in further parsing or be candidates for the final result. Once the multigraph cannot be extended by any rule (according to the used grammar), the intermediary edges need to be discarded from the multigraph since we want only the most complex feature structures to be processed in the transfer phase. This clean-up is somewhat similar to garbage collection in programming languages with automatic memory management.

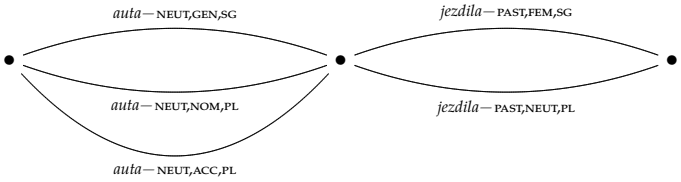
As an example, let us consider the following Czech verb phrase as the input of the parser:

(7.6) *auta* *jezdila*
cars-NEUT,NOM,PL move-LPART,NEUT,PL

"The cars moved/were moving." (Cze)

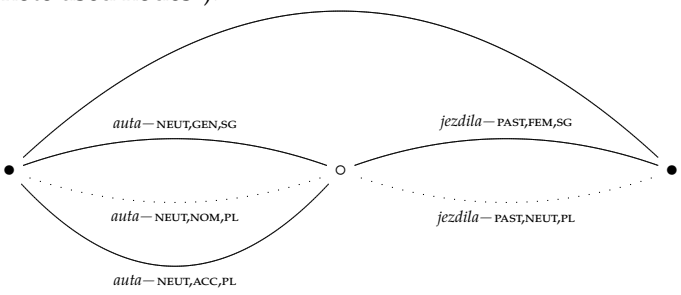
The input of the parser is the following morphologically preprocessed multigraph (the multisets of edges between the same pair of nodes reflect the morphological ambiguity of a word form):

(7.7)



One rule will be applied to this multigraph. Namely the one that attaches a noun in nominative (the subject) to its predicate (a resultative participle in this case). The following multigraph is the result of the syntactic analysis (dotted lines denote used edges, circles denote used nodes[4]):

(7.8)



[4]We define the used node as a node that has at least one used edge to the left and at least one used edge to the right.

Now we need to get rid of all obsolete edges:

1. First of all, we remove all used edges (denoted by dotted lines).
2. We remove all edges which start or end in a used node (i.e., the edges that reflect morphological variants of a used edge which are morphologically misanalyzed in the given context according to the used grammar).
3. For each path p from the initial node to the end node, we calculate the number $u(p)$ of used edges it contains. Then we assign each edge e the score $s(e) = \min_{p \in P} u(p)$. The score for the whole graph is defined as $s = \min_{e \in E} s(e)$. Finally, we remove all edges where $s(e) > s$.[5]

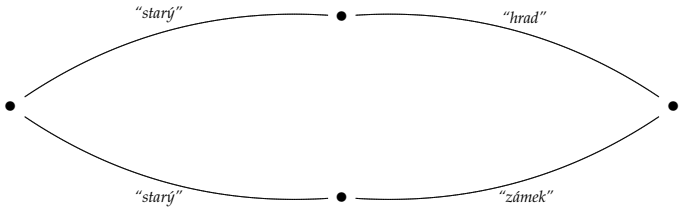
The last step ensures that every edge which remains in the multigraph lies on a path from the initial node to the end node. The resulting graph will be processed by the transfer module and at the same time, all complex feature structures (that represent syntactic trees) are syntactically synthesized (the transfer is described in Chapter 8).

Processing long sentences may result in very large multigraphs with the number of edges growing exponentially. If we had to translate the Russian phrase *старый замок* "old castle" into Czech, the transfer would give the following two features structures:

(7.9) $$\begin{bmatrix} \text{"замок"} \\ \text{ADJ} \quad [\text{"старый"}] \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} \text{"hrad"} \\ \text{ADJ} \quad [\text{"starý"}] \end{bmatrix}, \begin{bmatrix} \text{"zámek"} \\ \text{ADJ} \quad [\text{"starý"}] \end{bmatrix} \right\}$$

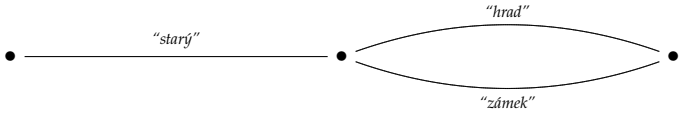
The syntactically synthesized multigraph would be as follows:

(7.10)



As the two edges with the feature structure for the adjective *starý* are identical, we can optimize the spatial complexity of the multigraph by contracting identical edges that have at least one common node. For the discussed example, we would get:

(7.11)



We call this process *compacting* the multigraph. It is obvious that in complex multigraphs, the number of edges can be lowered significantly. Immediately before mor-

[5] If there is at least one path from the initial node to the end node consisting only from unused edges then the algorithm is equal to the one described in (Colmerauer, 1969), i.e., all used edges are deleted as well as edges that do not belong to a path from the initial node to the end node.

phological synthesis, the optimization can be even more efficient if we do not contract only edges with identical feature structures but also with identical surface form in the target language (since there is an extensive syncretism in Slavic laguages).

7.4 Using the Parser in a Production Environment

The parser described in the previous sections of this chapter is written in a high-level language (Objective-C) which is more comfortable for the developer to use since the focus lies on linguistics. For grammar development and testing, the performance and resource consumption of the compiled code is not an issue. However, the performance is important for the processing of large texts while the resource consumption (the memory footprint) is crucial for the use of the parser on resource-restricted devices such as PDAs and smartphones. In this section, we briefly discuss a possible optimization of the parser.

We have tried to optimize the parsing process in the way that the rules are indexed by type signature, i.e., the concatenation of type names of all feature structures on the left-hand side of the rule. This optimization saved approximately 50% of processing time because the parser did not try to apply all rules on each subchain of the graph (only rules taken from the index for the particular subchain were considered to be applicable). Nevertheless, we wanted a much faster optimization and also a lower memory footprint. It turned out that transforming the grammar and the input into the Q-Systems format is a good solution.

The Q-Systems are significantly faster than the FS-based implementation of the parser mainly due to the different data structure used in unification. While the FS-based implementation unifies general feature structures, the Q-Systems use trees, thus the unification is similar to the unification of compound predicates in Prolog which makes it significantly faster.

Feature structures in grammar rules and in the input must meet several conditions in order to be transformable to the Q-Systems format. First of all, they must be typed and each type must be assigned a set of attributes the feature structure can contain. Another condition is that the order of attributes declared for a type is fixed. Finally, variables used as attribute values in feature structures may only contain atomic values or embedded feature structures.

Each feature structure is converted to a tree. The root of the tree is labelled by the type name of the feature structure while the sons of the root correspond to attribute values. The order of these nodes is the same as the order of attributes in the declaration for the particular type and all its supertypes. The structure of the rules remains the same including the 'shackles' (see Section 7.2.1). Attributes declared for a type that are not contained in a feature structure (and thus behave like free variables in Prolog) are represented by unique variables in the corresponding Q-Systems rule. It is obvious that type names and atomic attribute values must conform to the syntactic

rules of the Q-Systems. Variables are directly converted to tree-like variables in the corresponding tree and they get the same name.[6]

Let us consider the following type declarations (taken from a grammar for named entity recognition):

```
type sign
end

type shortdate
    prototype sign
    atomic day
    atomic month
end

type date
    prototype shortdate
    atomic year
end

type dateshorttime
    prototype date
    atomic hour
    atomic minute
end

type datetime
    prototype dateshorttime
    atomic second
end

type precisetime
    prototype datetime
    atomic millis
end
```

Each type has a unique name and a prototype (i.e., its supertype, except for the most general type "sign"). The type is assigned a list of attributes containing all attributes of its supertype followed by the declared (additional) attributes. The order of the attributes is not significant for the person who is writing a grammar, it is used only for the transformation of the feature structures. It is obvious that the same type declaration must be used to transform the rules and the input.

[6]The used implementation of the Q-Systems allows for using named variables (see below) while the originally Q-Systems designed by Colmerauer (1969) only allowed for indexed variables.

Let us consider the following feature structure of the aforementioned type *date*.

$$\begin{bmatrix} \textit{date} & \\ \text{DAY} & 23 \\ \text{MONTH} & 5 \\ \text{YEAR} & 2008 \end{bmatrix}$$

This feature structure would be automatically translated to the following Q-tree:[7]

```
DATE(23, 5, 2008)
```

If the structure would have the same content but the type *datetime*, it would be transformed to (the identifiers starting with I∗ are variables):

```
DATETIME(23, 5, 2008, I*ANONYMOUS1, I*ANONYMOUS2, I*ANONYMOUS3)
```

Since the attributes HOUR, MINUTE and SECOND are not listed in the feature structure, they are considered to be underspecified and we have to introduce anonymous variables to represent their values so that the unification works correctly. The name of the anonymous variable is generated automatically so that it is unique.

The interpreter of Q-Systems is implemented in C++ and it is equivalent to the original Q-Systems designed by Colmerauer (1969) except for the following extensions:

- The variables can be named, while in the original Q-Systems they could only be indexed. The name must be alphanumeric.
- If a rule has been successfully applied, the interpreter does not add the new subchain to the graph if there already is an identical subchain at the same position.
- The result of the parser is an empty graph if there is no path from the initial node to the end node in the final graph, after all used edges have been removed (the result of the original Q-Systems was the initial graph instead).

We have tested the aforementioned optimization on 1,000 text documents (most of them containing more than 200 words) with a grammar for named entity recognition. The processing time improved from 33 minutes to less than 4 minutes with a ten times smaller memory consumption.

[7] The interpreter of Q-Systems is not case sensitive thus we can use capitals to denote types in the Q-grammar.

8

Transfer and Syntactic Synthesis

Transfer and syntactic synthesis are performed jointly in one module. The task of the transfer module is to adapt complex structures created by the parser, which cover the whole source sentence continuously, to the target language lexically, morphologically and syntactically. In the following sections, we describe the phase of the lexical transfer and the structural transfer, the latter being split further into structural preprocessor and syntactic decomposer.

8.1 Lexical Transfer

The aim of the lexical transfer is to 'translate a feature structure lexically', i.e., the lemmas associated with feature structures are translated. Morphological features may be adapted as well where appropriate.

The following is a fragment of the dictionary used in lexical transfer (Czech-Slovenian):

(8.1)
```
hvězda| zvezda
dodat| dodati
kůň| konj
strom| drevo| gender=neut;
```

Let us have a brief look at the last line of the example. The Czech noun *strom* "tree" is in masculine gender while the gender of its Slovenian counterpart *drevo* is neuter, therefore there is the additional information *gender=neut* which instructs the transfer module to adapt the feature *gender* of the corresponding feature structure so it can be correctly synthesized morphologically.

8.2 Structural Transfer

The task of the structural transfer is to adapt the feature structures of the source language (their properties and mutual relationship) so that the synthesis generates a grammatically well-formed sentence with the meaning of the source sentence. It is worth noting that the well-formedness can generally be guaranteed only locally for the part of the sentence which the feature structure covers.

8.2.1 Transfer Directives

When changing the structure, one may do one of the following:

1. Change values of atomic features in the feature structure, add atomic features with a specific value or delete some atomic features.
2. Add a node to the syntactic tree.
3. Remove a node from the syntactic tree.

There are two types of structural changes:

Preprocessing of feature structures Such changes are performed prior to the lexical transfer.

Decomposition of feature structures These changes are performed after the lexical transfer and build up the syntactic synthesis.

All possible directives for the transfer module are listed in Table 8.1. The values in the column *Rule* say which kind of rules the directive applies to: *d* means decomposition rules and *p* means preprocessing rules. An asterisk means that the directive can be used in both types of rules. The values of attributes in a feature structure can be atomic or variables (alphanumeric identifiers beginning with $). A directive can succeed or fail. For example, the directive which represents unification succeeds if the corresponding feature structures can be unified, and fails if the unification fails. A rule is applied when all its directives succeed. The empirically composed set of rules for the language pair Czech-Macedonian consists of 9 decomposition rules and 11 preprocessing rules.

Let us give a couple of examples of transfer rules. The following rule is used to translate a preposition which requires a different case in the target language. In the feature structure of the noun that governs the preposition, its case is changed to the correct one.

```
(
preproc
(head= ((type word) (pos n)))
(hasChildren (prep))
(child= ((type word) (lemma u-1) (case gen)))
(lexChild ((lemma pri) (case loc)))
(copyup (case))
)
```

The following rule adds an auxiliary to an *l*-participle in the third person which may be required, for example, in the translation from Czech to Slovenian.

```
(
preproc
(head= ((type word) (pos verb) (vform lpart) (person 3)
 (number $number)))
(noChildren (aux))
(newChild ((gfunc aux) (relorder -9) (lemma být) (pos verb)
 (vform fin)(tense pres) (person 3) (number $number)))
)
```

Directive	Rule	Arguments	Description
attName	d	attribute name	Succeeds if the attribute name of the detached child in the feature structure of its governor is equal to the argument of the directive.
child=	*	FS	Unifies the child (for decomposition rules, the detached child, otherwise the first child identified by the directive *hasChildren*) with the given feature structure.
copydown	*	list of att. names	Copies the given attributes from the head to the child.
copyup	*	list of att. names	Copies the given attributes from the child to the head.
direction	d	l/r	Succeeds if the child is to the left or to the right of its governor.
hasChildren	*	list of att. names	Succeeds if the head contains all listed attributes.
head=	*	FS	Unifies the head with the given feature structure.
lexChild	p	FS	The same as *rewriteChild* but the feature structure will not be transferred lexically.
newChild	p	FS	Creates a new feature structure and attaches it to the head. The attribute name of the new feature structure is given by the attribute *gfunc*. The attribute *relorder* specifies the relative position of the new feature structure.
noChildren	*	list of att. names	Succeeds if the head does not contain any of the listed attributes.
removeChild	p	1	Removes the first child identified by the directive *hasChildren*.
rewriteChild	*	FS	Rewrites attributes in the feature structure of the child. Non-existent attributes will be added.
rewriteHead	*	FS	Rewrites attributes in the feature structure of the head. Non-existent attributes will be added.

Table 8.1: Transfer directives

The following rule removes an auxiliary from an *l*-participle in the third person which may be required, for example, for the translation from Slovenian to Czech.

```
(
preproc
(head= ((type word) (pos verb) (vform lpart) (person 3)))
(hasChildren (aux))
(removeChild 1)
)
```

The following rule rewrites the features gender, case and number of an adjective which is being detached by values of these features from the governing noun to preserve agreement between an adjectival attribute and a noun.

```
(
decomp
(recursive 1)
(head= ((type word) (pos n)))
(child= ((type word) (pos a)))
(copydown (gender case number))
)
```

An example of this rule's use would be the translation of the Czech phrase *velký strom* "big tree" into Macedonian (*големо дрво*) where the gender has changed from masculine to neuter. Without this transfer rule, we would get **голем дрво*.

The following rule changes the infinitive to an *l*-participle in periphrastic future tense constructions as required, for example, when translating from Czech to Slovenian or Polish.

```
(
decomp
(head= ((type word) (pos verb) (vform inf)))
(child= ((type word) (lemma bý t) (vform fin) (tense fut)
(gender $gender) (number $number)))
(rewriteHead ((vform lpart) (gender $gender) (number $number)))
)
```

A similar rule operating on VPs would be used, for example, when translating the Czech VP *napsal jsem* "I wrote/I have written" to Macedonian (*напишав/имам напишано*) since a word-for-word translation would give *напишал сум* which would be well-formed with different word order (*сум напишал*) but still semantically different (in Macedonian, it represents the renarrative).

8.2.2 Translation of Multiword Expressions

It is obvious that some words of the source language are translated as multiword expressions in the target language and vice versa, for example:

(8.2) *babička* "grandmother" (Cze) → *stará mama* (Slv)
zahradní jahoda "garden strawberry" (Cze) → *truskawka* (Pol)

Since these cases require removing or adding of a subordinated feature structure (for the adjective) which is equivalent to removing or adding a node from/to the syntactic tree, such cases are handled by special rules in the structural transfer.

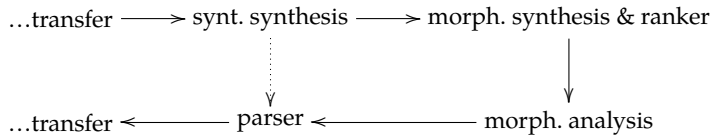
8.3 Chaining MT Systems

Machine translation is a very complicated task in itself and developing an MT system for a language pair is very expensive in terms of time and manpower. Furthermore, statistical MT needs huge bilingual corpora which are mostly not available for language pairs where at least one language is a small one, for example, Welsh or Macedonian. Therefore, attempts were made to find MT methods that would cope with these problems. One possible approach is to exploit an intermediary (natural) language and couple two MT systems together. In Chapter 4, Section 4.2, we have described an MT system from Norwegian into English which uses Danish as an 'interlingua' (Bick and Nygaard, 2007). Unfortunately, the, more or less, poor output of today's MT systems lets such a solution look naïve unless at least one language pair consists of closely related languages.

We did two experiments with coupled MT systems translating from Czech to Slovak through Slovenian as the intermediary language.[1] The first system simply pipes the output of the Czech-to-Slovenian MT system into the Slovenian-to-Slovak one. The other experiment couples both MT systems at a higher level, omitting morphological synthesis and statistical ranker in the first language pair. As our experiments have shown, the latter approach produces significantly better translation.

The high-level pipeline is presented in the following scheme:

(8.3)

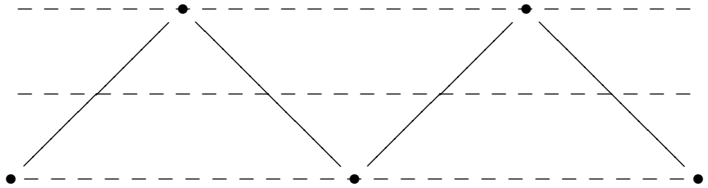


The dotted arrow denotes the 'shortcut' which has been taken in the system architecture to omit morphological analysis and ranker in the first language pair.

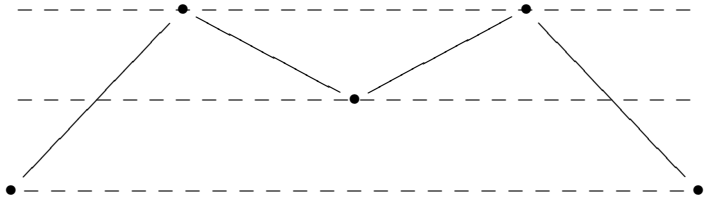
The two approaches could be schematically expressed by the de Vauquois' triangle. The scheme in (8.4) describes the simple pipeline whereas the scheme in (8.5) describes the high-level pipeline (with the 'shortcut').

[1] The work described in this section has been carried out together with Jernej Vičič from the University of Primorska, Koper, Slovenia.

(8.4)



(8.5)



We see that in the high-level pipeline, the process does not return to the lowest level of linguistic representation (linearized sequence of word forms) but remains at a middle stage, in our case—informally expressed—between morphology and syntax.

8.3.1 Discussion

Although machine translation which uses a natural language as a pivot language is typically not expected to produce good translation because it is obvious that a simple pipe of two MT systems multiplies the error rate, it is sometimes inevitable. For example, the system *Webforditas*[2], developed by Morphologic Kft., uses English as an interlingua to translate to/from Hungarian automatically (László Tihanyi, personal communication).

The evaluation of our experiments with MT from Czech to Slovak through Slovenian clearly shows that we get better results if we couple the two MT systems at a higher level. The main point is that the statistical ranker is not used in the first MT system, postponing the selection of one hypothesis to a later stage. At the first sight, this strategy may seem to cause huge ambiguity in the translation process. However, if we do not use morphological synthesis in the first MT system, we do not need morphological analysis in the second system either. Thus it is possible to avoid the morphological ambiguity of the input in the second MT system (which is extremely important for languages with rich inflection, such as Slovenian). In other words, the parser in the second MT system deals with ambiguity of a different type, namely a structural and semantic one which resulted from the first system and could not be resolved before ranking.

The comparison of both systems (the translations of the same input text) has brought an interesting observation: The MT system with the more sophisticated coupling

[2]http://www.webforditas.hu

	BLEU	NIST
simple pipe	0.1690	3.5916
high-level pipe	0.2303	4.1737

Table 8.2: Experimental results of chained MT systems

works faster, most probably due to the fact that morphological ambiguity of the intermediary representation (which is the input of the MT for the second language pair) is widely reduced. The evaluation of results in terms of BLEU (Papineni et al., 2001) and NIST (Doddington, 2002) are presented in Table 8.2.

9
Statistical Ranking and Evaluation

An essential part of the whole MT system is the statistical postprocessor. The main problem with the simple MT process described in the previous sections is that all modules (morphological analyzer, parser and transfer) introduce a high number of ambiguities into the translation. It would be very complicated (if possible at all) to resolve this kind of ambiguity by hand-written rules. That is why we have implemented a stochastichal post-processor which aims to select one particular sentence that best suits the given context.

9.1 Ranking

We use a simple language model based on trigrams (trained on word forms without any morphological annotation) which is intended to sort out "wrong" target sentences (these include grammatically ill-formed sentences as well as sentences with inappropriate lexical mapping). For example, the language model for Slovak has been trained on a corpus of approximately 20 million words which have been randomly chosen from the Slovak Wikipedia[1].

Let us present an example of how this component of the system works. In the source text, we had the following Czech segment (matrix sentence):

(9.1) *Společnost* *ve* *zprávě* *uvedla*
company-FEM,SG,NOM in report-FEM,SG,LOC inform-LPART,FEM,SG

"The company informed in the report…"

The rule-based part of the system is supposed to generate (if there were no rules for VPs) four target segments that collapse to the following two after morphological synthesis:

1. *Spoločnosť vo správe uviedli*,
2. *Spoločnosť vo správe uviedla*.

The reason for the ambiguity is that the Czech word *uvedla* is ambiguous (NEUT,PL and FEM,SG). According to the language model, the ranker is supposed to choose the second sentence as the most probable result.

There are also many homonymic word forms that result in different lemmas in the target languages. For example, the word *pak* means both "then" and "fool-pl.gen", the word *tři* means "three" and the imperative of "to scrub", *ženu* means "wife-sg.acc"

[1] http://sk.wikipedia.org

and "(I'm) hurrying out" etc. The ranker is supposed to sort out the contextually wrong meaning in all these cases if it has not been resolved earlier by the parser.

Let us briefly define the trigram language model formally (a detailed description can be found in (Jelinek, 1997)). For a given word sequence $W = \{w_1, \ldots, w_n\}$ of n words, we define its probability as:

$$p(W) = p(w_1, \ldots, w_n) = \prod_{i=1}^{n} p(w_i | w_0, \ldots, w_{i-1}) \tag{9.2}$$

where w_0 is chosen appropriately to handle the initial condition.

As it is computationally not viable to work with unlimited history, we use a mapping ϕ that approximates the history (in our case by trigrams):

$$p(W) \approx \prod_{i=1}^{n} p(w_i | w_{i-2}, w_{i-1}) \tag{9.3}$$

To estimate the trigram probabilities, we use a large training corpus:

$$f(w_3 | w_1, w_2) = \frac{c_{123}}{c_{12}} \tag{9.4}$$

where c_{123} is the number of times the sequence of words (w_1, w_2, w_3) is observed and, analogically, c_{12} is the number of times the sequence (w_1, w_2) is observed.

Due to the well-known problem of sparse data, we have to use smoothing. A common smoothing method is the linear interpolation of trigram, bigram and unigram frequencies and a uniform distribution on the vocabulary (λ_i are non-negative and sum up to 1):

$$p(w_3 | w_1, w_2) = \lambda_3 f_3(w_3 | w_1, w_2) + \lambda_2 f_2(w_3 | w_2) + \lambda_1 f_1(w_3) + \lambda_0 \frac{1}{V} \tag{9.5}$$

The values of the parameters λ_i are obtained using heldout data.

9.2 Evaluation

We have evaluated the system on 1,000 sentences for the language pairs Czech-Slovak and Czech-Russian against a reference translation and on 100 sentences for the language pairs Czech-Slovak and Czech-Macedonian[2] using a post-edited translation.[3]

[2]We have used the JRC corpus (Steinberger et al., 2006), the UMC corpus (Klyueva and Bojar, 2008) and the Multext-East corpus (Erjavec, 2004). For Czech-Slovak, we have used the dictionaries from the original system *Česílko*. For Czech-Russian, we have used the dictionary created by Ondřej Bojar and Natalia Kljueva. For Czech-Macedonian, a small dictionary was created from scratch for the experiments. We have selected three representative Slavic target languages with different stages of proximity to Czech.

[3]Although we did some practical experiments with Baltic languages, namely Lithuanian (Homola and Rimkutė, 2004), we did not include the language pair Czech-Lithuanian into the final evaluation because

The metrics we are using are Levenshtein edit distances between the automatic translation and a reference translation based on characters and words[4] as well as BLEU (Papineni et al., 2001) and NIST (Doddington, 2002). If we use an edit-distance based metric against post-edited translation, there are three basic possibilities of the outcome of translation of a segment.

1. The rule-based part of the system has generated a 'perfect'[5] translation (among other hypotheses) and the ranker has chosen this one.
2. The rule-based part of the system has generated a 'perfect' translation but the ranker has chosen another one.
3. All translations generated by the rule-based part of the system need post-processing.

In the first case, the edit distance is zero, resulting in accuracy equal to 1. In the second and third case, the accuracy is $1 - d$ with d meaning the edit distance between the segment chosen by the ranker and the post-edited translation divided by the length of the segment.

Once we have the accuracies for all sentences, we use the arithmetic mean as the translation accuracy of the whole text. The accuracy is negatively influenced by several aspects. If a word is not known to the morphological analyzer, it does not get any morphological information, which means that it is practically unusable in the parser. Another possible problem is that a lemma is not found in the dictionary. In such a case, the original source form appears in the translation, which penalizes the score, of course. Finally, sometimes the morphological synthesis component is not able to generate the proper word form in the target language. In such a case, the Slovak lemma appears in the translation.

The results are summarized in Tables 9.1–9.4. The column *original* contains evaluation results for the original architecture as described in (Hajič et al., 2000). The column *shallow* contains results for the improved architecture with a parser for NPs and PPs. The column *deep* contains results for the improved architecture with all parsing rules.

9.2.1 Discussion

In statistical machine translation, it is usual to evaluate test data using an independent reference translation (or more translations). We have done this for Czech-Slovak and Czech-Russian to provide results comparable to other MT systems (although, as has been stated by Callison-Burch et al. (2006), BLEU and similar metrics are believed

of the expiration of the license for Lithuanian morphology. For this language pair, a module for structural transfer has been used in an early version of our framework. In the final evaluation, we have used structural transfer for the pair of (typologically) distant languages Czech and Macedonian. There is practically no need for structural transfer in the case of Czech-to-Slovak MT. As for the Czech-Russian language pair, the transfer did not help at all, probably because of the lower quality of the dictionary which has been generated automatically.

[4] This metric corresponds to the well-known word error rate (WER).

[5] By 'perfect' we mean that the result does not need any human post-processing.

	original	shallow	deep
Czech-Slovak (WER)	52.68%	54.82%	54.22%
Czech-Slovak (character based)	64.92%	65.20%	64.93%
Czech-Russian (WER)	16.06%	18.26%	18.18%
Czech-Russian(character based)	32.94%	36.21%	36.16%

Table 9.1: Evaluation of Slavic language pairs (edit distance) using reference translation

	original	shallow	deep
Czech-Slovak (BLEU)	0.2161	0.2095	0.2082
Czech-Slovak (NIST)	5.8950	5.7490	5.7714
Czech-Russian (BLEU)	0.0512	0.0683	0.0690
Czech-Russian (NIST)	3.0508	3.4201	3.4455

Table 9.2: Evaluation of Slavic language pairs (BLEU and NIST) using reference translation

	original	shallow	deep
Czech-Slovak (WER)	88.96%	88.76%	87.68%
Czech-Slovak (character based)	96.32%	96.90%	96.62%
Czech-Slovak (BLEU)	0.7235	0.7349	0.7128
Czech-Slovak (NIST)	6.9444	7.1121	6.9971
Czech-Macedonian (WER)	54.59%	68.16%	70.94%
Czech-Macedonian (character based)	75.12%	83.77%	86.29%
Czech-Macedonian (BLEU)	0.3383	0.4161	0.5195
Czech-Macedonian (NIST)	4.3760	5.0766	5.4034

Table 9.3: Evaluation of Slavic language pairs using post-edited translation

	Apertium	our system
WER	87.1%	88.2%
character based	91.1%	92.4%

Table 9.4: Portuguese-to-Spanish evaluation (edit distance)

to penalize rule-based MT systems). In rule-based systems for related languages, on the other hand, evaluation metrics based on edit distance are often used, e.g., by Armentano-Oller et al. (2006) in the system *Apertium*. A significant flaw of the evaluation based on post-edited translations it the high human effort, that is why we have evaluated less sentences than with independent reference translations.

The results of the evaluation show that except for very closely related languages (Czech and Slovak), the improved architecture with the ranker produces a better translation than the original architecture proposed in (Hajič et al., 2000). As expected, there is no desperate need for deep syntactic analysis in case of language pairs of closely related languages. The experiments with the language pair Czech-Macedonian (distant languages except for the lexical level) show that the 'shallow' approach could be suitable even for this kind of language pairs[6] (although the use of the parser has its limits because of the lack of valence in the system).

To further support the hypothesis that the improved architecture is generally better than the tagger-based approach, we have modified the system *Apertium* (see Chapter 4, Section 4.5) (we have removed the tagger and added a ranker) for the language pair Portuguese-Spanish (Homola and Kuboň, 2008). The results (measured on 100 post-edited sentences) are presented in Table 9.4.[7]

[6]Nevertheless, a preliminary experiment with Czech-German has shown that shallow MT for this language pair is a dead end, hence here may be the limit of the usability of shallow NLP.

[7]We are very indebted to Sergio Duarte for his kind help with the evaluation.

10
Concluding Discussion

The main topic of this thesis is the contribution of syntactic analysis, especially partial syntactic analysis, to the machine translation between more or less related languages. We focused on the Balto-Slavic language family and presented the implementation of a predominantly rule-based MT system with shallow NLP. We have also validated our framework on a language pair from another language family, namely Romance. As for (typologically) distant languages, the shallow approach seems to be viable for Czech-to-Macedonian MT at most.

In this concluding chapter, we provide a broader discussion about the problematics of partial (shallow) NLP and the use of hybrid (rule-based and statistical) methods in the area of MT. Finally, we summarize the contribution of the thesis.

10.1 Shallow NLP and the Role of Statistics in MT

There are three main branches of MT: rule-based MT, statistical MT and example-based MT. There were also many attempts of combining these approaches to build a hybrid system. Like the original shallow-transfer MT system *Česílko*, our framework is predominantly rule-based, with one supporting statistical module. The following subsections summarize some findings from the development of our MT framework.

10.1.1 Dealing with Extensive Morphological Ambiguity

It is a well-known fact that Baltic and Slavic languages have a very rich morphology and an extremely free word order. This fact imposes a difficulty on the NLP of these languages as it is necessary to deal with a much higher ambiguity as compared, for example, with most Germanic or Romance languages.

The transfer-less approach suggested by Hajič et al. (2000) uses a statistical tagger as its first module to disambiguate the input at the beginning of the translation process. Although the accuracy of the tagger was comparatively high (96%), it has still proved to be insufficient for the given task in general.

On the other hand, full-fledged rule-based MT systems use a full parser to deal with the morphological ambiguity and, if the result of the parser is still ambiguous, then it means that the processed sentence is ambiguous structurally. This approach has the disadvantage that it is practically impossible to create a hand-written grammar that would be capable of processing general texts.

We have investigated a middle way between the two approaches. In our framework, there is no morphological disambiguation but the parser is only partial. In practice, this simplified approach does not have the flaws of the statistical tagger but it does not resolve the ambiguity of the processed sentence completely, thus subsequent modules work with partially ambiguous data and the final disambiguation is done at the end of the translation process by a ranker which is based on a simple trigram language model for the target language.

Theoretically, one could use this procedure without any parser and rely on the final language model. However, the parser not only restricts the ambiguity but it also adds important information for the transfer, and if there was the full morphological ambiguity, the translation would need a huge amount of time and resources (exponential with regard to the length of the processed sentence). Moreover, rule-based disambiguation is generally more reliable than statistical methods; so we simply follow the premise "don't guess if you know". Hence our method is a compromise between the two mentioned approaches.

10.1.2 On the Lexical and Structural Non-Determinism in MT

In the translation process, there are many sources of ambiguity. We have already mentioned the morphological ambiguity which is very important especially for languages with rich inflection, such as Baltic and Slavic languages. The other notable types of ambiguity are lexical and structural (syntactic) ambiguities.

The lexical ambiguity comprises the fact that a word in the source language may be translated differently into the target language depending on the context, style, etc. In our framework, where all modules are capable of dealing with potentially ambiguous input, this problem can be partially solved for free by letting the lexical transfer generate all possible translations and relying on the final ranker. In other words, if we are not able to provide a rule to solve a particular ambiguity, we let the ranker guess.

The same applies for the structural ambiguity. Nonetheless in the syntax, we can exploit frequent free-rides. For example, a well-known and hard to solve problem is the syntactic ambiguity of prepositional phrases which can often depend on a noun or on a verb. The decision is mostly of semantic nature and cannot be made within the parser, not to say within a shallow parser, so the parser causes a structural ambiguity in such a case. On the other hand, in many cases the ambiguity is resolved 'for free' in the phase of syntactic synthesis, as the target language would often express the prepositional phrase with the same syntactic ambiguity.

10.1.3 The Interplay between Rule-Based and Statistical Modules

Our experiments indicate that, in general, it is probably better to postpone statistical processing as far as possible in the translation process. In our framework, the only statistical module is the ranker at the end of the system and we achieve same or better results than the original architecture with the statistical tagger at the beginning.

Our claim is also supported by the experiment of coupling two MT systems to obtain a new translation pair. This practice is not very common for obvious reasons but it may be useful for closely related languages, as described by Bick and Nygaard (2007). We have coupled two MT systems as a simple pipe, i.e., with linearized sentences as the intermediary result, and using a set of translation hypotheses at a higher linguistic level as the intermediary result. The translation quality was significantly better in the latter case, hence we have another example when postponing the disambiguation leads to better results. In this experiment, not only the quality was better but the system also worked faster, as we could widely omit the morphological ambiguity in the second MT system.

10.2 Contribution of the Thesis

In the introductory Chapters 2 and 3, we described the notations used in the thesis and briefly presented the family of Baltic and Slavic languages. Chapter 4 reviewed the most notable MT systems that were designed for related languages. The system *Česílko* was of especially great importance since it has been designed for Slavic languages and we have re-used some modules of this system. In Chapter 5, we gave a broader perspective on the various free-rides and major differences among the researched language family.

The main part of the thesis, namely Chapters 6–8, focused on the syntax of Baltic and Slavic languages, on a concrete implementation of our MT framework and on parsing and transfer rules for the MT between Baltic and Slavic languages. In Chapter 9, the system was used to translate a set of sentences of several language pairs and the results were evaluated using a couple of automatic MT metrics. The results indicated that our framework is not worse (and often better) than the architecture of the original system *Česílko*, and that it also outperforms *Česílko's* direct successor *Apertium*, which uses the same shallow-transfer approach but focuses on typologically different Romance languages.

The thesis contributes to the art of partial syntactic analysis and MT for related languages by the following:

- re-evaluating the role of the tagger in rule-based MT for related languages,
- designing a partial grammar for languages with rich inflection with a twofold purpose: to overcome morphosyntactic differences in local constituents and to restrict the huge morphological ambiguity which is symptomatic for these languages,
- formalizing the functions and interrelationships of lexical, morphological and syntactic transfer,
- suggesting and evaluating a novel method of coupling two MT systems to obtain a new translation pair with better translation quality as compared to a simple pipe of two MT systems,

Furthermore, we have designed and implemented the following modules.

- unification-based chart parser,
- module for non-deterministic lexical, morphological and syntactic transfer,

As in most research projects, there remain many open questions. In many cases, this thesis only foreshadows a solution.

For example, further research is needed to localize the level of similarity of two languages where statistical MT gives better results than shallow approaches or where the development of a rule-based MT system would be too costly.

It also remains to be examined how would non-projective parsing improve our system since we are parsing only projective syntactic structures which may be a problem for Baltic and Slavic languages.

Major improvements of the system in its current state could be probably achieved by a more sophisticated implementation of the ranker and by extending the parser and lexicon by valency information.

A
Czech Parser Rules

This section lists the syntactic rules which we used in our system to parse Czech input. The grammar is based on observations presented in Chapter 6, but of course only some of the morphosyntactic phenomena are handled by the rules; the set of rules was composed empirically during the experiments. In the source file of the system, we use *s*-expressions[1] for rule declaration since this format is simple to parse and it is still easily readable by humans. When adding a rule, one may start with designing the rule in the LFG notation (Kaplan and Bresnan, 1982). Example A.1 declares a simple rule for an NP which governs a PP (the agreement in case is expressed by the first equation):

(A.1) PP $\rightarrow$ P **NP**, $\uparrow$ CASE = $\downarrow$ CASE & $\uparrow$ PREP = $\downarrow$

The rule attaches a preposition to an NP. The first part (before the comma) declares the categories of the subchain the rule will be tentatively applied to. The bold font denotes that the feature structure of the right element will be propagated as the head (the core of the feature structure) of the phrase. It takes a preposition and an NP to the right of it that agree in case, which is declared in the other part of the rule—the conditions. Thus the resulting feature structure is the feature structure of the NP extended with a new attribute—*prep*—that is unified with the feature structure of the preposition.

Once converted to the notation of our formalism, the rule can be written as follows:[2]

(A.2) $$\begin{bmatrix}\text{POS} & \text{prep}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{HAS\_PREP} & 1\\ \text{PREP} & \$1\end{bmatrix}$$

Finally, the form of the rule in the source code of our grammar is as follows (with *s*-expressions denoting attribute-value pairs):

```
(
  ( ((type word) (pos prep) (case $case))
    ((type shackle))   ((type word) (pos n) (case $case)) )
  ( $3 ((prep $1) (has_prep 1)) )
)
```

[1] Lists in round brackets in a Lisp-like notation.

[2] The dollar sign ($) followed by an alphanumeric identifier denotes a variable. The dollar sign ($) followed by a number can occur only on the right-hand side of the rule and refers to a feature structure on the left-hand side of the rule excluding the 'shackle' structures ($1 refers to the first feature structure etc.)

A.1 Shallow Rules

(A.3) $\begin{bmatrix}\text{POS} & \text{prep}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{HAS\_PREP} & 1\\ \text{PREP} & \$1\end{bmatrix}$

(A.4) $\begin{bmatrix}\text{POS} & \text{prep}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{PRONTYPE} & \text{pers}\\ \text{CASE} & \$\text{case}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{HAS\_PREP} & 1\\ \text{PREP} & \$1\end{bmatrix}$

(A.5) $\begin{bmatrix}\text{POS} & \text{prep}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{PRONTYPE} & \text{indef}\\ \text{CASE} & \$\text{case}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{HAS\_PREP} & 1\\ \text{PREP} & \$1\end{bmatrix}$

(A.6) $\begin{bmatrix}\text{POS} & \text{prep}\\ \text{CASE} & \$\text{case}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{PRONTYPE} & \text{dem}\\ \text{CASE} & \$\text{case}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{HAS\_PREP} & 1\\ \text{PREP} & \$1\end{bmatrix}$

(A.7) $\begin{bmatrix}\text{POS} & \text{adv}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{a}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{PREP} & \$1\end{bmatrix}$

(A.8) $\begin{bmatrix}\text{POS} & \text{num}\\ \text{NUMTYPE} & \text{indef}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \text{gen}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{NUM} & \$1\end{bmatrix}$

(A.9) $\begin{bmatrix}\text{POS} & \text{pron}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{CASE} & \$\text{case}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{DEF} & \text{def}\\ \text{CASE} & \$\text{case}\end{bmatrix} \rightarrow \$2 \wedge$

$\begin{bmatrix}\text{DET} & \$1\end{bmatrix}$

(A.10) $\begin{bmatrix}\text{POS} & \text{a}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{CASE} & \$\text{case}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{CASE} & \$\text{case}\end{bmatrix} \rightarrow \$2 \wedge$

$\begin{bmatrix}+\text{ATT} & \$1\end{bmatrix}$

(A.11) $\begin{bmatrix}\text{POS} & \text{n}\\ \text{0ATT} & \text{0}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{PREP} & \text{nil}\\ \text{TYPE} & \text{word}\\ \text{ATT-GEN} & \text{nil}\\ \text{CASE} & \text{gen}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{ATT-GEN} & \$3\end{bmatrix}$

(A.12) $\begin{bmatrix}\text{POS} & \text{n}\\ \text{TYPE} & \text{word}\\ \text{ATT-GEN} & \text{nil}\\ \text{0ATT} & \text{0}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{PREP} & \text{nil}\\ \text{TYPE} & \text{word}\\ \text{ATT-DAT} & \text{nil}\\ \text{CASE} & \text{dat}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{ATT-DAT} & \$3\end{bmatrix}$

(A.13) $\begin{bmatrix}\text{POS} & \text{n}\\ \text{TYPE} & \text{word}\\ \text{ATT-GEN} & \text{nil}\\ \text{0ATT} & \text{0}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{ATT-DAT} & \text{nil}\\ \text{POS} & \text{a}\\ \text{PREP} & \text{nil}\\ \text{TYPE} & \text{word}\\ \text{DEF} & \text{def}\\ \text{CASE} & \text{dat}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{ATT-DAT} & \$3\end{bmatrix}$

(A.14) $\begin{bmatrix}\text{POS} & \text{n}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{!HAS\_PREP} & \text{1}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{+ADJ} & \$3\end{bmatrix}$

A.2 Deep rules

(A.15) $\begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{PRONTYPE} & \text{pers}\\ \text{CASE} & \text{dat}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{IOBJ} & \$1\end{bmatrix}$

(A.16) $\begin{bmatrix}\text{POS} & \text{conj}\\ \text{LEMMA} & \text{aby}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{CONJ} & \$1\end{bmatrix}$

(A.17) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{inf}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \text{dat}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{IOBJ} & \$3\end{bmatrix}$

(A.18) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \text{acc}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{OBJ} & \$3\end{bmatrix}$

(A.19) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{CASE} & \text{acc}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{OBJ} & \$3\end{bmatrix}$

(A.20) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{inf}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{a}\\ \text{TYPE} & \text{word}\\ \text{DEF} & \text{def}\\ \text{CASE} & \text{dat}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{IOBJ} & \$3\end{bmatrix}$

(A.21) $\begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{FORM} & \text{se}\\ \text{PRONTYPE} & \text{refl}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{REFL} & 1\end{bmatrix}$

(A.22) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{FORM} & \text{se}\\ \text{PRONTYPE} & \text{refl}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{REFL} & 1\end{bmatrix}$

(A.23) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{!HAS\_PREP} & 1\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{+ADJ} & \$3\end{bmatrix}$

(A.24) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{fin}\\ \text{NUMBER} & \$\text{number}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{part\_short}\\ \text{NUMBER} & \$\text{number}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADJ} & \$1\end{bmatrix}$

(A.25) $\begin{bmatrix}\text{FORM} & \text{ale}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADJ-XBUT} & \$1\end{bmatrix}$

(A.26) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{FORM} & \text{se}\\ \text{PRONTYPE} & \text{refl}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{REFL} & 1\end{bmatrix}$

(A.27) $\begin{bmatrix}\text{POS} & \text{pron}\\ \text{TYPE} & \text{word}\\ \text{FORM} & \text{se}\\ \text{PRONTYPE} & \text{refl}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{REFL} & 1\end{bmatrix}$

(A.28) $\begin{bmatrix}\text{POS} & \text{adv}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADV-ADV-L} & \$1\end{bmatrix}$

(A.29) $\begin{bmatrix}\text{POS} & \text{adv}\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{fin}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADV-ADV-L} & \$1\end{bmatrix}$

(A.30) $\begin{bmatrix}\text{POS} & \text{n2}\\ \text{HAS\_PREP} & 1\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADV-PP-L} & \$1\end{bmatrix}$

(A.31) $\begin{bmatrix}\text{POS} & \text{pron}\\ \text{!HAS\_PREP} & 1\\ \text{TYPE} & \text{word}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{VFORM} & \text{lpart}\\ \text{TYPE} & \text{word}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{ADV-PP-L} & \$1\end{bmatrix}$

(A.32) $\begin{bmatrix}\text{POS} & \text{n}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{CASE} & \text{nom}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{PERSON} & 3\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{lpart}\\ \text{NUMBER} & \$\text{number}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{SUBJ} & \$1\end{bmatrix}$

(A.33) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{PERSON} & 3\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{lpart}\\ \text{NUMBER} & \$\text{number}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{n}\\ \text{GENDER} & \$\text{gender}\\ \text{TYPE} & \text{word}\\ \text{NUMBER} & \$\text{number}\\ \text{CASE} & \text{nom}\end{bmatrix} \rightarrow \$1 \wedge \begin{bmatrix}\text{SUBJ} & \$3\end{bmatrix}$

(A.34) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{LEMMA} & \text{být}\\ \text{PERSON} & \$\text{person}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{fin}\\ \text{TENSE} & \text{fut}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{PERSON} & \$\text{person}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{inf}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{AUX} & \$1\end{bmatrix}$

(A.35) $\begin{bmatrix}\text{POS} & \text{verb}\\ \text{LEMMA} & \text{být}\\ \text{PERSON} & \$\text{person}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{fin}\\ \text{TENSE} & \text{pres}\end{bmatrix} + \begin{bmatrix}\text{TYPE} & \text{shackle}\end{bmatrix} + \begin{bmatrix}\text{POS} & \text{verb}\\ \text{PERSON} & \$\text{person}\\ \text{TYPE} & \text{word}\\ \text{VFORM} & \text{lpart}\end{bmatrix} \rightarrow \$2 \wedge \begin{bmatrix}\text{AUX} & \$1\end{bmatrix}$

(A.36) $\begin{bmatrix} \text{POS} & \text{verb} \\ \text{LEMMA} & \text{být} \\ \text{PERSON} & \text{\$person} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{fin} \\ \text{TENSE} & \text{pres} \end{bmatrix} + \begin{bmatrix} \text{TYPE} & \text{shackle} \end{bmatrix} + \begin{bmatrix} \text{POS} & \text{verb} \\ \text{PERSON} & \text{\$person} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{part\_short} \end{bmatrix} \rightarrow \text{\$2} \wedge$

$\begin{bmatrix} \text{AUX} & \text{\$1} \end{bmatrix}$

(A.37) $\begin{bmatrix} \text{POS} & \text{verb} \\ \text{LEMMA} & \text{být} \\ \text{PERSON} & \text{\$person} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{lpart} \\ \text{TENSE} & \text{pres} \end{bmatrix} + \begin{bmatrix} \text{TYPE} & \text{shackle} \end{bmatrix} + \begin{bmatrix} \text{POS} & \text{verb} \\ \text{PERSON} & \text{\$person} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{part\_short} \end{bmatrix} \rightarrow \text{\$2} \wedge$

$\begin{bmatrix} \text{AUX} & \text{\$1} \end{bmatrix}$

(A.38) $\begin{bmatrix} \text{POS} & \text{verb} \\ \text{LEMMA} & \text{být} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{fin} \\ \text{NUMBER} & \text{\$number} \end{bmatrix} + \begin{bmatrix} \text{TYPE} & \text{shackle} \end{bmatrix} + \begin{bmatrix} \text{POS} & \text{n} \\ \text{PERSON} & 3 \\ \text{TYPE} & \text{word} \\ \text{NUMBER} & \text{\$number} \\ \text{CASE} & \text{nom} \end{bmatrix} \rightarrow \text{\$1} \wedge$

$\begin{bmatrix} \text{SUBJ} & \text{\$3} \end{bmatrix}$

(A.39) $\begin{bmatrix} \text{POS} & \text{n} \\ \text{PERSON} & 3 \\ \text{TYPE} & \text{word} \\ \text{NUMBER} & \text{\$number} \\ \text{CASE} & \text{nom} \end{bmatrix} + \begin{bmatrix} \text{TYPE} & \text{shackle} \end{bmatrix} + \begin{bmatrix} \text{POS} & \text{verb} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{fin} \\ \text{NUMBER} & \text{\$number} \end{bmatrix} \rightarrow \text{\$2} \wedge$

$\begin{bmatrix} \text{SUBJ} & \text{\$1} \end{bmatrix}$

(A.40) $\begin{bmatrix} \text{POS} & \text{verb} \\ \text{LEMMA} & \text{být} \\ \text{TYPE} & \text{word} \\ \text{VFORM} & \text{fin} \end{bmatrix} + \begin{bmatrix} \text{TYPE} & \text{shackle} \end{bmatrix} + \begin{bmatrix} \text{POS} & \text{n} \\ \text{PERSON} & 3 \\ \text{TYPE} & \text{word} \\ \text{CASE} & \text{ins} \end{bmatrix} \rightarrow \text{\$1} \wedge$

$\begin{bmatrix} \text{NOM-PRED-N} & \text{\$3} \end{bmatrix}$

Summary

This thesis explores the contribution of syntactic analysis to the machine translation (MT) between related languages and it also attempts to explore the limits of shallow MT methods. We focus on one group of languages, the Balto-Slavic language family, and one MT architecture, namely hybrid systems with prevalently rule-based modules.

First, we present related work for Slavic, Scandinavian, Turkic, Celtic and Romance languages. We review different approaches of MT between related languages including the MT system for Slavic languages *Česílko* which constitutes the basis of our system.

Second, we suggest a modification of the commonly used shallow-transfer approach. We describe in detail the implementation of the proposed framework, namely the partial parser, shallow transfer and stochastic ranker, and evaluate the improved architecture on three language pairs using several well-known metrics such as WER, BLEU and NIST.

Third, we examine how our architecture behaves if we couple two MT systems to obtain a new translation pair as compared to a simple pipe of two MT language pairs. This experiment enlightens some aspects of the relationship between deterministic and non-deterministic approaches to morphological analysis, parsing and transfer.

In the concluding chapter, we provide a broader perspective on hybrid methods in MT between related languages and finally, we summarize the contribution of the thesis.

Bibliography

Lars Ahrenberg and Maria Holmqvist. Back to the Future? The Case for English-Swedish Direct Machine Translation. In *Proceedings of Recent Advances in Scandinavian Machine Translation*, Uppsala, Sweden, 2005.

Kemal Altintas and Ilyas Cicekli. A Machine Translation System between a Pair of Closely Related Languages. In *Proceedings of the 17th International Symposium on Computer and Information Sciences (ISCIS 2002)*, pages 192–196, Orlando, Florida, 2002.

Vytautas Ambrazas. *Dabartinės lietuvių kalbos gramatika*. Mokslo ir enciklopedijų leidykla, Vilnius, 1996.

Vytautas Ambrazas, Aleksas Girdenis, Kazys Morkūnas, Algirdas Sabaliauskas, Vincas Urbutis, Adelė Valeckienė, and Aleksandras Vanagas. *Lietuvių kalbos enciklopedija*. Mokslo ir enciklopedijų leidybos institutas, Vilnius, 1999.

Carme Armentano-Oller, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A. Scalco. Open-source Portuguese-Spanish machine translation. In *Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese*, Rio de Janeiro, Brasil, 2006.

Eckhard Bick and Lars Nygaard. Using Danish as a CG Interlingua: A Wide-Coverage Norwegian-English Machine Translation System. In *Proceedings of NODALIDA*, Tartu, Estonia, 2007.

Hadumod Bußmann. *Lexikon der Sprachwissenschaft*. Alfred Kroener Verlag, Stuttgart, 2002.

Alevtina Bémová, Karel Oliva, and Jarmila Panevová. Some Problems of Machine Translation Between Closely Related Languages. In *Proceedings of the 12th conference on Computational linguistics*, volume 1, pages 46–48, Budapest, Hungary, 1988.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, 2006.

Alain Colmerauer. Les systèmes Q ou un formalisme pour analyser et synthétiser des phrases sur ordinateur. Technical report, Mimeo, Montréal, 1969.

Antonio Corbi-Bellot, Mikel Forcada, Sergio Prtiz-Rojas, Juan Antonie Perez/Ortiz, Gema Remirez-Sanchez, Felipe Sanchez Martinez, Inaki Alegria, Aingeru Mayor, and Kepa Sarasola. An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain. In *Proceedings of the 10th Conference of the European Association for Machine Translation*, Budapest, 2005.

Łukasz Dębowski, Jan Hajič, and Vladislav Kuboň. Testing the limits — adding a new language to an MT system. *Prague Bulletin of Mathematical Linguistics*, 78, 2002.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the ARPA Workshop on Human Language Technology*, 2002.

Helge Dyvik. Exploiting Structural Similarities in Machine Translation. *Computers and Humanities*, 28:225–245, 1995.

Tomaž Erjavec. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Paris, 2004.

Hans-Werner Eroms. *Syntax der deutschen Sprache*. Walter de Gruyter, Berlin, 2000.

Stefano Federici, Simonetta Montemagni, and Vito Pirrelli. Shallow Parsing and Text Chunking: A View On Underspecification in Syntax. In *Workshop on Robust Parsing, 8th ESSLLI*, pages 35–44, 1996.

M. Forcada, A. Garrido, R. Canals, A. Iturraspe, S. Montserrat-Buendia, A. Esteve, S. Ortiz Rojas, H. Pastor, and P.M. Pérez. The Spanish-Catalan machine translation system interNOSTRUM. *0922-6567 - Machine Translation*, VIII:73–76, 2001.

Berthold Forssman. *Lettische Grammatik*. Verlag J.H. Roell, Dettelbach, 2001.

Victor Friedman. *Macedonian*. SEELRC, 2001.

L. T. F. Gamut. Logic, language and meaning 2: Intensional logic and logical grammar. *University of Chicago Press, Chicago*, 1991.

Jan Hajič. An MT System Between Closely Related Languages. In *Proceedings of the third conference of the European Chapter of the Association for Computational Linguistics*, pages 113–117, Copenhagen, Denmark, 1987.

Jan Hajič and Vladislav Kuboň. Tagging as a Key to Successful MT. In *Proceedings of the Malý informatický seminář*, Josefův Důl, 2003.

Jan Hajič, Jan Hric, and Vladislav Kuboň. Machine translation of very close languages. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 7–12, Seattle, Washington, USA, 2000.

Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidová-Hladká. Prague depedency treebank 1.0. *CDROM, CAT: LDC2001T10, ISBN 1-58563-212-0*, 2001.

Jan Hajič, Petr Homola, and Vladislav Kuboň. A simple multilingual machine translation system. In *Proceedings of the MT Summit IX*, New Orleans, 2003.

Petr Homola and Vladislav Kuboň. Improving machine translation between closely related Romance languages. In *Proceedings of the EAMT*, Hamburg, 2008.

Petr Homola and Erika Rimkutė. Artimų kalbų mašininis vertimas. *Kalbų studijos*, 6:77–81, 2004.

Pětr Janaš. *Niedersorbische Grammatik*. Domowina-Verlag, Bautzen, 1976.

Frederick Jelinek. *Statistical methods for speech recognition*. Massachusetts Institute of Teechnology, 1997.

Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *Mental Representation of Grammatical Relations*. MIT Press, Cambridge, 1982.

Lauri Kartunnen. D-PATR: A development environment for Unification-based Grammars. In *Proceedings of Coling*, pages 74–80, 1986.

Natalia Klyueva and Ondřej Bojar. UMC 0.1: Czech-Russian-English Multilingual Corpus. In *Proceedings of the Conference "Korpusnaja lingvistika — 2008"*, pages 188–195, Sankt-Peterburg, Russia, 2008. ISBN 978-5-288-04769-5.

Blaže Koneski. *Историја на македонскиот јазик [History of the Macedonian languages]*. Kočo Racin, Skopje, 1965.

Vladislav Kuboň. Problems of robust parsing of Czech. *Ph.D. thesis, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Praha*, 2001.

Jerzy Kuryłowicz. Le problème du classement des cas. *Bulletin de la Société Polonaise de Linguistique IX*, pages 20–43, 1949.

James E. Lavine. Subject properties and ergativity in North Russian and Lithuanian. In Gerbert Coats Katarzyna Dziwirek and Cynthia Vakareliyska, editors, *Formal Approaches to Slavic Linguistics 7*, pages 307–328. Michigan Slavic Publications, 1999.

James E. Lavine. The morphosyntax of Polish and Ukrainian -no/-to. *Journal of Slavic Linguistics*, 13(1):75–117, 2005.

Dmitry Levinson. Aspect in Negative Imperatives and Genitive of Negation: A Unified Analysis of Two Phenomena in Russian. 2005. URL `http://www.stanford.edu/~dmitryle/Levinson2005_ImperfectiveAndGenitiveOfNegation.pdf`.

Willy Mayerthaler, Günther Fliedl, and Christian Winkler. *Lexikon der natürlichkeitstheoretischen Syntax und Morphosyntax*. Stauffenberg Verlag, Tübingen, 1998.

Jarmila Panevová. *Formy a funkce ve stavbě české věty*. Academia, Praha, 1980.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, 2001.

Ernesta Račienė. Zur Frage des Aspekts und der Aktionsarten im Deutschen und Litauischen. *Žmogus ir žodis, Vilniaus pedagoginis universitetas*, 1, 1999.

Kevin P. Scannell. Machine translation for closely related language pairs. In *Proceedings of the Workshop Strategies for developing machine translation for minority languages*, Genoa, Italy, 2006.

Petr Sgall, Eva Hajičová, and Eva Buráňová. *Aktuální členění věty v češtině*. Academia, Praha, 1980.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reider Publishing Company, 1986.

Manfred Starosta. *Niedersorbisch schnell und intensiv 2*. Ludowe nakładnistwo Domowina, Bautzen, 1992.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufis, and Daniel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. Sep 2006. URL `http://arxiv.org/abs/cs.CL/0609058`.

Charles E. Townsend and Laura A. Janda. *Gemeinslavisch und Slavisch im Vergleich. Einführung in die Entwicklung von Phonologie und Flexion*. Verlag Otto Sagner, München, 2003.

Valentin I. Trubinskij. *Очерки русского диалектного синтаксиса*. Izdatěľstvo Leningradskogo universiteta, Leningrad, 1984.

Nikolaos H. Trunte. *Altkirchenslavisch*, volume 1 of *Словенскъи языкъ. Ein praktisches Lehrbuch des Kirchenslavischen in 30 Lektionen. Zugleich eine Einführung in die slavische Philologie*. Verlag Otto Sagner, München, 2005.

Aloyzas Vidugiris. *Zietelos lietuvių šnekta*. Presvika, Vilnius, 2004.

Jernej Vičič. Rapid development of data for shallow transfer RBMT translation systems for highly inflective languages. In *Proceedings of 6th Language Technologies Conference 2008*, 2008.

Rasuolė Vladarskienė. Iliatyvo vartojimas dabartinėje lietuvių kalboje. *Kalbos kultūra*, 76:47–57, 2003.

Eva Žáčková. *Parciální syntaktická analýza (češtiny)*. PhD thesis, Fakulta informatiky Masarykovy univerzity, Brno, 2002.

Harald Weydt and Alicja Kaźmierczak. Gibt es ein Perfekt im modernen Polnisch? *Linguistik online*, 4(3), 1999.

Dan Zeman. Neprojektivity v Pražském závislostním korpusu (PDT). Technical Report 22, Univerzita Karlova, Praha, 2004.

Zigmas Zinkevičius. *Lietuvių kalbos dialektologija*. Mokslo ir enciklopedijų leidybos institutas, Vilnius, 1994.

Zigmas Zinkevičius. *The history of the Lithuanian language*. Mokslo ir enciklopedijų leidybos institutas, Vilnius, 1998.

Index