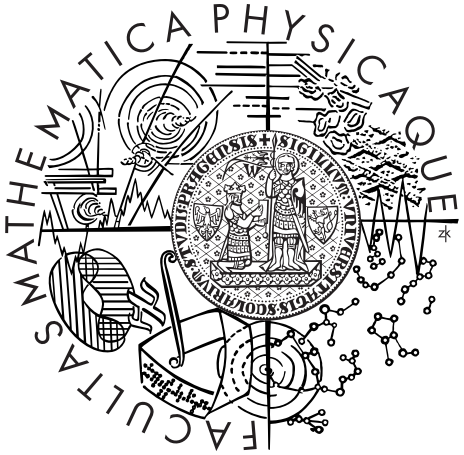
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Rudolf Rosa

Automatic post-editing of phrase-based machine translation outputs

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: RNDr. David Mareček, Ph.D.

Study programme: Informatics

Specialization: Mathematical Linguistics

Prague 2013

I would like to thank all my colleagues at Institute of Formal and Applied Linguistics for providing me with support. I would especially like to thank Mgr. Ondřej Dušek, Mgr. Aleš Tamchyna, Mgr. Martin Popel and doc. Ing. Zdeněk Žabokrtský, Ph.D., who directly helped me with several parts of Depfix. And most of all, I would like to thank my advisor, RNDr. David Mareček, Ph.D..

Another thanks go to Karel Škoch, Tina Resslerová, Alžběta Stříbrná, Zuzana Škardová, Lucie Kadeřábková, and Vendulka Michlíková, the post-editors and annotators who post-edited and evaluated several thousands of sentences for the evaluation of Depfix.

Most of all, I would like to thank my girlfriend for support and patience throughout the countless hours spent on this work.

I dedicate this thesis to them.

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

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

In Prague date 12th April 2013 Rudolf Rosa

Název práce: Automatická post-editace výstupů frázového strojového překladu

Autor: Rudolf Rosa

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. David Mareček, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Představujeme Depfix, systém pro samočinnou post-edititaci výstupů frázových strojových překladů z angličtiny do češtiny, založený na jazykovědných znalostech. Nejprve jsme rozebrali druhy chyb, kterých se dopouští typický strojový překladač. Poté jsme vytvořili sadu pravidel a statistickou komponentu, které opravují takové chyby, které jsou běžné nebo závažné a může přicházet v úvahu jejich oprava pomocí našeho přístupu. Používáme řadu nástrojů pro zpracování přirozeného jazyka, které nám poskytují rozbor vstupních vět. Navíc jsme reimplementovali závislostní analyzátor a několika způsoby jej upravili pro provádění rozboru výstupů statistických strojových překladačů. Provedli jsme automatická i ruční vyhodnocení, která potvrdila, že kvalita překladů se zpracováním v našem systému zlepšuje.

Klíčová slova: automatická post-editace, strojový překlad, závislostní rozbor, Treex

Title: Automatic post-editing of phrase-based machine translation outputs

Author: Rudolf Rosa

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. David Mareček, Ph.D., Institute of Formal and Applied Linguistics

Abstract: We present Depfix, a system for automatic post-editing of phrase-based English-to-Czech machine translation outputs, based on linguistic knowledge. First, we analyzed the types of errors that a typical machine translation system makes. Then, we created a set of rules and a statistical component that correct errors that are common or serious and can have a potential to be corrected by our approach. We use a range of natural language processing tools to provide us with analyses of the input sentences. Moreover, we reimplemented the dependency parser and adapted it in several ways to parsing of statistical machine translation outputs. We performed both automatic and manual evaluations which confirmed that our system improves the quality of the translations.

Keywords: automatic post-editing, machine translation, dependency parsing, Treex

Contents

Chapter 1

Introduction

Statistical machine translation (SMT) has become the state-of-the-art approach to machine translation in many languages. However, it is primarily not linguistically motivated, and its outputs contain many errors that rule-based translation systems would probably not make.

In this thesis, we present Depfix, a complex automatic post-editing system, which uses linguistic knowledge to correct errors in English-to-Czech statistical machine translation outputs. As shown in the blocksheme in Figure 1.1, the input to Depfix is both the target (Czech) sentence produced by an SMT system, and its source (English) sentence. Its output is the corrected Czech sentence.

We focus on correcting errors that are serious and/or frequently appear in SMT outputs. However, we only attempt to fix errors that seem to be caused by a lack of linguistic knowledge incorporated in the SMT system, as there is a high chance of Depfix, being linguistically motivated, to be able to correct such errors. Usually these are grammatical errors, especially in morphological agreement. We employ both a rule-based and a statistical approach to fixing the errors, depending on which seems to be more appropriate for the respective error type.

We implemented the system in Treex (Popel and Žabokrtský, 2010), a natural language processing framework in Perl. The framework supports many languages, but focuses primarily on English and Czech. It is built upon the theory of Functional Generative Description (Sgall, 1967), which was adapted upon creating the Prague Dependency Treebank (Hajič, 1998), and later adapted again for the use in Treex. It defines three layers of language representation: the morphological (word analysis) layer, the analytical (surface syntax) layer, and the tectogrammatical (deep syntax) layer.[1] We follow this division in our work, describing the Depfix system from the perspective of these layers.

To analyze and correct the input sentences, Depfix makes use of a range of tools that are incorporated in Treex, such as taggers and parsers. However, the performance of the tools is not optimal when applied to the outputs of SMT systems, as these are often erroneous, whereas the analysis tools are trained on gold-standard data which are not erroneous. To account for that, we try to improve the performance of the tools by applying rule-based post-processing of their outputs that corrects some of the errors often made by the tools.

[1] In Treex, the morphological layer was merged with the analytical layer, but in our work we use the original three-layer division of the Prague Dependency Treebank.

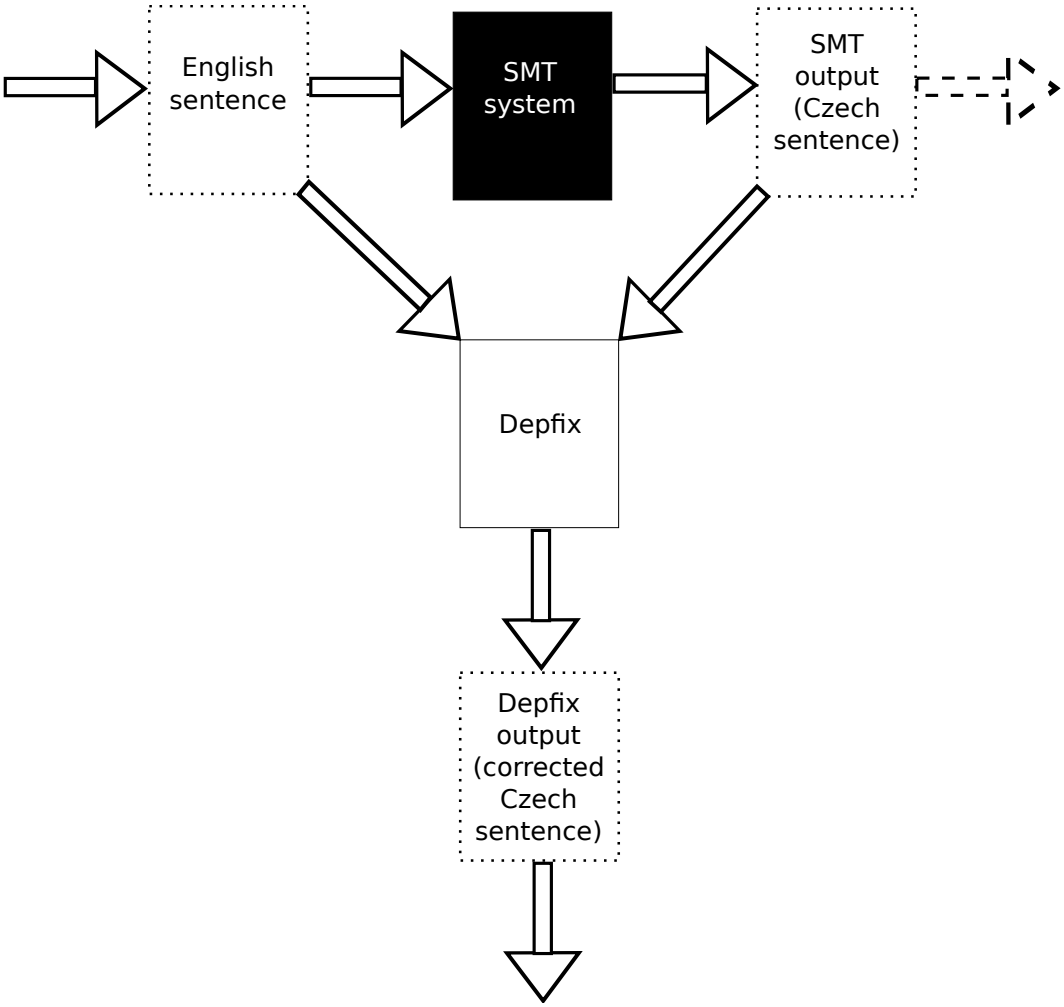


Figure 1.1: The block scheme of Depfix operation.

One of the tools that Depfix heavily relies on is a dependency parser. To get a high-performance parser for Czech SMT outputs, we reimplemented the Maximum Spanning Tree Parser (McDonald et al., 2005) and adapted it in many ways for that specific task by adding several types of new features and modifying its training data.

We evaluated the performance of the Depfix system both automatically and manually on outputs of 13 English-to-Czech machine translation systems that participated in the WMT translation task (Callison-Burch et al., 2010, 2011, 2012). The evaluations show that Depfix is able to correct many errors in the outputs of most today's SMT systems, leading to an improvement of translation quality, which is around 0.4 BLEU points on average.

1.1 A Brief Example

To introduce Depfix quickly, let us show how it works when processing one short sentence. This example also allows us to introduce some basic terminology which we then use throughout the thesis.

Please note that the example does not contain any detailed descriptions – these constitute the major part of this thesis.

1.1.1 Input

The input for Depfix is a source English sentence, and its Czech translation produced by an SMT system (the target sentence), as shown in Example 1.1. The sentences, as most of the examples in the thesis, are adapted from our development data: the WMT_{10} dataset (Callison-Burch et al., 2010) translated by an English-to-Czech version of the Moses phrase-based machine translation system (Koehn et al., 2007), which was adapted by Bojar et al. (2012a). We denote this system as Moses throughout the thesis.

Source:	All the winners received a diploma.
SMT output:	Všem výhercům obdržel diplom.
Gloss:	To all the winners he received a diploma.

Example 1.1

1.1.2 Analysis

To be able to inspect and correct the structure of the sentences and their individual words, Depfix analyzes both the English and the Czech sentence. The most important result of the analysis are analytical dependency trees, shown in Figure 1.2.

1.1.3 Fixes

The translation contains several errors, which is typical for most sentences produced by today's SMT systems. However, Depfix is able to fix the errors,

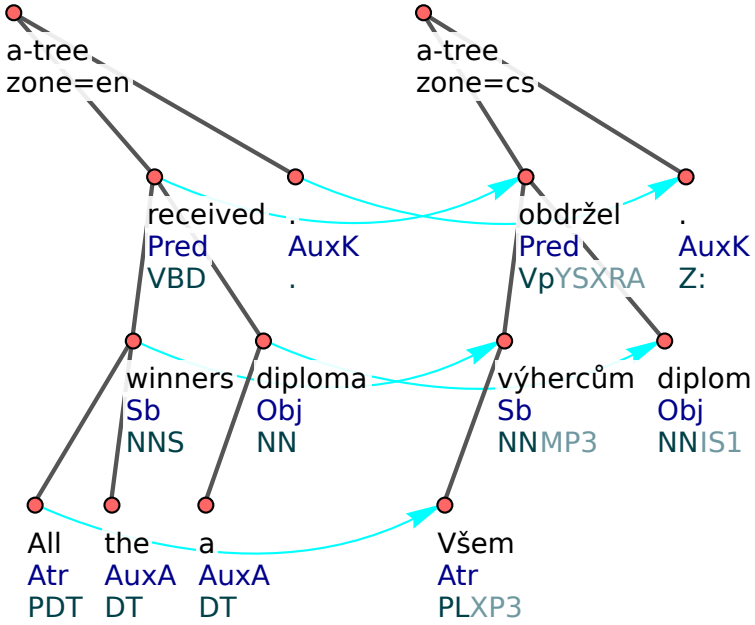


Figure 1.2: English and Czech analytical trees, word-aligned, with analytical functions and tags.

as can be seen in Example 1.2. The errors in this particular sentence are fixed by subsequently applying three fix rules:

1. "Subject morphological case" (Section 6.4.7), which changes the morphological case of the subject 'výhercům' ('winners') from dative to nominative ('výherci')

2. "Noun - adjective agreement" (Section 6.3.6), which changes the morphological case of the adjective-like pronoun 'všem' ('all') from dative to nominative and sets its morphological gender to masculine animate ('všichni'), to match the morphological case and morphological gender of the subject

3. "Subject - past participle agreement" (Section 6.3.3), which changes the morphological gender of the predicate 'obdržel' ('received') from singular to plural and its morphological gender from masculine to masculine animate ('obdrželi'), to match the morphological case and morphological gender of the subject

The word forms are then regenerated according to the new morphological categories.

1.1.4 Output

The output of Depfix is the corrected sentence. It does not contain any errors anymore, as Depfix was able to correct all of the errors in this sentence, without making any incorrections (i.e. paradiorthoses).

Source:	All the winners received a diploma.
SMT output:	**Všem výhercům obdržel** diplom.
Gloss:	**To all** the **winners he received** a diploma.
Depfix output:	**Všichni výherci obdrželi** diplom.
Gloss:	**All** the **winners received** a diploma.

Example 1.2

1.2 Thesis Structure

Chapter 2 provides a description of work related to automatic post-editing of machine translation outputs. Chapter 3 contains error analysis which motivates our work, described in the following chapters. Depfix description is divided by the layer on which the fixing is performed: the morphological layer, where only forms, lemmas and tags are present (Chapter 4), the analytical (shallow-syntax) layer, which adds the dependency tree structure (Chapter 5 and Chapter 6), and the tectogrammatical (deep-syntax) layer, providing some abstractions over the analytical layer (Chapter 7). Chapter 8 evaluates Depfix performance in detail and contains both automatic and manual evaluations. The thesis is concluded by Chapter 9.

We include a set of examples of Depfix outputs in Attachment A, the Treex scenarios used in Depfix in Attachment B and the feature sets used by our parser and labeller in Attachment C.

Chapter 2

Related Work

This chapter discussed work of other authors that is related to the task of automatic post-editing of statistical machine translation (SMT) outputs. We begin with a brief section on the quality of translations produced by SMT systems (Section 2.1). We then describe the existing research in the area of statistical post-editing of rule-based machine translation (Section 2.2), statistical post-editing of SMT (Section 2.3), and rule-based post-editing of SMT (Section 2.4).

Please note that the work related to error analysis of SMT outputs is part of Chapter 3, and dependency parsing related work is included directly in the sections of Chapter 5.

2.1 Statistical Machine Translation and its Quality

Statistical Machine Translation (SMT) is the current state-of-the-art approach to machine translation (MT) and its performance has been on a steady rise for several years – see e.g. (Callison-Burch et al., 2010, 2011, 2012). State-of-the-art SMT systems, such as Google Translate[1] or Moses-based SMT systems (Koehn et al., 2007), are successfully being employed into everyday use both by individuals and business, signalling that their performance has reached a level of being more useful than harmful.

However, the SMT outputs are still typically significantly worse than human translations, containing various types of errors, both in lexical choices and in grammar (see Chapter 3). Therefore, in real-world applications, SMT is usually employed only as one part of a more complex translation scenario, which is often represented by a computer-aided translation (CAT) tool (Langlais et al., 2000; Koehn, 2009b). CAT combines the power of an SMT system with the expertize of a human translator and has been shown by Koehn (2009a) to be more efficient than pure human translation – not only is the translation performed quicker, but also its quality is usually higher.

[1] `http://translate.google.com`

2.2 Statistical Post-editing of Rule-based MT

The first reported results of automatic post-editing of MT are (Simard et al., 2007) where the authors successfully performed statistical post-editing (SPE) of rule-based machine translation (RBMT) outputs. To perform the post-editing, they used a phrase-based statistical machine translation (SMT) system in a monolingual setting, trained on the outputs of the RBMT system as the source and the human-provided reference translations as the target, to achieve massive translation quality improvements. The authors also compared the post-edited RBMT performance to directly using the SMT in a bilingual setting, and reported that the SMT system alone performed worse than the post-edited RBMT. The authors also tried to post-edit the bilingual SMT system with another monolingual instance of the same SMT system, but concluded that no improvement in quality was observed.

2.3 Statistical Post-editing of SMT

The first known positive results in SPE of SMT are reported by Oflazer and El-Kahlout (2007) on English to Turkish machine translation. The authors followed a similar approach to Simard et al. (2007), training an SMT system to post-edit its own output. They use two iterations of post-editing to get an improvement of 0.47 BLEU points (Papineni et al., 2002). The authors used a rather small training set and do not discuss the scalability of their approach.

To the best of our knowledge, the best results reported so far for SPE of SMT are by Béchara et al. (2011) on French-to-English translation. The authors start by using a similar approach to Oflazer and El-Kahlout (2007), getting a statistically significant improvement of 0.65 BLEU points (Papineni et al., 2002). They then further improve the performance of their system by adding information from the source side into the post-editing system by concatenating some of the translated words with their source words, eventually reaching an improvement of 2.29 BLEU points. However, similarly to Oflazer and El-Kahlout (2007), the training data used are very small, and it is not clear how their method scales on larger training data.

In Section 7.4.1, we evaluate the utility of the approach of Béchara et al. (2011) for the English-Czech language pair and observe no improvement.

2.4 Rule-based Post-editing of SMT

All of post-editing systems known to us perform statistical post-editing (SPE), typically by training an SMT system in a monolingual setting. To the best of our knowledge, we are the first to have created a full rule-based post-editing (RBPE) system,[2] which we first presented in (Mareček et al., 2011).

[2]We believe that rule-based post-editing of rule-based machine translation is hard to define properly, as any rule-based block used to post-edit outputs of a rule-based MT system can be regarded as a part of the system itself. Therefore, we assume that RBPE can only be performed on outputs of statistical MT systems.

The approach of RBPE of SMT can be seen as being complementary to SPE of RBMT as performed by Simard et al. (2007) and thus can similarly benefit from the difference in issues of the rule-based and statistical approach, combining them to outperform both of them.

It is clear that even statistical MT systems contain rule-based components. However, these are typically only very simple scripts, performing such tasks as tokenization and detokenization or sentence-initial capitalization, and a set of such scripts cannot classify as a full post-editing system. If SMT systems try to tackle the same issues as we do, such as morphological agreement, they try to do so by statistical means.

An SMT system coupled with a RBPE system could also be classified as a hybrid MT system (Eisele et al., 2008), such as TectoMT (Popel and Žabokrtský, 2010). Hybrid MT systems combine RBMT and SMT systems or their parts in various ways, trying to improve over both of the approaches by exploiting both the linguistic knowledge present in RBMT systems and large-scale text analysis performed by SMT systems. While our work builds on similar assumptions on partial complementarity of the two approaches, it differs from the hybrid MT approach. Most importantly, our target is not a full MT system, but a standalone post-editing system to be used on top of an SMT system, which we treat as a black (or grey) box.

Chapter 3

Error Analysis of Statistical Machine Translation Outputs

Our work on Depfix is motivated by an analysis of error in SMT outputs, which we present in this chapter. We start from SMT error analyses performed by other researchers, which we describe in Section 3.1. We continue with two sections that describe errors that we found in the outputs of Moses on the WMT_{10} dataset, which we classify into two categories: lexical errors (Section 3.2, and word form errors (Section 3.3). We conclude with an overview of the results of the error analysis and detail our choice of errors that we attempt to correct by Depfix in Section 3.4.

Most of the examples of Moses errors presented in this chapter reappear later in the thesis when the part of Depfix that fixes the error is presented, together with the output that Depfix produces.

3.1 Related Work

Detailed error analyses of SMT systems have already been performed by several researchers. The first set of error analyses was done by Vilar et al. (2006), who analyzed errors of their SMT system (Vilar et al., 2005) on English to Spanish, Spanish to English and Chinese to English translation. The authors notice that the error statistics differ for different language pairs and translation directions.

An overview of existing SMT error analyses is given in (Fishel et al., 2012). The authors present a collection of corpora with annotations of translation errors for translation directions: French to German, German to English, English to Serbian, and English to Czech. Only the last direction is relevant for Depfix; however, this analysis was already presented in more detail in (Bojar, 2011).

The work by Bojar (2011) is of high interest for us, as, apart from analyzing the translation direction that we attempt to post-edit, it even analyzes the errors of a variant of Moses, i.e. the system that we post-edit in Depfix.

The analysis identifies the word form errors (or morphology errors) and the lexical errors (or lexical choice errors) as the first and second most common type of errors in outputs of Moses, respectively, with the lexical errors being very serious at the same time. This helped us to narrow our focus, disregarding error types that are both infrequent and not seen as serious, such as errors in word-order or in punctuation.

Moreover, the analysis also makes it clear that purely statistical MT systems, such as Moses, are generally better in lexical choices but worse in generating the correct word forms when compared to fully or partially rule-based systems, such as TectoMT (Popel and Žabokrtský, 2010). This makes us believe that Depfix, being a rule-based system for post-editing of SMT outputs, should focus on word form errors rather than lexical errors.

3.2 Lexical Errors

Lexical errors are, according to Bojar (2011), the second most common error type in Moses outputs, at the same time being one of the most severe errors. As a lexical error, we understand an incorrect lexical choice, a missing content word, or an extra content word.

We did not perform a detailed analyzes of lexical errors, as we decided not to focus on them in Depfix, as a higher gain is to be expected by correcting word form errors for reasons already mentioned in Section 3.1. Still, we detail three lexical error types that we noticed in our data and that we believe could be successfully addressed even by a rule-based post-editing system, provided that the system is able to perform translations of individual words, probably even using only a simple lexicon.

3.2.1 Missing Reflexive Verbs

We observed one kind of error that a rule-based system which employs at least basic linguistic analysis should be at least able to detect. This is the error of a missing reflexive verb, shown in Example 3.1.

Source:	. . . a thousand protesters *gathered* before the DTP's buildings in Diyarbakir. . .
SMT output:	. . . tisíce demonstrantů *se* před DTP je budovy v Diyarbakiru. . .
Gloss:	. . . thousands of protesters *themselves* before the DTP is buildings in Diyarbakir. . .
Correct:	. . . tisíce demonstrantů *se* **shromáždily** před DTP budovami v Diyarbakiru. . .
Gloss:	. . . thousands of protesters **gathered** *themselves* before the DTP buildings in Diyarbakir. . .

Example 3.1

A reflexive verb in Czech is a verb which carries a reflexive particle 'se' or 'si'. The particle is always a separate token, although it usually apeares near the verb.

SMT systems usually use unsupervised word-alignment, such as the one provided by GIZA++ (Och and Ney, 2003), to find out which word or phrase in one language corresponds to a given word or phrase in the other language. We believe that the error type being described is caused by the aligner erroneously

aligning the English verb to the Czech reflexive particle only, which can easily happen if the two are not adjacent. This can then lead to the SMT system translating the source verb by the particle only.

3.2.2 Fake Named Entities

A fake named entity error is an error where a word was not translated although it should have been, and is capitalized instead. Such a word is probably an out-of-vocabulary item of the Moses translation model, which has no other option than to leave the word untranslated. The statistical truecaser in Moses then encounters the untranslated word and capitalizes it, presumably believing it to be a named entity.

We believe that this error arises because Moses lowercases all its input, which makes it harder for it to detect named entities correctly.

See Example 4.8, where Moses probably mistook the word 'unleashed' for the title of a 2005 film,[1] although in the source sentence, 'unleashed' is used as a verb and is not even capitalized, making its interpretation as a named entity highly improbable.

Source:	Having *unleashed* 238 world records since February 2008...
SMT output:	S **Unleashed** 238 světové rekordy od února 2008...
Gloss:	With **Unleashed**$_{\textbf{as the title of the film}}$ 238 of world records since February 2008...
Correct:	S **uvolněním** 238 světových rekordů od února 2008...
Gloss:	With **unleashing** 238 world records since February 2008...

Example 3.2

3.2.3 Wrong Part-of-speech

A type of error that is somewhere in between a lexical error and a word form error is an error in the part-of-speech, although the lexical choice is correct. We most often observed the part-of-speech error in a situation where a verb was incorrectly translated as a noun, as in Example 3.3. This typically leads to a predicate-less sentence, which is, according to some researchers, such as (Lo and Wu, 2011), one of the most serious errors that an SMT system can make.

In English there is often much ambiguity between nouns, verbs and adjectives. See Example 3.4, where the word 'school' is used as three different parts of speech, also providing the correct Czech translations.

[1] `http://www.imdb.com/title/tt0342258/`

Source:	...who *place* wreaths on graves at Arlington.
SMT output:	...který **místo** věnce na hroby v Arlingtonu.
Gloss:	...who **location** wreaths on graves at Arlington.
Correct:	...kteří **položili** věnce na hroby v Arlingtonu.
Gloss:	...who **placed** wreaths on graves at Arlington.

Example 3.3

Noun:	I go to **school**.
Czech:	Chodím do **školy**.
Adjective:	I use the **school** bus.
Czech:	Používám **školní** autobus.
Verb:	I **school** my students.
Czech:	**Školím** své studenty.

Example 3.4

3.3 Word Form Errors

The errors in word form are, according to Bojar (2011), the most common type of errors in the outputs of Moses, and are expected to be easier to correct by a rule-based system.

Word form errors are errors where the lexical choice made by the SMT system is correct, but the word form is not correct because of a wrong inflection of the word. Typically, this would be an incorrect morphological number, morphological case, morphological gender, person or tense, i.e. an error in a value of a morphological category of the word's morphological tag, as defined in Section 4.1; we also include errors in negation into this category, as negation is also expressed by the morphological tag.

We also understand an incorrect, missing or extra auxiliary word, such as an auxiliary verb or a preposition, to be a word form error. Using the notion of tectogrammatics, described in Section 7.1, a word form error is defined as an error which can be corrected without changing any t-lemma and without adding or removing any t-node.

3.3.1 Agreement Errors

One of the most common type of error that we observe in our data is an agreement error.

The rules of Czech grammar require morphological agreement among many parts of a sentence, such a predicate and its subject ('trpaslíci zpívají' – 'the dwarfs sing'), a preposition and its noun ('bez peněz' – 'without money'), or a noun and its attribute ('veliká ostuda' – 'a big shame'). The words typically have to agree in one or more morphological attributes, such as the morphological

number (singular/plural), the morphological gender (masculine/feminine/...), the morphological case (nominative/genitive/...), and other.

Source:	...the total *bonuses* awarded by the business this year, despite today's announcement, *exceeds* 20 billion dollars.
SMT output:	...celkové *odměny* udělované byznys letos navzdory dnešní oznámení, **přesahuje** 20 miliard dolarů.
Gloss:	...the total $bonuses_{pl}$ awarded the business this year despite today announcement, $\mathbf{exceeds_{sg}}$ 20 billion dollars.
Correct:	...celkové *odměny* udělované letos byznysem **přesahují**, navzdory dnešnímu oznámení, 20 miliard dolarů.
Gloss:	...the total $bonuses_{pl}$ awarded this year by the business $\mathbf{exceed_{pl}}$, despite today's announcement, 20 billion dollars.

Example 3.5

However, the outputs of SMT systems often contain a lot of errors in agreement, especially on long-distance dependencies, such as the subject-predicate pair in Example 3.5 where the agreement in morphological number is violated. In the example, there are at least 9 tokens between the subject and the predicate on the source side and at least 6 tokens on the target side, which is beyond both the translation model phrase length and the language model n-gram length of a typical today's SMT system. To capture such a long-distance agreement, the MT system typically has to employ a more advanced approach than a simple phrase-based one, such as a support for gapped phrases, or a very strong language model.[2] Moreover, the agreement is often violated even on adjacent word pairs, as shown by the violated noun-adjective agreement in morphological gender in Example 3.6.

Source:	...*this half-hearted increase* will bear the same fruit...
SMT output:	...**tato polovičatá** *nárůst* bude nést stejné ovoce...
Gloss:	...$\mathbf{this_{fem}}$ $\mathbf{half\text{-}hearted_{fem}}$ $increase_{masc}$ will bear the same fruit...
Correct:	...**tento polovičatý** *nárůst* ponese stejné ovoce...
Gloss:	...$\mathbf{this_{masc}}$ $\mathbf{half\text{-}hearted_{masc}}$ $increase_{masc}$ will bear the same fruit...

Example 3.6

[2]However, a strong language model with a poor translation model can lead to serious hidden errors, as such an SMT system often produces translations that are fluent and grammatically correct, but convey a different meaning than the source sentence.

3.3.2 Valency Errors

Another type of errors that we often encountered in our data are errors in *valency*. The term *valency*, explained in detail in Section 7.1.4, stands for the way in which governing words and their arguments, typically predicates and their noun arguments, are used together. In Czech, valency is usually expressed in choice of prepositions and morphological cases on the arguments; in English, it is typically the prepositions only.

Source:	... we need to *spend on* our middle *schools*.
SMT output:	... musíme **utrácet** *naše střední školy*.
Gloss:	... we must **destroy** *our middle schools*.
Correct:	... musíme **utrácet za** *za naše střední školy*.
Gloss:	... we must **spend on** *our middle schools*.

Example 3.7

Example 3.7 shows Moses making an error in the valency of the 'utrácet – škola' ('spend – school') pair. The missing preposition changes the meaning dramatically, as the verb 'utrácet' is polysemous and can mean 'to spend (esp. money)' as well as 'to kill, to destroy (esp. animals)'. The seriousness of valency errors is usually not so high (although similar cases are not completely uncommon), but they are a common error type in our data and lower the quality of the output.

SMT systems are typically good at capturing valency correctly by means of both the translation model and the language model, provided that it is expressed locally enough to fit into a reasonable phrase length and/or n-gram length. For example, the phrase translation 'spends on' – 'utrácí za' is likely to be contained in the translation tables, and the 'za školy' ('on schools') n-gram or even the 'utrácí za školy' ('spends on schools') n-gram is likely to be assigned a high enough score by the language model, together enabling the SMT system to yield a correct translation. However, even a single inserted word, such as 'middle' ('střední'), significantly increases data sparseness, and can thus introduce a new phrase boundary and/or make the n-gram scores much less reliable, which in turn often leads to translation errors.

We generally believe that treating translation of valency frames by statistical methods is reasonable. In our opinion, it is the lack of linguistic abstraction, especially in determining the governor-argument relations, which accounts for the observed high frequency of valency errors.

3.3.3 Errors in Transfer of Meaning to Morphology

There is a large group of various errors which could be described as errors in transferring a meaning from the source sentence into morphological attributes of a word or words in the target sentence.

A typical example of such an error is the translation of a subject. Here, the meaning is the property of a word being a subject to a predicate. In English,

a subject is typically marked by being a left constituent of the predicate, while in Czech, a subject is marked by being in the nominative morphological case. Without a proper analysis of the source sentence, which is able to distinguish the source subject, and an approach to Czech morphology able to put the target subject into the nominative morphological case, SMT systems often fail in transferring the meaning correctly, as shown in Example 3.8.

Source:	At a time when Swiss *voters have called* for a ban on the construction of minarets...
SMT output:	V době, kdy švýcarské **voliče** vyzvali k zákazu výstavby minaretů...
Gloss:	At a time, when Swiss **voters$_{\mathbf{acc}}$** *were called* to ban the construction of minarets...
Correct:	V době, kdy švýcarští **voliči** vyzvali k zákazu výstavby minaretů...
Gloss:	At a time, when Swiss **voters$_{\mathbf{nom}}$** *called* for a ban on the construction of minarets...

Example 3.8

We found many similar situations where a meaning, expressed by various means in English, is to be expressed by specific values of morphological attributes in Czech, but the transfer is often not performed correctly by Moses. We list such cases in a simplified way in Table 3.1, which is accompanied by a set of examples showing Moses making an error in the transfer of the meaning.

Transfer of verb tenses and transfer of negation are more complicated, therefore they are not included in the table but are described separately.

Source:	*I* don't blame them.
SMT output:	**Já** se jim *nedivím*.
Gloss:	**I** them *don't-blame$_{1st\ person\ sg}$*.
Correct:	*Nedivím* se jim.
Gloss:	*Don't-blame$_{1st\ person\ sg}$* them.

Example 3.9

Errors in transfer of verb tenses

The systems of verb tenses in English and Czech are very different, and there is no exact one-to-one or many-to-one correspondence between them. Moses does not handle verb tenses explicitly. It neither employs analysis of the verb tenses nor even tries to translate a whole compound verb form as a one unit. The result is that the transfer of tenses is not consistent and sometimes is even rather random – consider Example 3.16 and Example 3.17, where the past and present tenses are switched in both directions. In the data, we have seen all kinds of tenses

Meaning	English	Czech	Moses error
Subject	left constituent of verb	nominative morphological case	Example 3.8
Morphological gender, number and person of a general subject	morphological gender, number and person of the subject pronoun	morphological gender, number and person of the predicate	Example 3.9
Possessor	possessive ending ''s', or preposition 'of'	possessive adjective form, or genitive morphological case	Example 3.10
Something possessed by the subject	possessive adjective according to morphological gender, number and person of the subject	reflexive possessive pronoun 'svůj'	Example 3.11
Current action	present form of 'be' and gerund of the content verb	present form of the content verb	Example 3.12
Action with unexpressed actor	the passive	the deagentive with the reflexive particle 'se', or the passive	Example 3.13
Passive actor	preposition 'by'	instrumental morphological case or active construction	Example 3.14
Plural noun	singular noun with suffix 's'	plural morphological number, expressed in many ways based on morphological case and paradigm	Example 3.15

Table 3.1: The differences in ways in which some meanings are usually expressed in English and in Czech, with references to examples of Moses errors in the transfer of the meaning.

Source:	... unsustainable deficit level *of public finances*.
SMT output:	... neudržitelná úroveň schodku **veřejné finance**.
Gloss:	... unsustainable deficit level **public finances**$_{\textbf{nominative}}$.
Correct:	... neudržitelná úroveň schodku **veřejných financí**.
Gloss:	... unsustainable deficit level **of public finances**$_{\textbf{genitive}}$.

Example 3.10

Source:	During the Triassic period, the *dinosaurs* still shared *their* habitat with a large amount of other reptiles...
SMT output:	Během doba triasu, *dinosauři* stále sdíleli **jejich** stanoviště s velkým množstvím další plazi...
Gloss:	During the Triassic period, the *dinosaurs* still shared **their$_{\textbf{possessive}}$** habitat with a large amount of other reptiles...
Correct:	Během doby triasu, *dinosauři* stále sdíleli **své** stanoviště s velkým množstvím další plazů...
Gloss:	During the Triassic period, the *dinosaurs* still shared **one's$_{\textbf{reflexive possessive}}$** habitat with a large amount of other reptiles...

Example 3.11

Source:	...she *is explaining* mathematical equations...
SMT output:	...**je vysvětlovat** matematické rovnice...
Gloss:	...she **is explain$_{\textbf{infinitive}}$** mathematical equations...
Correct:	...**vysvětluje** matematické rovnice...
Gloss:	...she **explains** mathematical equations...

Example 3.12

Source:	...a wave of them *was expected*.
SMT output:	...vlna z nich **očekává**.
Gloss:	...a wave from them **expects**.
Deagentive:	...vlna **se** jich **očekává**.
Gloss:	...a wave of them **expects itself**.
Passive:	...vlna jich **byla očekávána**.
Gloss:	...a wave of them **was expected**.

Example 3.13

Source:	The timing of his strategy is foiled *by his voluntarism*.
SMT output:	Načasování jeho strategie je zmařena **jeho voluntarismu**.
Gloss:	The timing of his strategy is foiled **of his voluntarism$_{\textbf{genitive}}$**.
Correct:	Načasování jeho strategie je zmařeno **jeho voluntarizmem**.
Gloss:	The timing of his strategy is foiled **by his voluntarism$_{\textbf{instrumental}}$**.

Example 3.14

Source:	. . . a slander suit against three ČSSD *commissioners*.
SMT output:	. . . soudní proces proti třem ČSSD **komisaři**.
Gloss:	. . . a suit against three ČSSD **commissioner**$_{\textbf{singular dative/plural nominative}}$.
Correct:	. . . soudní proces pro pomluvu proti třem ČSSD **komisařům**.
Gloss:	. . . a slander suit against three ČSSD **commissioners**$_{\textbf{plural dative}}$.

Example 3.15

switched in translation, including switching past tenses with future tenses, as in Example 3.18.

Source:	The generals *are defending* themselves. . .
SMT output:	Generálové se **bránili**. . .
Gloss:	The generals **were defending** themselves. . .
Correct:	Generálové se **brání**. . .
Gloss:	The generals **are defending** themselves. . .

Example 3.16

Source:	Amnesty also *cited* the case of a former detainee. . .
SMT output:	Amnesty rovněž **cituje** případ bývalého vězně. . .
Gloss:	Amnesty also **cites** the case of a former detainee. . .
Correct:	Amnesty rovněž **citovala** případ bývalého vězně. . .
Gloss:	Amnesty also **cited** the case of a former detainee. . .

Example 3.17

Compound verb forms, such as the present continuous or past simple passive, are often mistranslated by Moses. The most common error is that he auxiliary verbs ('be', 'have') are translated literally, while usually the way in which the corresponding verb forms are composed in English and in Czech are different. Example 3.12 showed a case where 'is explaining' is translated word-by-word by Moses, instead of using the correct one-word verb form.

Errors in transfer of negation

Errors in negation, especially verb negation, are uncomfortably frequent. It should be noted that a negation error is a rather serious one: a missing or extra negation typically inverts the meaning of a phrase or even of the whole sentence, and, to make matters worse, such error can be very difficult to spot by the user. Therefore, the user can be seriously misled by the translation, putting them in a

Source:	... the direct service from Prague - Letohrad *will be cut* dramatically.
SMT output:	... přímé spojení z Prahy - letohrad se dramaticky **snížil**.
Gloss:	... the direct service from Prague - Letohrad **was lowered** dramatically.
Correct:	... přímé spojení z Prahy do Letohradu se dramaticky **sníží**.
Gloss:	... the direct service from Prague to Letohrad **will be lowered** dramatically.

Example 3.18

much worse situation than an incomprehensible translation: an incomprehensible translation's value is close to none, while a value of a misleading translation is negative.

A missing negation, such as the one in Example 3.19, is much more frequent than an extra negation, although both of them occur in our data.

Source:	... he feels that he *does not* wholly *deserve* it.
SMT output:	... cítí, že si plně **zaslouží**.
Gloss:	... he feels that he wholly **deserves**.
Correct:	... cítí, že si ji plně **nezaslouží**.
Gloss:	... he feels that he wholly **does not deserve** it.

Example 3.19

The error is most probably caused by difficulties in aligning the negation correctly, as in Czech it is usually expressed by a negative prefix, 'ne-', while English often uses the word 'not'. Therefore, the English positive verb is often aligned to the Czech negative verb and 'not' stays unaligned. This poses two threats:

- In case of a positive English sentence, there is the threat of translating the English verb by a negative Czech verb. This does not happen too often as the positive Czech verb is likely to occur more frequently in the training data than the negative one, and it is therefore more probable that the decoder will select the positive Czech verb as the translation. However, this error does occur, especially when the verb is rare in the training data.
- In case of a negative English sentence, there is the threat of leaving out the 'not' word and translating the English verb by a positive Czech verb. This happens quite often, as, for aforementioned reasons, the decoder is likely to select the positive Czech verb as a more probable translation than the negative one.

To the best of our knowledge, Moses does little to overcome this issue. Thus, the translation model probably often generates both the positive and the negative

Czech verb as possible translations (the positive one presumably having a higher score than the negative one), and it is up to the language model to choose, which variant is better. Obviously, the language model is rather unsuitable for making such a choice (apart from fixed expressions), and it should therefore not be surprising that the choice is often erroneous.

3.3.4 Noun Clusters

A common error of Moses is translating English noun clusters by Czech noun clusters, as shown in Example 3.20 (although many of the noun clusters were translated correctly). A noun cluster is a chain of nouns with the head noun at the end, the preceding nouns being modifiers of the following nouns. Although noun clusters are starting to appear even in Czech language, as in 'O_2 mobilní internet sada' ('O_2 mobile internet set'), they are typically considered incorrect, and should usually be translated by a structure with the head noun being the first one and the modifiers put after the head noun, typically in the genitive morphological case, such as 'sada mobilního internetu O_2' ('set of mobile internet of O_2'). Usually, the correct order of the nouns in Czech is reverse to their order in English.

Source:	...the chairman of the Litoměřice$_1$ Nature$_2$ Protection$_3$ Society$_4$ civil association$_5$...
SMT output:	...předseda občanského **sdružení$_5$ Litoměřice$_1$ ochrana$_3$ přírody$_2$ společnosti$_4$**...
Gloss:	...the chairman of the civil **association$_5$ Litoměřice$_1$ protection$_3$ of nature$_2$ society$_4$**...
Correct:	...předseda občanského **sdružení$_5$ Společnost$_4$ ochrany$_3$ přírody$_2$ Litoměřice$_1$**...
Gloss:	...the chairman of the civil **association$_5$ Society$_4$ of protection$_3$ of nature$_2$ Litoměřice$_1$**...

Example 3.20

3.3.5 Casing Errors

The casing of words is often incorrect, and, although such an error is typically not serious, it can occasionally mislead the user of the translation. Most often, a word is not capitalized even when it should be; usually it is a word that exists as a lowercase word in Czech, but in the particular sentence it is a part of a named entity and therefore should be capitalized. For example, the word 'psa' can be a form of the word 'pes' ('dog'), but it can also be the shortcut for 'Peugeot Société Anonyme', as in Example 3.21.

We believe that the main cause of the casing errors is the fact that Moses is case-insensitive: it lowercases all its input prior to any processing, and only performs statistical truecasing on its final output. Moreover, the standard MT quality metrics are typically case-insensitive, and therefore the researchers in

Source:	Suicides at France Telecom, *PSA* and Renault...
SMT output:	Sebevražd y na France Telecom, **psa** a renault...
Gloss:	Suicides on France Telecom, *a dog* and renault...
Correct:	Sebevražd y ve France Telecom, **PSA** a Renaultu...
Gloss:	Suicides at France Telecom, *PSA* and Renault...

Example 3.21

SMT are not motivated by the automatic evaluation to provide correctly cased outputs.

We also observed cases where words are capitalized even if they should not. We already discussed these cases, referred to as fake named entities, in Section 3.2.2.

3.3.6 Tokenization Errors

To decrease the amount of out-of-vocabulary items, Moses uses an aggressive tokenization strategy, which is most apparent in splitting English dash-separated compounds into separate tokens. This strategy allows it to translate the individual words that form the compound, even if the whole compound is unknown to the system, and is therefore generally good.

However, Moses pays little attention to detokenizing its output, leaving many superfluous spaces in the translation, as shown in Example 3.22. Although such errors are of a very low importance and are unlikely to confuse the user of the translation, they still may distract them, and they can consider the subjective quality of the translation to be lower.

Source:	on-line	al-Somali	Jean-Marie
SMT output:	on - line	al - Somali	Jean - Marie

Example 3.22

We noticed that tokenization errors are severely penalized by the automatic translation quality evaluation metrics, as these are typically token-based.

3.4 What We Decided to Fix in Depfix

We found that outputs of Moses contain many errors of many types, which we divided into two categories – the lexical errors and the word form errors.

The lexical errors are very common and at the same time one of the most serious errors in SMT outputs, and fixing them would probably bring a very high improvement of translation quality. However, they are generally hard to fix by rule-based methods, as the statistical systems typically outperform rule-based systems in lexical choices. Still, there are two types of lexical errors that we decided to try to fix in Depfix. These are the missing reflexive verbs, which we try to fix in "Missing reflexive verbs" (Section 6.4.1), and fake named entities, which

we address in "Source-aware truecasing" (Section 4.3.1). In future, we would like to research the possibilities of detecting and correcting the part-of-speech errors described in Section 3.2.3.

We decided to focus on word form errors in our work, as they are both the most common type of error of Moses and are typically better approached by rule-based systems than by purely statistical systems. Probably the most frequent are the errors in morphological agreement, which we address by a set of 7 rules in Section 6.3, and in valency, for the correction of which we decided to employ a statistical approach, described in Section 7.4, as valency most probably cannot be captured by a small set of rules.

There are also many errors in the transfer of a meaning that is to be expressed by means of morphology in Czech, such as a verb tense or possessiveness. We address these errors by a set of rules described in Section 6.4, except for the transfer of verb tenses and of negation, which have to be handled on a deeper analysis level and are described in Section 7.3.

Some of the least serious errors that we encountered, although quite frequent, are the errors in casing and tokenization. We decided to try to fix them only in cases where the correct casing or tokenization can be simply projected from the source to target, as described in Section 4.2.2 and Section 4.3.1, as the effort necessary to employ such fixes is minimal.

There are still many error types, such as the translations of noun clusters or the usage of the reflexive possessive pronoun 'svůj', that we decided not to address in Depfix. In our opinion, these errors usually belong neither to serious nor common errors (such as the lexical or agreement errors), and do not seem to be completely trivial to fix either (such as the casing errors), and therefore we decided to focus on other error types instead. However, we believe that even the error types that are disregarded by Depfix should be addressed in future, as correcting them would probably lead to a small but still positive change of the translation quality.

Chapter 4

Morphological Layer

The morphological layer, or m-layer, as defined in PDT (Hajič et al., 2006), is the first layer on which the input, i.e. the English source sentence and the Czech target sentence, are analyzed and fixed.

We describe the representation of the sentences on this layer in Section 4.1. Section 4.2 details the analysis steps that construct the m-layer representation, and the way we fix them for an improved performance. In Section 4.3, we detail Depfix fix rules that operate on m-layer.

4.1 M-nodes, Lemmas and Tags

The m-layer representation of a sentence is flat. It only describes the morphology of individual tokens of the sentence, represented by m-nodes, without any relations between them.

Each m-node corresponds to one word or to one piece of punctuation. It contains several attributes that describe the represented token, the most important being:

- the form, which is identical to the token as it appears in the sentence
- the tag – morphological tag for Czech m-nodes (see Section 4.1.1), or the POS tag for English m-nodes (see Section 4.1.2
- the lemma (see Section 4.1.3)

The other attributes are rather technical, specifying the order of the token in the sentence, whether it is followed by a space, etc.

4.1.1 Czech Morphological Tags

The Czech tagset uses positional morphological tags. The tag contains full morphological analysis of the respective word. We present a brief overview of the structure of the morphological tag in Table 4.1; a full description can be found in the PDT 2.0 manual (Hajič et al., 2006).[1] The default value is usually "-", meaning "not relevant", and there is also a universal "X" value, meaning "any or

position	description	values	example of values
1	part of speech	11	N (noun), V (verb), P (pronoun), R (preposition), A (adjective)
2	detailed part of speech	74	B (verb, present or future form), s (verb, past participle, passive
3	morphological gender	9	F (feminine), H (feminine/neuter)
4	morphological number	4	S (singular), P (plural)
5	morphological case	7	1 (nominative), 2 (genitive), 3 (dative), 4 (accusative), 5 (vocative), 6 (locative), 7 (instrumental)
6	possessor's morphological gender	3	M (masculine animate), F (feminine), Z (not feminine)
7	possessor's morphological number	2	S (singular), P (plural)
8	person	3	1 (I/we), 2 (you), 3 (he/she/it/they)
9	tense	4	R (past), P (present), F (future), H (past/present)
10	degree of comparison	3	1 (positive), 2 (comparative), 3 (superlative)
11	negation	2	A (affirmative), N (negated)
12	voice	2	A (active), P (passive)
13	unused	0	
14	unused	0	
15	variant, style, register	9	2 (archaic), 5 (colloquial)

Table 4.1: An overview of morphological tags, used on the target side. Adapted from the PDT 2:0 manual (Hajič et al., 2006).

Tag	Meaning
NN	Noun, singular or mass
NNS	Noun, plural
NP	Proper noun, singular
NPS	Proper noun, plural
POS	Possessive ending
MD	Modal
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present

Table 4.2: Some of the POS tags, used on the source side

not recognized" (these two values are not included in the count of possible values in the table).

The tagset is very suitable for the purpose of Depfix, since it is very easy to both check and change the individual morphological categories. At the same time, the large number of possible combinations of the values and thus of possible morphological tags makes some of the tags very sparse.[2]

4.1.2 English POS Tags

For English, we use the Penn Treebank tagset (Marcus et al., 1993), which defines 45 different tags. Table 4.2 lists some of the tags, focusing on the ones that we use in the text of this work; a full description can be found online.[3]

4.1.3 Lemmas

The lemma is the base form of a word, common for all inflections (but not derivations) of the word. It typically is the form with the following properties, if appropriate for the given part-of-speech, listed according to their importance in descending order: nominative morphological case, singular morphological number, masculine (animate) morphological gender, infinitive, affirmative, present tense, 1st person.

The level of abstraction of Czech lemmas is slightly higher than that of the English ones. For example, the Czech pronouns 'já' ('I'), 'ho' ('him') and 'jejich' ('their') all share the same lemma, 'já', whereas in English, each of the pronoun forms is its own lemma.

The lemma can have a "tail", i.e. a string appended to the form that further specifies it. It can specify the variant of a polysemous lemma, the type of named entity represented by the lemma, etc. However, we disregard the tails in Depfix, stripping them for most subsequent processing steps.

[1] `http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/ch02s02s01.html`

[2] There are several thousands of possible morphological tags.

[3] `http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

A pair of a lemma and a tag unambiguously defines a word form (except for a small number of exceptions), which significantly simplifies further processing.

4.2 Analysis, its Errors and their Fixes

The morphological layer is the first layer on which the input, i.e. the English source sentence and the Czech target sentence, are analyzed. We use a pipeline of state-of-the-art NLP tools integrated in Treex, as described in Section 4.2.1.

However, the analysis of the Czech sentences provided by the tools is far from perfect. One of the reasons is that all of the tools are intended to be used on correct sentences, which the target sentence is often not. In such a setup, the performance of the tools can be expected to be lower.

We therefore proceed with correcting some of the errors of the tools, typically by exploiting the information from the source sentence, as the tools themselves are typically monolingual and are unable to make use of the knowledge of the English sentence. Most of the corrections are performed during the analysis, immediately after getting the output from the tool that they fix, as the subsequent processing phases typically rely on the outputs of the previous ones.

4.2.1 Analysis Pipeline

The sentences are processed by the following pipeline of NLP tools integrated in Treex:

1. **tokenization** of the English and Czech sentences, performed by a rule-based tokenizer,

2. **tagging and lemmatization** of each token, performed:

   - by the Featurama tagger[4] for the Czech sentence, using the morphology created by Hajič (2004),
   - by the Morče tagger (Spoustová et al., 2007) for the English sentence,

3. **word alignment** of each pair of source sentence and target sentence, performed by GIZA++ (Och and Ney, 2003); we use the intersection symmetrization, which means that each token can be aligned to at most one other token in the parallel sentence,

4. **named entities recognition** on the English sentence, using the Stanford named entity recognizer (Finkel et al., 2005); there is currently no good named entity recognizer for Czech in Treex.

Please note that this description only lists the important parts of the analysis process, omitting several steps that are rather technical. The full Treex scenario is shown in Attachment B.

[4] `http://featurama.sourceforge.net/`

4.2.2 Fixing The Tokenization Errors: Tokenization Projection

The target sentence tokenizer does not make use of the source sentence, and therefore sometimes tokenizes the target sentence differently from the source sentence – see Example 4.1. This makes the subsequent word-alignment step unnecessarily harder, which in turn lowers the performance of many following processing phases (all that make use of both source and target).

Source:	on-line	al-Somali	Jean-Marie
SMT output:	on - line	al - Somali	Jean - Marie

Example 4.1

This kind of errors probably should not be considered to be a shortcoming of the tokenizer, but of the Moses system, which makes frequent errors in tokenization, as described in Section 3.3.6. In fact, it is highly beneficial in many applications if the tokenizer is simple and consistent, even if this means that the tokenization is not always correct theoretically. However, for the needs of Depfix, the blame is irrelevant. What is relevant is the fact that such cross-lingually incoherent tokenization poses unnecessary obstacles for Depfix, increasing the risk of incorrections. Therefore, we attempt to fix such errors in the tokenizer output, partially retokenizing the target sentence according to the source sentence.

We therefore apply a preprocessing phase before invoking the tokenizer, performing "Tokenization projection" by removing superfluous spaces from the target sentence to match the source sentence. This leads to the words that are identical in the source and the target, which are mostly named entities, to be tokenized in the same way – such as the ones in Example 4.1. To further increase the recall, we also assume the words to match if they are identical after removing the diacritics and lowercasing, as in Example 4.2.

Source:	The RATP segment runs from *Saint-Germain-en-Laye* and Nanterre to *Boissy-Saint-Léger* and *Marne-La-Vallée*.
SMT output:	RATP segment běží od **Saint - Germain - en - Laye** a Nanterre na **Boissy - Saint - Léger** a **Marně - La - Vallée**.
Depfix output:	RATP segment běží od **Saint-Germain-en-Laye** a Nanterre na **Boissy-Saint-Léger** a **Marně-La-Vallée**.

Example 4.2

The rule is language-ignorant, i.e. it does not account for any language-specific tokenization differences, such the different ways to tokenize units in English and in Czech. It therefore makes occasional incorrections, as shown in Example 4.3.

However, for Depfix processing, such consistent tokenization is actually better than the correct one. We therefore do not try to avoid the errors, but we employ a clever detokenizer at the end of the pipeline which corrects the tokenization of

Source:	He weighs *130kg*.
SMT output:	Váží **130 kg**.
Depfix output:	Váží **130kg**.
Correct:	Váží *130 kg*.

Example 4.3

units.[5] Thus, the processing steps in Depfix benefit from the higher similarity of the tokenizations of the two sentences, but the tokenization on the output is usually correct.

Example 4.4 shows such case. If "Tokenization projection" is not applied, the parser parses the sentence incorrectly, and "Preposition - noun agreement" (Section 6.3.1) cannot be applied as the noun 'svah' ('slope') is not a child of the preposition 'před' ('before'). However, changing the tokenization helps the parser, the sentence is parsed correctly, and "Preposition - noun agreement" can be applied. Both the incorrect and the correct parse trees are shown in Figure 4.1.

Source:	. . . a short jump ramp before a *100m*-long slope.
SMT output:	. . . krátký skok rampy *před 100 m* - **dlouhý svah**.
Gloss:	. . . a short ramp jump $before_{loc}$ *100m* - **a long slope$_{nom}$**.
Depfix output:	. . . krátký skok rampy *před 100 m* - **dlouhým svahem**.
Gloss:	. . . a short ramp jump $before_{loc}$ a *100m*-**long slope$_{loc}$**.

Example 4.4

This rule serves mainly as an analysis fixing rule for our purposes, but it also changes the final output by removing some intra-word spaces. The effect on the quality of the translation is not crucial, but it leads to an increase of automatic scores, since they are typically token-based and penalize different tokenization severely.

4.2.3 Fixing The Tagging Errors: Fixing Morphological Number of Nouns

The morphological tags assigned by the tagger often contain errors, especially in morphological number and morphological case. Such errors are generally common, since in Czech, there is much homonymy here – see Table 4.3 for a few examples. However, the errors are significantly more frequent in the analysis of SMT outputs. Because of the erroneousness of SMT outputs, ambiguity in selecting the correct morphological tag is higher – if the word form is wrong, it is often impossible to tell the morphological case and morphological number that it is in. Nevertheless, the tagger often produces analyses that are clearly wrong.

[5] We created a list of common units that the detokenizer recognizes and retokenizes, which are: m, g, l, s, b, B, V, A (including variants with prefixes m, c, d, h, k, M, and G), h, and min.

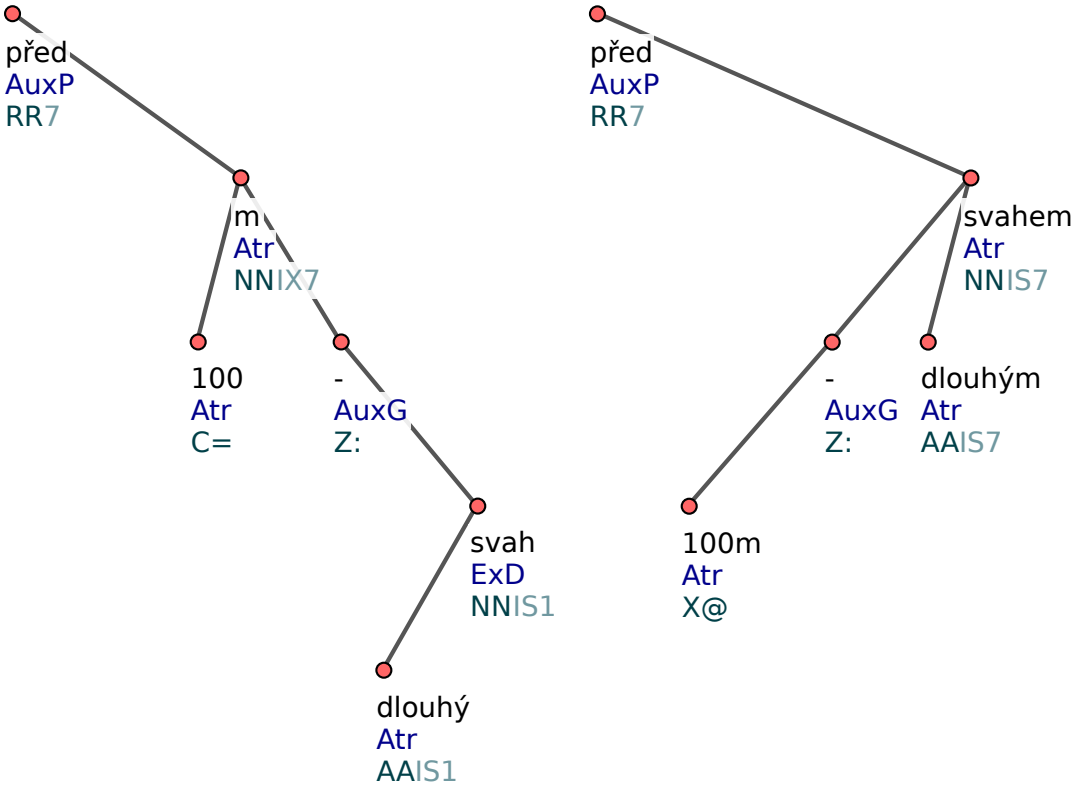


Figure 4.1: Tokenization projection

Word form	Possible analyses	
	number	case
hrad	singular	nominative
	singular	accusative
hradu	singular	genitive
	singular	dative
	singular	locative
feny	singular	genitive
	plural	nominative
	plural	accusative
	plural	vocative

Table 4.3: Examples of noun forms homonymy in Czech.

We believe that the source sentence should be used to provide additional information that could help with the disambiguation of the possible analyses. We implement this idea in Fixing morphological number of nouns, where we try to use the morphological number of the English words to choose the correct morphological number analysis for the Czech words.

It would be nice to also use the source sentence to choose the correct morphological case analysis, but this would be much less straight-forward to do. It could probably be done by finding an approximate mapping from English analytical functions and prepositions to Czech morphological cases, but we have not followed this research path – however, we take a similar approach in correcting valency errors, as described in Section 7.4.

Fixing morphological number of nouns assumes that the word form is correct, only the morphological tag is incorrect, and performs only such fixes that preserve the word form. Moreover, it is applied only if the Czech noun is singular and the aligned English noun is plural (the `NNS` POS tag), as the inverse case performs poorly on our development data – this is mostly because English often uses the singular morphological number even when referring to multiple entities by uncountable nouns, whereas Czech often uses the plural morphological number in such cases. Of course, the same problem does exist even in the other direction, i.e. an English plural noun corresponding to a Czech singular noun, but is much less common.

The rule first tries to only switch the Czech morphological number from singular to plural and checks whether the new morphological tag is consistent with the word form. If it is not, it tries to also change the morphological case, trying out all of the Czech morphological cases, until it finds a morphological case that is consistent with the word form and the plural morphological number. If such an alternative analysis is found, it is assumed to be correct (or at least better than the original analysis) and the morphological tag of the noun is changed.

See Example 4.5, where the noun 'komisaři' is first reanalyzed from 'comissioner$_{\text{sg dative}}$' to 'comissioner$_{\text{pl nominative}}$', based on the plural in source. A subsequent application of "Preposition - noun agreement" (Section 6.3.1) then correctly changes the morphological case of the noun to dative, based on the morphological case of the preposition 'proti' ('against'). Note that if Fixing morphological number of nouns was not applied, the morphological case would not be fixed by Preposition - noun agreement, as the original analysis indicated the noun to already bear the correct morphological case.

Source:	. . . a slander suit against three ČSSD *commissioners*$_{pl}$.
SMT output:	. . . soudní proces *proti* třem ČSSD **komisaři**.
Gloss:	. . . a slander suit *against*$_{dative}$ three ČSSD **commissioner$_{\text{sg dative}}$**.
Depfix output:	. . . soudní proces *proti* třem ČSSD **komisařům**.
Gloss:	. . . a slander suit *against*$_{dative}$ three ČSSD **commissioners$_{\text{pl dative}}$**.

Example 4.5

A similar fixing rule is "Prepositional morphological case" (Section 6.2.3), which tries to correct errors in the analysis provided by both the tagger and the lemmatizer for nouns that are part of a prepositional phrase, so that the morphological case of the noun is consistent with its parent preposition.

We leave further exploration of possibilities of providing source information to the target tagger as future work. Based on similar experiments with the parser (see Section 5.5), we believe that this would significantly improve the performance of the tagger.

4.2.4 Lemmatization Errors

We have not observed a substantially larger amount of errors in lemmatization than usual, probably because the Czech lemmas show only little ambiguity. However, we have noticed that very often the tail of the lemma is assigned incorrectly – this is why we often strip these in further processing, as they rarely convey any information relevant for Depfix or the analysis tools.

The lemma still might be changed in "Prepositional morphological case" (Section 6.2.3), which tries to change both the morphological tag and the lemma to find a correct analysis of a word form.

4.2.5 Fixing The Alignment Errors: Adding Missing Alignment Links

GIZA++ does not consider the similarity of word forms or word lemmas as an indicator of a higher alignment probability. However, in SMT outputs, we often encounter out-of-vocabulary items,[6] which are simply kept untranslated, or tokens that the translation system decided not to translate, probably believing them to be a named entity. If such a non-translation is indeed a correct translation, GIZA++ will probably align these tokens. However, if this is a mistake, which we would like to be able to detect and correct in Depfix, it is very probable that GIZA++ will be unable to align these tokens, as it itself encounters a target-side out-of-vocabulary item. This, however, makes many subsequent fixes impossible.

We try to fix that kind of errors by "Adding missing alignment links". Our fixing rule is fitted to the intersection alignment, which we use in Depfix, where each token is aligned to at most one other token. In the fix, we try to find pairs of matching words which do not have any alignment link, and add that link. Matching is done on forms and lemmas, lowercased with stripped diacritics. The fixing rule tries to match the form or lemma of one word to the form or lemma of another word; it is allowed to match the form of one word to the lemma of the other word. There are three levels of matching, tried one after another, going from exact matching to fuzzy matching:

1. identity match – the form/lemma of one word is identical to the form/lemma of the other word

[6] Out-of-vocabulary items are tokens that the SMT system is unable to translate, as it has not encountered them in its training data.

2. substring match – the form/lemma of one word is a substring of the form/lemma of the other word (not performed if one of the lemmas is less than 3 characters long)

3. prefix match – the first 4 characters of the form/lemma of one word are identical to the first 4 characters of the form/lemma of the other word (not performed if one of the lemmas is less than 4 characters long)

If a source word and a target word match, a new alignment link is created between them.

The later steps are less accurate, but the rise of recall is still much higher than the decrease of precision. However, it is important to first apply the more accurate matching and then proceed with the less accurate (nodes that already have an alignment link are not taken into account). This tries to ensure that the best possible alignment is found.

Example 4.6 shows the application of this correction on 'Alliot - Marie'. First, the surplus spaces are removed by "Tokenization projection" (Section 4.2.2), producing 'Alliot-Marie'. Next, after creating the word-alignment, the missing link between the identical 'Alliot-Marie' tokens is added. Later on, "Subject - past participle agreement" (Section 6.3.3) fixes the predicate 'poslal' ('sent') as, thanks to the added link, it is able to confirm that 'Alliot-Marie' is most probably the true subject.

Source:	... Michèle Alliot-Marie had sent a communication...
SMT output:	... Michèle *Alliot - Marie* **poslal** sdělení...
Gloss:	... Michèle *Alliot - Marie* $\textbf{sent}_{\textbf{masc}}$ a communication...
Depfix output:	... Michèle *Alliot-Marie* $\textbf{poslala}_{\textbf{fem}}$ sdělení...
Gloss:	... Michèle *Alliot-Marie* **sent** a communication...

Example 4.6

4.3 Translation fixes on morphological layer

This section describes fixes that change the word forms. The following fixes are performed on the m-layer, but they are not performed right after the analysis, since they do not influence the subsequent steps. Quite the opposite, it is beneficial to invoke these rules at the end of the whole Depfix pipeline, since changes made to the sentence by the other rules, especially when changing the word order or adding/removing words, can change the desired output of the rules.

4.3.1 Source-aware Truecasing

Moses has quite a low performance in correctly casing its outputs, see Section 3.3.5. We can therefore improve the output by simply projecting the casing from the source to the target, assuming that if the spellings of the aligned words are similar enough, their casing should be identical. See Example 4.7.

The rule finds pairs of aligned words that match in form or lemma, also allowing cases when the form of one of the words matches the lemma of the other one. The diacritics are ignored in the matching. The rule then matches the casing of the Czech word to the casing English word. Usually this means capitalizing the Czech word, but not necessarily – consider such cases as 'iPhone' or 'VMware'.

Source:	... the director of the best hotel in Pec, Karel Rada.
SMT output:	... ředitel nejlepší hotel v **peci**, Karel **rada**.
Gloss:	... the director of the best hotel in **the oven**, Karel **advice**.
Depfix output:	... ředitel nejlepšího hotelu v **Peci**, Karel **Rada**.
Gloss:	... the director of the best hotel in $\mathbf{Pec}_{\mathbf{town}}$, Karel $\mathbf{Rada}_{\mathbf{surname}}$.

Example 4.7

If the word is sentence-initial in the Czech or the English sentence, we do not attempt to fix it for obvious reasons.[7] If both the source and the target word are sentence-initial, we handle that by "Sentence-initial capitalization" (Section 4.3.3).

The rule uses a small manual list of words that should not be capitalized in Czech even if they are capitalized in English, which was quickly developed by observing errors on our development dataset.[8]

This rule also tries to detect the fake named entities error (Section 3.2.2), where a word was not translated because the SMT system believed the word to be a named entity when in fact it is not. This sub-rule is triggered when the form of the word is identical in source and in target, i.e. the word was not translated, but in source the word is lowercased whle in target in is capitalized, i.e. it is not a named entity in source but is formatted as one in target. It then tries to find a translation for the source lemma, using the simple translation tool described in Section 6.1.3, and if one is found, it substitutes the original target word with it.

See Example 4.8, where Moses probably mistook the word 'unleashed' for the title of a 2005 film. Depfix is able to detect that, and the translation tool provides us with a valid translation, although we are not able to generate the correct form of the verb.

[7] We are aware of cases where this is not enough – if the first token of the sentence is a punctuation mark, the sentence-initial word is not the first token. However, such cases in which the rule would incorrect the sentence are so rare in our development data that we decided not to handle these cases explicitly.

[8] The following words are not capitalized in Czech: 'Eur', 'Euro', 'Muslim', 'Islam', 'Protestant', 'Media', 'Internet', 'Hotel', 'Management', 'Manager', 'Premier', 'General', 'President', 'Lord', 'Sir'. Of course, such rules do not apply always, one can always find both positive and negative exceptions. For example. the phrase 'Finance Minister' should not be capitalized in Czech ('ministr financí'), while both of the individual words should be capitalized in Czech if they are capitalized in English. Some words, such as 'Hotel' in 'The Overlook Hotel', should be capitalized in Czech only if they are considered to be part of the name of the hotel, e.g. 'Overlook Hotel', but not of they are not, e.g. 'hotel Overlook'.

Source:	Having unleashed 238 world records since February 2008...
SMT output:	S **Unleashed** 238 světové rekordy od února 2008...
Gloss:	With **Unleashed**$_{\textbf{as the title of the film}}$ 238 of world records since February 2008...
Depfix output:	S **uvolnit** 238 světové rekordy od února 2008...
Gloss:	With **unleash**$_{\textbf{as the verb}}$ 238 of world records since February 2008...

Example 4.8

4.3.2 Vocalization of Prepositions

Prepositions 'k' ('to'), 's' ('with'), 'v' ('in') and 'z' ('from') are *vocalized*, i.e. changed to 'ke', 'se', 've', 'ze' where necessary. The vocalization rules in Czech are similar to 'a'/'an' distinction in English, motivated by the ease of pronunciation.

This fixing rule only uses a vocalization block that is already part of Treex.[9] The block is rule based, vocalizing the preposition if the following word starts with a character or a group of characters that require the preposition to be vocalized. For example, the preposition 's' is changed to 'se' if the following word starts by the characters 's' or 'z', by one of 21 bigrams, such as 'kt', 'vz', 'vš', 'mn', 'šk', 'že', or 'čt', by one of the trigrams 'čle', 'jmě', 'ple', 'šam', 'lst', 'prs', 'dvě', 'dře', or if the word is a number starting with the digit 7 ('sedm') or it is the number 17 ('sedmnáct').

The results of applying the rule can be seen in Example 4.9. (There is no difference in the meaning of the original and the fixed sentence.)

Source:	The work being done by experts from three institutions...
SMT output:	Práci odborníků **z** *třech* institucí...
Gloss:	Work by experts **from** *three* institutions...
Depfix output:	Práce odborníků **ze** *třech* institucí...
Gloss:	Work by experts **from** *three* institutions...

Example 4.9

4.3.3 Sentence-initial Capitalization

An error in sentence-initial capitalization is not very common in SMT outputs (although it can occur), but it is often created by Depfix fixes which delete words, as they can delete the first word of the sentence, but they do not capitalize the following word in such case and rely on this rule to do that for them.

This rule capitalizes the first word of the target sentence, unless the first word of the source sentence begins with a lowercase character, as this could indicate

[9] It is the `T2A::CS::VocalizePrepos` block.

that the first word is a special token, such as ‘www.nikde.eu’ or ‘iPhone’, which should not be capitalized.

Chapter 5

Parsing to Analytical Layer

In the previous chapter, we used several NLP tools to analyze the source and target sentences to the morphological layer, which contains analyses of the individual words, but does not provide us with any information about the relations between them.

Therefore, we proceed to the analytical layer, where the structure of the sentences is analyzed into *analytical trees*, which we briefly describe in Section 5.1.

Section 5.2 describes the parsers that we originally used to analyze the sentences to the analytical layer. However, as the performance of the standard parser on the Czech target sentences is too low, we explore possibilities to adapt the parser in the further sections.

5.1 Analytical Trees

The analytical layer, or a-layer, represents the sentences by labelled dependency syntax trees, called analytical trees or a-trees. We use the Treex (Popel and Žabokrtský, 2010) a-layer representation, which is based on (Hajič et al., 2006).

An a-tree is an oriented tree, with edges oriented from leaves to root. It consists of analytical nodes, or a-nodes, that have a one-to-one correspondence to the tokens of the sentence, and one (technical) root node. A pair of a-trees for the sentence 'Rudolf and David will go to school.' – 'Rudolf a David budou chodit do školy.' is shown in Figure 5.1.

Each edge in the a-tree is labelled with an analytical function, which is the function that the dependent (the child node) performs on its governor (the parent node) – such as being its subject (the `Sb` analytical function) or expressing its attribute (the `Atr` analytical function). The analytical functions are actually stored as attributes of the dependent nodes and sometimes even have little to do with the parent node, such as an a-node that is a head of a coordination and is labelled with the `Coord` analytical function. Some of the analytical functions that we often use in Depfix are listed in Table 5.1; the full list of analytical functions can be found in the PDT 2.0 manual (Hajič et al., 2006).

While some properties of dependency syntax trees are universal or nearly universal, such as the subject node being a descendant of the predicate node or an adjectival attribute being a child of the noun it modifies, other conventions are not consistent across different dependency grammars, as they are motivated by different theories or different design decisions. We therefore list several properties

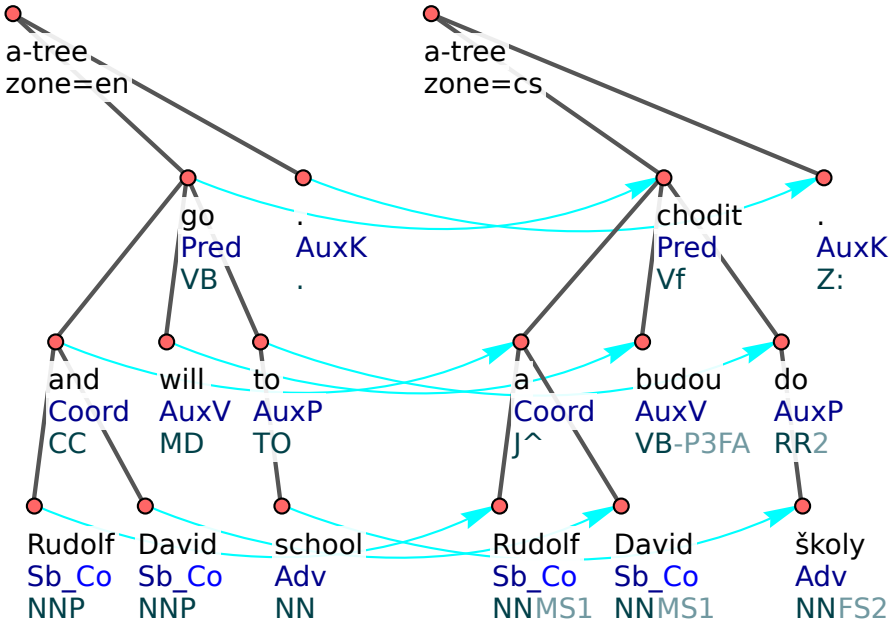


Figure 5.1: A pair of a-trees for the sentence 'Rudolf and David will go to school.' – 'Rudolf a David budou chodit do školy.'

Label	Meaning
`Pred`	predicate
`Sb`	subject
`Obj`	object
`Atr`	attribute
`Adv`	adverbial
`AuxP`	preposition
`AuxV`	auxiliary verb
`AuxK`	terminal punctuation
`Coord`	root of coordination

Table 5.1: Some analytical functions. Adapted from (Hajič et al., 2006).

Relation	Topological	Effective
Parent of 'David'	'and'	'go'
Children of 'go'	'and', 'will', 'to'	'Rudolf', 'David', 'will', 'to'

Table 5.2: An example of the difference between topological relations and effective relations.

of the a-nodes in Treex, especially those that are important in the design of Depfix rules. All of the properties can be observed in Figure 5.1.

- predicate verbs govern clauses
- prepositions govern prepositional constructions
- conjunctions govern coordinations; the coordinated nodes are children of the conjunction and are marked by the `is_member` flag, which is, for historical reasons, visualized as a `_Co` suffix on the analytical function label[1]
- punctuation marks are also represented by a-nodes; the sentence-final stop is typically a child of the (technical) root node
- compound verb forms are represented by a subtree with the full verb in the root, and the verb arguments (e.g. subjects and objects) and auxiliary words (e.g. auxiliary verbs and negation) as its children
- the notion of *effective relations* is used, which skip coordination heads; see Table 5.2 for an example showing the difference between topological and effective relations on the English sentence shown in Figure 5.1

Analytical trees allow us to explore the structure of the sentence. The morphological agreement in Czech, which is one of the major phenomena of the sentence that we want to check and correct in Depfix, is typically required along a dependency edge; or, to be more precise, between a child and its effective parent, such as an adjective and its parenting noun or a subject and its parenting predicate. This means that many Depfix corrections are local to one edge (a child and its effective parent); only sometimes we have to explore more distant nodes in the tree.

The topological relations are of much less use for us than the effective relations. Therefore, by saying "a parent" or "a child", we will typically mean "an effective parent" or "an effective child".

5.2 Analysis to Analytical Layer

To analyze source English sentences, we use the Maximum spanning tree (MST) parser (McDonald et al., 2005), in its original implementation,[2] which is included in the Treex framework. As we found its performance on the source sentences sufficient for our needs, we did not attempt to adapt it in any way.

[1] The meaning of the `is_member` flag is "is a member of a coordination or apposition". An apposition would be visualized by an `_Ap` suffix.

[2] `http://sourceforge.net/projects/mstparser`

The first version of Depfix used the original MST Parser for analysis of Czech sentences as well, in its adaptation by Novák and Žabokrtský (2007), which we will further refer to as the MSTA parser. The MSTA parser uses first-order and second-order features for parsing, and a k-best MIRA with an update performed by quadratic analysis.

However, when analyzing the target sentences, we observed the a-trees produced by the parser to be highly erroneous, and we therefore decided to try to create an adapted version of the parser which would be able to handle the target data better. The parser that we now use in Depfix to parse the target sentences is a reimplementation of the MST Parser, described in Section 5.3, which we further adapted in several ways for our specific task: by introducing SMT-like errors in its training data (Section 5.4), by incorporating information from the source sentences (Section 5.5, and by adding large-scale information (Section 5.6).

5.3 Reimplementing the Maximum Spanning Tree Parser

We reimplemented the Maximum Spanning Tree (MST) parser (McDonald et al., 2005) This allows us to easily modify any part of the parser – not only the feature set, but also e.g. the loss function.

The MST Parser uses the Margin Infused Relaxed Algorithm (MIRA), described in Crammer and Singer (2003). MIRA is an online learning algorithm for large-margin multiclass classification, successfully used in McDonald et al. (2005) for dependency parsing and suggested to be used for dependency relations labelling in a second stage labeller. MIRA makes it possible to use a large number of features, both first- and higher-order. We used the *single-best MIRA* variant, with a closed form update instead of the quadratic optimalization.

We briefly describe our implementation of the MST Parser in Section 5.3.1.

Assigning a correct analytical function label to each dependency relation is an important part of parsing. Some parsers perform joint parsing and labelling, producing a labelled dependency tree as their output. Others produce only an unlabelled tree, requiring a standalone labeller to assign the labels in a second step. The MST Parser is an unlabelled parser. Therefore, we also implemented a second-stage MIRA-based labeller to be used for labelling the dependency trees, which we describe in Section 5.3.2.

A very important part of a parser is its feature set. We describe the base features we use in Section 5.3.3.

We implemented the parser in Perl and included it into the Treex framework.

5.3.1 The Unlabelled Parser

We use the Maximum Spanning Tree Parser (MST Parser), which is an unlabelled dependency parser by McDonald et al. (2005). To parse a sentence, the MST Parser takes the following approach:

1. construct a complete directed graph on all of the nodes (words)

2. assign a score to each edge, using a trained model

3. find the maximum spanning tree in the graph

4. return that spanning tree as the parse tree of the sentence

Because Czech is a non-projective language, we only implemented its non-projective version (McDonald et al., 2005). The non-projective version uses an $O(n^2)$ parsing algorithm based on the Chu-Liu-Edmonds maximum spanning tree algorithm (Chu and Liu, 1965; Edmonds, 1967), instead of the $O(n^3)$ projective parsing algorithm of Eisner (1996). For simplicity, we only implemented the first-order parser.

Based on evaluations of the performance of the parser, we use only 3 iterations of the training process instead of 10.

5.3.2 Second Stage Labelling

As the MST Parser is an unlabelled parser, the analytical functions have to be assigned by a second stage labeller. We implemented a MIRA-based labeller following McDonald (2006), Gimpel and Cohen (2007) and Rosenfeld et al. (2006). We already presented the labeller in (Rosa and Mareček, 2012).

A *labelling* $\mathcal{L}$ of a dependency tree is a mapping $\mathcal{L} : E \to L$, which assigns a label $l \in L$ to each edge $e \in E$ of the dependency tree. The general approach that we follow to find the best labelling is using edge-based factorization, i.e. to assign a score to each possible pair $[e, l]$, and to try to find a labelling with a maximum overall score.

Following McDonald et al. (2006), we treat the dependency relations labelling as a *sequence labelling problem*, with a sequence defined as a sequence of adjacent edges, i.e. edges sharing the same parent node. Starting at the root node of the tree, we label all edges going from the current node to its children from left to right, and then continue in the same way on lower levels of the tree. This implies that at each step we have already processed all ancestor edges and left sibling edges. We utilize this fact in designing the feature set.

MIRA assigns a score for each label that can be assigned to a dependency relation. Feature-based factorization is used, thus the score of a label l to be assigned to an edge e which has the features F is computed as follows:

$$score(l, e) = \sum_{f \in F} score(l, f) \tag{5.1}$$

where $score(l, f)$, also called the *weight* of the feature (l, f), is computed by MIRA, trying to minimize the classification error on the training data, iterating over the whole dataset several times. The final scores are then averaged to avoid overtraining.

5.3.3 The Basic Features

We present here the features that we use for both the unlabelled parser and the second-stage labeller as the base features in a monolingual setting. We construct the feature set by combining features suggested in McDonald et al. (2005), McDonald et al. (2006), McDonald and Pereira (2006) and Carreras (2007) and tuning it to maximize performance on Czech data.

In this section, we only describe the individual features used, which are very similar to the features proposed by the aforementioned authors. To construct the full feature set, we further join the features in various ways, which is realized by concatenating the values of the individual features. The final feature set was developed by starting off with the combinations proposed by other authors, and then iteratively performing slight changes of the feature set and observing its effect on the performance, until we reached a performance level which we could not further improve. We list the final feature set in Attachment C,

For the first-order features, which are used in both the parser and the labeller, we can only make use of the information that is stored in the m-nodes (see Section 4.1), i.e.:

- `form` – the word form of the node
- `lemma` – the lemma of the node
- `coarse_tag` – the coarse morphological tag of the node
- `ord` – the order of the node in the sentence

Following Collins et al. (1999), we decided to use the coarse morphological tags instead of the full morphological tags for data sparsness reasons. A coarse morphological tag is a simplified morphological tag, capturing only the most important information. Its structure is as follows:

1. the **part of speech**
2. the **morphological case** if the part-of-speech expresses one (nouns, adjectives, pronouns, some numerals)
   or the **detailed part of speech** if it does not (verbs, adverbs...)

We performed a set of experiments, trying to utilize some more information from the full morphological tag than that included in the coarse morphological tag, but none of the experiments lead to an increase of performance. We therefore only use coarse morphological tags for the parsing.

In the second-stage labelling, we can also make use of the knowledge of the whole parse tree, which is the output of the first-stage parser. This allows us to easily introduce additional non-local and higher-order features into the labeller feature set.

We denote a field of the (potential) parent node of the edge by using uppercase, e.g. `LEMMA` for the lemma of the parent node; the child node is indicated by using lowercase.

First-order features

Our set of first-order features is based on features described in McDonald et al. (2005),[3] which were primarily designed for unlabelled parsing but proved to be useful for labelling as well. They consist of the basic fields available on the input, and of several context features, providing information about nearby words:

[3]Different to McDonald et al. (2005), we use lemmas instead of 5-gram prefixes.

- `COARSE_TAG` / `coarse_tag` – the parent/child coarse morphological tag
- `FORM` / `form` – the parent/child word form
- `LEMMA` / `lemma` – the parent/child lemma
- `PRECEDING(coarse_tag)` / `preceding(coarse_tag)` – the coarse morphological tag of the word immediately preceding the parent/child node
- `FOLLOWING(coarse_tag)` / `following(coarse_tag)` – the coarse morphological tag of the word immediately following the parent/child node
- `between(coarse_tag)` – a bag of coarse morphological tags of nodes positioned between the parent node and the child node
- `attdir()` – the direction of attachment of the child to the parent (left or right)
- `distance()` – 1-based signed bucketed distance of the parent node and the child node in the sentence (order of parent minus order of child), using buckets 1, 2, 3, 4, 5 and 11; each value gets bucketed into its closest bucket with an equal or lower absolute value (e.g. 2 gets to 2, 7 gets to 5, -20 gets to -11)

Non-local features

Based on non-local features described in McDonald et al. (2006), we extend the second-stage labeller feature set with the following four features:

- `CHILDNO()` / `childno()` – number of child nodes of the parent/child node
- `isfirstchild()` / `islastchild()` – a boolean indicating whether the child node is the leftmost/rightmost child node of its parent node

To compute these features, the whole parse tree must be known. Therefore, these features cannot be included in the feature set of the first-stage parser.

Higher-order features

Higher order features are based on multiple edges. These can be sibling features as described in McDonald et al. (2006), parent-child-grandchild features as described in Carreras (2007), or other variations and conjunctions of these concepts.

The possibility to use some of the features depends on the order of assigning labels to edges in the dependency tree. As stated in Section 5.3.2, we label the nodes in a top-down left-to-right order, which enables us to use the information about the labels assigned to sibling edges which are on the left from the current edge, and to edges between ancestor nodes, such as the parent or grandparent.

We use the following set of higher-order features:

- `LABEL()` – label assigned to the edge between the parent and the grandparent of the child node

- `l.label()` – label assigned to the left sibling edge, i.e. the edge between the parent and the left sibling of the child node
- `r.coarse_tag` – coarse tag of the right sibling of the child node[4]
- `G.coarse_tag` – coarse tag of the grandparent of the child node
- `G.label()` – label assigned to the edge between the grandparent and the great-grandparent of the child node
- `G.attdir()` – whether the grandparent node precedes or follows the child node in the sentence

From all of the higher-order features, the grandparent features have the biggest influence on accuracy. Their inclusion in the feature set led to an improvement of accuracy by 2% LAS (labelled attachment score), whereas most of the other features contribute with less than 0.5% LAS. They are especially useful for e.g. prepositional and coordination structures.

The performance of the parser with the final feature set is evaluated in Section 8.4, showing that it reaches the performance of MSTA parser.

While adding higher-order features into the first-stage parser is possible, as described in (McDonald and Pereira, 2006), it is not straight-forward, and necessitates the introduction of approximation techniques for the problem to remain tractable, as second-order non-projective MST parsing was shown to be NP-hard. We therefore decided not to include higher-order features into the first-stage parser.

5.4 Worsening the Training Data

Addressing the issue of great differences between the gold standard parser training data and the actual analysis input (target), we introduced artificial inconsistencies into the training treebanks, in order to make the parsers more robust in the face of grammar errors made by SMT systems. We have concentrated solely on modelling incorrect word flection, i.e. the dependency trees retained their original correct structures and word lemmas remained fixed, but the individual inflected word forms have been modified according to an error model trained on real SMT output. We simulate thus, with respect to morphology, a treebank of parsed MT output sentences.

Section 5.4.1 discusses previous research related to that approach. In Section 5.4.2 we describe the steps we take to prepare the worsened parser training data. Section 5.4.3 contains a description of our monolingual greedy alignment tool which is needed during the process to map SMT output to reference translations.

All of the tools that perform the training data worsening were implemented in Treex by David Mareček and Martin Popel. We only incorporated them into Depfix and evaluated their performance. We already described this approach in (Rosa et al., 2012a).

[4]Note that this is different from the `following(coarse_tag)` feature, as the node that follows the child node does not have to be its sibling; it can be its child, its grandchild, its parent...

5.4.1 Related work

To the best of our knowledge, the only work presenting a similar approach to ours is by Foster et al. (2008), who introduce various kinds of artificial errors into the training data to make the final parser less sensitive to grammar errors. However, their approach concentrates on mistakes made by humans (such as misspellings, word repetition or omission etc.) and the error models used are hand-crafted. Our work focuses on morphology errors often encountered in SMT output and introduces statistical error modelling.

5.4.2 Creating the Worsened Parser Training Data

The whole process of treebank worsening consists of five steps:

1. We translated the English side of PCEDT[5] (Hajič et al., 2012) to Czech using SMT (we chose the Moses system (Koehn et al., 2007) for our experiments) and tagged the resulting translations using the Morče tagger (Spoustová et al., 2007).

2. We aligned the Czech side of PCEDT, now serving as a reference translation, to the SMT output using our Monolingual Greedy Aligner (see Section 5.4.3).

3. Collecting the counts of individual errors, we estimated the Maximum Likelihood probabilities of changing a correct morphological tag (of a word from the reference) into a possibly incorrect morphological tag of the aligned word (from the SMT output).

4. The tags on the Czech side of PCEDT were randomly sampled according to the estimated "morphological tag error model". In those positions where morphological tags were changed, new word forms were generated using the Czech morphological generator by Hajič (2004).[6]

We use the resulting "worsened" treebank to train our parser described in Section 5.3.1.

5.4.3 The Monolingual Greedy Aligner

Our monolingual alignment tool, used in treebank worsening to tie reference translations to MT output (see Section 5.4.2), scores all possible alignment links and then greedily chooses the currently highest scoring one, creating the respective alignment link from word A (in the reference) to word B (in the target) and deleting all scores of links from A or to B, so that one-to-one alignments are

[5]This approach is not conditioned by availability of parallel treebanks. Alternatively, we might translate any text for which reference translations are at hand. The model learned in the third step would then be applied (in the fourth step) to a different text for which parse trees are available. We plan to evaluate such setup in future.

[6]According to the "morphological tag error model", about 20% of morphological tags were changed. In 4% of cases, no word form existed for the new morphological tag and thus it was not changed.

enforced. The process is terminated when no links with a score higher than a given threshold are available; some words may thus remain unaligned.

The score is computed as a linear combination of the following four features:

- word form (or lemma if available) similarity based on Jaro-Winkler distance (Winkler, 1990),
- morphological tag similarity,
- similarity of the relative position in the sentence,
- and an indication whether the word following (or preceding) A was already aligned to the word following (or preceding) B.

Unlike bilingual word aligners, this tool needs no training except for setting weights of the four features and the threshold.[7]

5.4.4 Evaluation

An extrinsic evaluation confirmed that worsening the training data makes the parser more robust to errors contained in SMT outputs, as the effect of using it in Depfix has a consistently positive effect on Depfix performance. The results of the evaluation are detailed in Section 8.4.2.

5.5 Adding Parallel Information

An advantage of parsing of SMT outputs over general dependency parsing is that one can also make use of the source – English sentences in our case. Moreover, although the target sentences are often in many ways ungrammatical, the source sentences are usually grammatical and therefore easier to process; in our case, especially to tag and parse.

To make use of this advantage, we devised a simple way of providing the additional information to the parser:

1. we parse the source sentence by a monolingual parser
2. we compute features on the parsed source sentence
3. we add the features into the feature set of the target parser
4. we parse the target sentence by the enriched parser

We discuss work related to parsing of bilingual texts in Section 5.5.1, and then describe the set of parallel features we use in Section 5.5.2. We also describe our experiments with manually boosting the weights of the parallel features in Section 5.5.3.

[7]The threshold and weights were set manually using just ten sentence pairs. The resulting alignment quality was found sufficient, so no additional weights tuning was performed.

5.5.1 Related Work

Our approach to parsing with parallel features is similar to various works which seek to improve the parsing accuracy on parallel texts ("bitexts") by using information from both languages. Huang et al. (2009) employ "bilingual constraints" in shift-reduce parsing to disambiguate difficult syntactic constructions and resolve shift-reduce conflicts. Chen et al. (2010) use similar subtree constraints to improve parser accuracy in a dependency scenario. Chen et al. (2011) then improve the method by obtaining a training parallel treebank via SMT. In recent work, Haulrich (2012) experiments with a setup very similar to ours: adding alignment-projected features to an originally monolingual parser.

However, the main aim of all these works is to improve the parsing accuracy on correct parallel texts, i.e. human-translated. This paper applies similar methods, but with a different objective in mind – increasing the ability of the parser to process ungrammatical target sentences and, ultimately, improve rule-based SMT post-editing.

Xiong et al. (2010) use SMT parsing in translation quality assessment, providing syntactic features to a classifier detecting erroneous words in SMT output, yet they do not concentrate on improving parsing accuracy – they employ a link grammar parser, which is robust, but not tuned specifically to process ungrammatical input.

There is also another related direction of research in parsing of parallel texts, which is targeted on parsing under-resourced languages, e.g. the works by Hwa et al. (2005), Zeman and Resnik (2008), and McDonald et al. (2011). They address the fact that parsers for the language of interest are of low quality or even non-existent, whereas there are high-quality parsers for the other language. They exploit common properties of both languages and de-lexicalization. Zhao et al. (2009) uses information from word-by-word translated treebank to obtain additional training data and boost parser accuracy.

This is different from our situation, as there exist high performance parsers for Czech (Buchholz and Marsi, 2006; Nivre et al., 2007; Hajič et al., 2009). Boosting accuracy on correct sentences is not our primary goal and we do not intend to *replace* the Czech parser by an English parser; instead, we aim to increase the robustness of an already *existing* Czech parser by adding knowledge from the corresponding English source, parsed by an English parser.

Other works in bilingual parsing aim to parse the parallel sentences directly using a grammar formalism fit for this purpose, such as Inversion Transduction Grammars (ITG) (Wu, 1997). Burkett et al. (2010) further include ITG parsing with word-alignment in a joint scenario. We concentrate here on using dependency parsers because of tools and training data availability for the examined language pair.

5.5.2 Parallel Features

We devised three parallel features, computed for the parent and child node of an edge, which make use of the source a-nodes aligned to the parent and child node. (It is necessary that the source sentence is already analyzed up to a-layer, and the intersection word-alignment has been computed.) The parallel features are

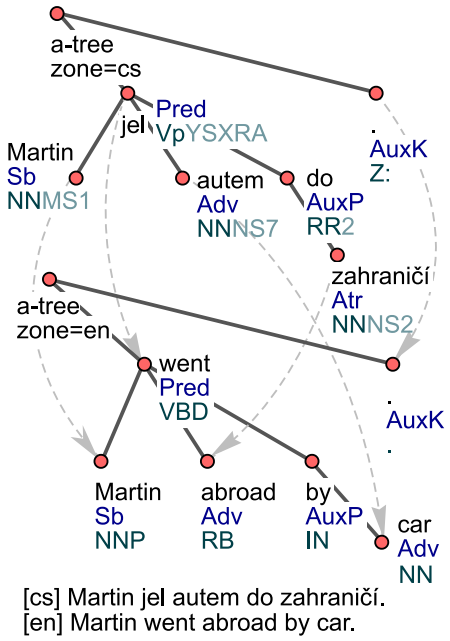


Figure 5.2: Example sentence for parallel features illustration (see Table 5.3).

then conjoined with the monolingual `coarse_tag` and `lemma` features in various ways; the full feature set is listed in Attachment C.

An example of a pair of parallel sentences is given in Figure 5.2 with the corresponding values of parallel features for several edges in Table 5.3.

Aligned tag feature

The value of the `aligned_tag` feature is the POS tag of the English node aligned to the Czech node, or an empty string if there is no aligned English node.

Aligned analytical function feature

The value of the `aligned_afun` feature is the analytical function of the English node aligned to the Czech node, or an empty string if there is no aligned English node.

Aligned edge existence feature

The `aligned_edge` feature tries to find the English nodes aligned to both the Czech parent node and the Czech child node. The value of the feature is:

- **1** if the aligned nodes are found and have the desired relation, i.e. the source node aligned to the potential target parent node is a parent node of the source node aligned to the target child node
- **0** if the aligned nodes are found but do not have the desired relation
- **-1** if at least one of the aligned nodes is not found, i.e. the word alignment left at least one member of the pair of source nodes unaligned

Feature	Feature value on	
	parent node	child node
word form	jel	Martin
aligned_tag	VBD	NNP
aligned_afun	Pred	Sb
aligned_edge	1	
word form	jel	autem
aligned_tag	VBD	NN
aligned_afun	Pred	Adv
aligned_edge	0	
word form	do	zahraničí
aligned_tag	”	RB
aligned_afun	”	Adv
aligned_edge	-1	
word form	#root#	.
aligned_tag	#root#	.
aligned_afun	AuxS	AuxK
aligned_edge	1	

Table 5.3: Parallel features for several edges in Figure 5.2.

The extrinsic evaluation of the modified parser, detailed in Section 8.4.2, shows that the addition of information from the aligned source sentence leads to a relatively large increase in performance of Depfix.

5.5.3 Manually Boosting Feature Weights

By manual inspection of the parsing errors on our data, we found that in many cases, the error would be prevented if the tree structure was simply projected from source to target. Even after including the parallel features, this phenomenon did not disappear, although it was considerably diminished. We therefore decided to try to modify the model manually, artificially boosting weights of some of the parallel features – namely the `aligned_edge` feature.

We first inspected the weights of the `aligned_edge` feature in a trained parser model. We found the weights of the feature to be as follows:

- -0.5683 for `aligned_edge:1`
- -0.6714 for `aligned_edge:-1`
- -0.8338 for `aligned_edge:0`

Please note that the absolute values of the weights of the features are not important, as one of these features is always present; what is important are the relative differences in their values. The `0` value has a much lower weight than the other two features; however, the weight of the `-1` value is closer to the weight of `1`.

We therefore tried to manually increase the weight of the `aligned_edge:1` feature by steps of 0.05, and automatically evaluated such modified models in

Depfix using NIST. The scores showed and approximate tendency of first rising, as the increased feature weight was helping the parser to produce a better parse tree, and then falling, as the weight was getting too high and the parser started to project the structure of the source parse tree onto the target sentence even in cases when it was not appropriate.

We decide to use a -0.25 feature weight in our final model, as the NIST score seemed to peek around that value. This means that we increased the feature weight by about 0.30, which is above most of the individual feature weights in the model. The evaluation of the final model on other datasets in Section 8.4.2 shows our approach to be promising, leading to a moderate improvement of Depfix performance.

It seems that the set of feature weights of the discriminative model is not a black box, and manual changing of the feature weights, when motivated by a good intuition, can be a viable way for modifying the models – as in our case, where the data that we need to process with the parser are substantially different from the data it is trained on, and we need to account for that.

However, our approach is very simple and coarse. The final model contains more than 9 million feature weights, and more could be added easily as MIRA is known for being able to handle really large feature sets. We only modified one of all of these feature weights and got a slight but consistent improvement; much more could probably be gained be trying to change other feature weights as well.

5.6 Adding Large-scale Information

As another attempt to increase the accuracy of the parser, we try to exploit large-scale parsed data to provide additional lexical features to the parser.

We performed several simple observations when tuning the feature set of our parser, and when comparing PCEDT 2.0 (Hajič et al., 2012), a (mostly) manually created parallel treebank of approximately 1 million sentences which we use as the training data for our parser, with CzEng 1.0 (Bojar et al., 2012b), an automatically created parallel treebank of approximately 15 million sentences.

The research presented in this section is motivated by the following observations:

- the lexical features (`form`, `lemma`) are important for the parser, as its performance drops considerably when they are removed; they are probably even more important in our setup, as the coarse morphological tags are not as reliable

- the Czech side of PCEDT 2.0 contains roughly 80,000 different word forms (40,000 different lemmas)

- the Czech side of CzEng 1.0 contains roughly 1,700,000 different word forms (400,000 different lemmas)

These observations indicate that the performance of the parser on words that are not contained in its training data is probably significantly lower. They further lead us to believe that utilizing CzEng or other large-scale dataset to provide

additional lexical information for the parser might bring an improvement of its performance.

Our approach is, simply put, to compute the Pointwise mutual information (PMI) for each pair of words that we encounter in CzEng having a parent-child relation, and to add the value of the PMI as a feature to the parser.

It should be noted here that the Czech side of CzEng was automatically analyzed by an instance of the MST Parser, i.e. we are trying to improve the performance of the parser by utilizing its outputs on a dataset different from its training set.

5.6.1 Pointwise Mutual Information of Parents and Children

The Pointwise mutual information of a pair of outcomes, p and c, is computed by the formula (5.2):

$$PMI(p,c) = log\frac{p([p,c])}{p([p,*]) \cdot p([*,c])} \tag{5.2}$$

For our needs, we define:

- $p([p,*])$ as the probability of an edge to have the word p as its parent
- $p([*,c])$ as the probability of an edge to have the word c as its child
- $p([p,c])$ as the probability of an edge to have both the word p as its parent and the word c as its child

All of the probabilities can be easily estimated as frequencies of the outcomes in the data. Moreover, as all of the frequencies have a common denominator – the total number of edges in the data, $c([*,*])$ – we can leave out the denominator and use only the counts of the outcomes instead of frequencies; we denote such modification of PMI as PMI'. Thus, we can estimate the PMI' by the formula (5.3), where $c()$ stands for the count:

$$PMI'(p,c) = log\frac{c([p,c])}{c([p,*]) \cdot c([*,c])} = PMI(p,c) - log(c([*,*])) \tag{5.3}$$

Please note that the probabilities used to compute the PMI could be defined differently, most probably leading to different results. However, we did not explore any other possible definitions of the probabilities.

In the formulas, we refer to p and c as words. However, a word can be represented in many ways. We experimented with representing the words both by their forms and by their lemmas. The performance of both of these approaches was very similar; we therefore decided to use the lemmas, following the Occam's razor principle.

5.6.2 Definition of the New Feature

Computing the PMI on the Czech side of CzEng yields over 6 million distinct values when using double-precision floats. Because the MST Parser only supports discrete-valued features, a transformation of the space of the computed PMI values has to be done. We use the bucketing approach, which was already successfully used for the `distance()` feature.

The distribution of the values of PMI seems to approach the normal distribution.[8] We therefore tried to find a set of buckets that would lead to an approximately binomial distribution of the values. We evaluated several such bucket sets, observing little difference in the performance; still, the best performing set of buckets seems to be the following one: -7, -10, -12, -13, -14, -15, -16, -17, -18, -19, -20, -21. Each value is bucketed into the nearest equal or lower bucket, except for the values lower than the lowest bucket, which get bucketed into the lowest bucket.

Thus, we can define a new feature, `pmibucketed(lemma)`, which returns the bucketed value of the PMI for the pair of the parent and child lemmas. If the PMI is unknown, the feature has a '`?`' value.

5.6.3 Cutting Off Low Counts

Initially, we would not include word pairs with a low frequency into the model, typically if they occurred only once in the data. To the best of our knowledge, it is a very common technique, used to make the data cleaner and their distribution smoother. It uses the assumption that the "low counts" are typically errors or some extremely rare or obscure cases, and that removing them cannot hurt the performance much – on the contrary, the performance is expected to be higher if this approach is used.

However, in our experiments, cutting off infrequent values, such as word pairs seen only once or e.g. seen less than 5 times, made the results worse. This is probably because by cutting off the low counts, we lose the very information that we are trying to get by this approach – information about words so infrequent that they do not occur frequently enough in the parser training data.

This is surprising, because we know that the data are noisy – we expected that the low counts would represent mainly errors, and when we looked at a small sample of the data, our observations confirmed our expectations. However, it seems that either the data are not as noisy as we originally thought, or, more probably, that adding a lot of erroneous word pairs hurts the performance much less than omitting a few correct word pairs.

Therefore, we decided not cut off any values.

5.6.4 Conclusion and Future Work

In the extrinsic evaluation of the described setup, detailed in Section 8.4.2, using the adapted feature set resulted in an increase of Depfix performance only on two out of four evaluation datasets. We are therefore unsure whether our approach can be thought of as promising; however, the average difference in Depfix performance

[8] This was estimated from observing the histogram of the values; no tests were performed.

is positive, so we believe that this approach does improve the performance of the parser, although most probably only very slightly.

There are many aspects of our approach that require further research. Probably the most important one is the unconfirmed assumption that the performance of the MST Parser can be increased by exploiting the outputs of a previously trained instance of the parser to provide features in training a new instance of the parser. Our choice of the way to compute the PMI, without exploring other possible ways, also requires a reconsideration, and probably either a theoretical analysis or a set of experiments should be employed to explore the set of possible ways to compute the PMI, choosing the most promising one or ones. And finally, manual extrinsic evaluation of the modified parser should be done, as the automatic evaluation is probably too coarse-grained to reliably identify the effect of our approach on the parsing quality.

The approach we took is also by no means the only way of utilizing the observations made. For example, another reasonable way of overcoming the gap between the number of existing words and the number of distinct words present in the training data could be the employment of word classes (Brown et al., 1992). Much care would have to be taken to appropriately choose both the size of the set of the word classes to be used, and the way to devise it. However, it would then be easy to either substitute the lexical features by the word-class features, or to include the word-class features as additional features in the feature set of the parser. This might make the parser more robust to previously unseen words while at the same time not harming its peformance on words that were frequent enough in the training data.

5.7 Modifying the Loss Function

In Depfix processing, the correct identification of some relations by the parser is more important than of other relations – e.g., in "Noun - adjective agreement" (Section 6.3.6), we need the noun-adjective edges to be correct, while the correctness of other edges is largely irrelevant.

The approach that we took to incorporate this knowledge into the parser was by trying to modify the loss function of MIRA, which seemed to be rather natural. The MST parser uses a very simple loss function by default, assigning a loss equal to the number of incorrectly assigned parents. It is therefore straightforward to adjust the loss function to our needs.

We tried several modifications of the loss function, such as:

- loss = -10 if the child is an adjective and its correct parent is a noun[9]
- loss = -10 if the child is an adjective
- loss = -10 if the correct parent is a conjunction

Although the idea seems straightforward, we have not observed any significant improvement for any of the modified loss functions. This might both mean that our approach is wrong, or that the our approach is correct but we have been unable to find a good alternative loss function.

[9]This leads to nearly every adjective becoming a child of a noun even if it should not.

Chapter 6

Fixes on Analytical Layer

Depfix uses a set of hand-written fix rules that operate on the a-layer, i.e. on analytical trees. The fix rules try to correct various errors that are common in SMT outputs, as analyzed in Chapter 3.

Each rule takes a child-parent pair as its input (technically, each of the rules is invoked for each of the a-nodes in the a-tree of a target sentence, together with its parent node). We use the term *parent* to denote the effective parent (as defined in Section 5.1), since we do not use the topological parent in any of the rules. Naturally, each rule has access to the whole a-tree of both the target and the source tree, including the alignment of the a-nodes; however, most of the fixes are highly local, accessing at most parents, children and siblings of the two a-nodes on their input and their English source counterpart a-nodes.

A fix rule first checks a set of conditions to decide whether it applies to the given child-parent pair, often inspecting the morphological and syntactical categories of the two given nodes and other relevant nodes, and other information. If the conditions are met and an error is found, the rule attempts to correct it. This usually involves changing morphological categories of one of the nodes (such as morphological number, morphological gender and morphological case) and regenerating the corresponding word form if necessary, using the morphological generator described in Section 6.1.2. More rarely, the fix is accompanied by deleting superfluous particles or auxiliary words, changing the target a-tree structure, changing the word order, or even changing the lemma (i.e. making a different lexical choice).

The rules can be classified into several categories:

- Analysis Fixing Rules, which correct the tagger and parser errors (Section 6.2)
- Agreement Fixing Rules, which enforce agreement (Section 6.3)
- Translation Fixing Rules, which directly correct translations (Section 6.4)

Further sections provide descriptions of the rules, together with examples from our development data. In the examples, the word that is being fixed is shown in **bold**, while other important words, such as the word that determines the correct values of the morphological categories of the fixed word, are shown in *italic*.

To keep the descriptions concise, we use the term *aligned child* for the English node aligned to the Czech child node; similar terms, such as *aligned parent*, have similar meanings.

The order in which the rules are applied is discussed in Section 6.5.

6.1 Common Parts of A-layer Fixes

The three categories of a-layer fixes are very different and are therefore described in separate sections. This section describes their common parts.

6.1.1 Named Entities

We have observed that fixing named entities is usually hard, mainly because the analysis of named entities by the NLP tools (see Section 4.2.1) is highly erroneous. It seems that the tools have little or no abilities to adapt to unknown and new named entities, often yielding an incorrect lemma and/or morphological tag.

For this reason, many of the rules do not fix nodes that appear to be a named entity. However, there is currently no reliable named entity recognizer for Czech included in Treex; there is one only for English. Therefore, we have to resort to one of the following two solutions if we need to detect Czech named entities and avoid fixing them:

- projecting the named entity markers from English – has a high precision but a lower recall, due to alignment errors, tokenization differences and other Czech-English differences
- using a very simple guesser, which assumes each non-lower case word[1] to be a possible named entity – has a low precision but a very high recall for obvious reasons

We select the former or the latter approach in each rule where the proportion of named entity-related errors is too high, based on performance observed on development data. Naturally, projecting the named entity markers from English allows us to make more corrections, and we prefer to choose it if possible. However, if this still leads to too many incorrections, we resort to the simple guesser, which avoids nearly all named entity-related incorrections.

6.1.2 Morphological Generator

A morphological generator is a tool inverse to the tagger: based on a lemma and a morphological tag, it generates the corresponding word form. The generator is used after performing a fix which requires regeneration of the word form according to a new morphological tag (or, less frequently, a new lemma). This occurs in all agreement fixes (Section 6.3), many translation fixes (Section 6.4) and most t-layer fixes (Section 7.3, Section 7.4).

We use the morphological generator which is part of Treex, and is built upon the morphology of Hajič (2004).

[1] A non-lower case word is not necessarily a capitalized or uppercase word – consider the named entity 'iPhone'.

Fixing word-forms generation errors

Generating new word-forms is generally very reliable, rarely producing an incorrect word-form. However, its recall is lower than we had expected.

There are only two cases when we fix this kind of errors:

- in "Translation of possessive nouns" (Section 6.4.5), where we use the genitive morphological case if the generator is unable to generate the possessive noun form
- in "Negation translation" (Section 7.3.1), where we prefix the word form with the negative prefix 'ne' if the generator is unable to generate a negated word form.

6.1.3 A Simple Static Translator

In Depfix, we generally do not try to fix lexical translation errors. However, there are a few cases where we can be nearly sure that a mistranslation has happened, either by omitting a word in the output or by making an inappropriate lexical choice. In these situations, we try to correct that error, knowing that the translation that Depfix provides will most probably not be perfect, but hoping that it will still be better than the original mistranslation produced by the SMT system.

Here, we benefit from the fact that the TectoMT translation system (Popel and Žabokrtský, 2010) is implemented in Treex, i.e. in the same framework as Depfix, and we "borrow" one of its translation models for this task. We use the so-called Static model,[2] a simple model for lemma-to-lemma translations, which simply estimates the probabilities of the translations from their frequencies in the data.

We pass a source lemma of the word we want to translate to the model, and it provides us with a list of possible translations of the lemma, together with their probabilities. We always choose the lemma with the highest assigned probability from the list, and return it as the translation of the source lemma.

6.1.4 Identifying Time Expressions

Sometimes it is useful to identify time expressions – usually when fixing the translation of some prepositions, as they have a specific meaning with time expressions, different from their meaning in other situations.

We use a hand-written list of 35 time expressions: the days of the week, the months of the year, periods of time (from 'second' to 'century'), and the words 'beginning' and 'end'.

6.1.5 Removing Negated Auxiliary Verbs

If a negated auxiliary verb is being removed in a fix, the negation must not be lost. It is therefore moved to the parent full verb, as shown in Example 6.1. In case

[2]The name of the block is `TranslationModel::Static::Model` and it uses the `tlemma_czeng09.static.pls.slurp.gz` model file.

of a double negative, i.e. both the auxiliary verb and the full verb are negated, the auxiliary gets removed without changing the full verb. This is because the double negation is often used in the meaning of a single negation in Czech.[3]

Source:	...the authorities are not showing enough interest in this problem.
SMT output:	...orgány **nejsou** dostatečně **prokazují** zájem o tento problém.
Gloss:	...the authorities **aren't** enough **show** interest in this problem.
Depfix output:	...orgány dostatečně **neprokazují** zájem o tento problém.
Gloss:	...the authorities enough **don't show** interest in this problem.

Example 6.1

6.2 Analysis Fixes

Analysis fixing rules try to detect and rectify tagger and parser errors. They do not change word forms and are therefore invisible on the output as such; however, rules of other types benefit from their corrections.

6.2.1 Fixing Reflexive Tantum

This rule was created by Ondřej Dušek. However, we use it as a part of Depfix and therefore provide a description of it here.

If the word form 'se' or 'si' is classified as reflexive tantum particle by the labeller (`AuxT` analytical function), but does not belong to an actual reflexive tantum verb (or a deverbative noun or an adjective), its analytical function is changed to a different value, based on the context.

Typically, the `AuxT` analytical function is changed to `AuxR` (a reflexive passive particle) – in Czech, the passive can be expressed by an active verb and a reflexive particle 'se', as shown in Example 6.2.

Source:	The pig is being roasted.
Czech:	Prase **se** peče.
Gloss:	The pig roasts **itself**.

Example 6.2

The application of the fix rule is shown in Example 6.3 and Figure 6.1. The analytical function of 'se' is changed to `AuxR`, because the verb 'provádí'

[3]However, there is no instance of a fix being performed on a double negation in our development data.

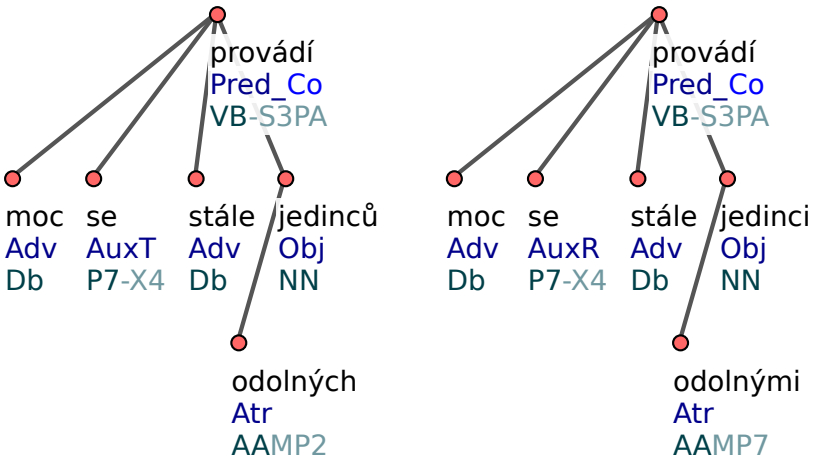


Figure 6.1: Fixing reflexive tantum

is not a reflexive tantum verb. Subsequent application of "Translation of 'by'" (Section 6.4.2) then detects the presence of a passive construction and marks 'jedinců' ('individuals') as the actor of the passive action by changing it morphological case to instrumental.[45]

Source:	. . . much is still done by hardy individuals. . .
SMT output:	. . . moc **se** stále *provádí* **odolných jedinců**. . .
Gloss:	. . . much **itself** still *does* **of hardy individuals**$_{\textbf{genitive}}$. . .
Depfix output:	. . . moc **se** stále *provádí* **odolnými jedinci**. . .
Gloss:	. . . much **itself** still *does* **by hardy individuals**$_{\textbf{instrumental}}$. . .

Example 6.3

6.2.2 Rehanging Children of Auxiliary Verbs

Auxiliary verbs must not have child nodes – the verb arguments are to be children of the full verb. Therefore, we rehang all child nodes of an auxiliary verb to its parent node (if the parent is a full verb, which it should be).

See Example 6.4 and Figure 6.2, where the subject 'většina' ('majority') is moved from its original parent, 'nebyly' ('weren't') to its correct parent, 'zvýšit' ('increase'). Thanks to this correction, subsequent applications of "Translation of passive voice" (Section 6.4.4) and "Subject - past participle agreement" (Section 6.3.3) are then able to find the correct form of the verb, which is in agreement with the subject 'většina', and even the form of the auxiliary verb is

[4]The fix is correct in nature, but the translation by passive is inappropriate here; an active structure would be better: 'moc stále provádí odolní jedinci' ('hardy individuals still do much').

[5]"Noun - adjective agreement" (Section 6.3.6) is also applied, changing the morphological case of 'odolných' ('hardy') to agree with 'jedinci'.

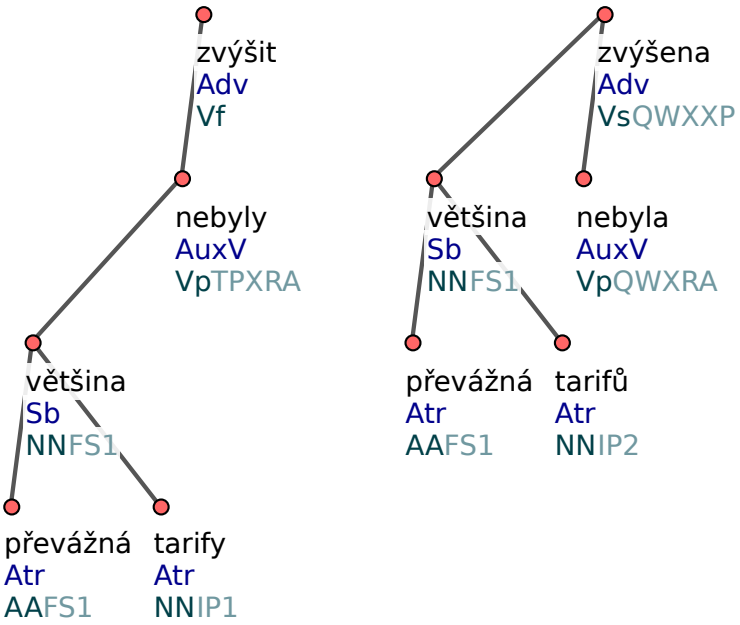


Figure 6.2: Rehanging children of auxiliary verbs

then corrected by "Passive - auxiliary 'be' agreement" (Section 6.3.4) to agree with the full verb.[6]

Source:	. . . the great majority of fares have not been increased.
SMT output:	. . . převážná *většina* tarify **nebyly zvýšit**.
Gloss:	. . . great $majority_{sg\ fem}$ of fares $\mathbf{weren't_{pl\ fem}}$ $\mathbf{increase_{inf}}$.
Depfix output:	. . . převážná *většina* tarifů **nebyla zvýšena**.
Gloss:	. . . great $majority_{sg\ fem}$ of fares $\mathbf{wasn't_{sg\ fem}}$ $\mathbf{increased_{pass\ sg\ fem}}$.

Example 6.4

6.2.3 Prepositional Morphological Case

This rule was created by Ondšej Dušek. However, we use it as a part of Depfix and therefore provide a description of it here.

This rule corrects tagger and lemmatizer errors in prepositional phrases, trying to find an analysis such that the agreement in morphological case is not violated, without changing the word forms. It is similar to the m-layer "Fixing morphological number of nouns" (Section 4.2.3), but it operates on the a-layer as it requires information about the structure of the sentence, provided by the a-tree.

The rule is applied if:

[6]The morphological case of 'tarify' ('fares') is also corrected, applying "Translation of 'of'" (Section 6.4.3).

- the child is a noun, adjective, pronoun or numeral
- the parent is a preposition
- the morphological case of child and parent do not match

The fix rule reanalizes the form of both the child and the parent, listing all possible analyses as combinations of lemma and morphological tag which do not change the part-of-speech.[7] It then goes through the possible analyses, looking for pairs of child and parent analyses that have the same morphological case. If more such pairs are found, one is chosen based on the following criteria, listed decreasingly by importance:

1. do not change the morphological case of the preposition
2. do not change the morphological case of the child node
3. prefer more likely morphological cases[8]
4. do not change the lemma
5. do not change the morphological gender
6. do not change the morphological number

If a better alternative analysis is found, the morphological tags and lemmas are changed accordingly.

If the morphological case of the preposition is about to change, but it has children that agree in the original morphological case, the fix is not performed, believing that the error is in the child word form, not in the preposition analysis. Such error is then fixed in a subsequent "Preposition - noun agreement".

The result of this fix rule is, similarly to other analysis fixing rules, not directly visible in the output – it only improves the performance of other rules, which can make use of the better analysis. However, this rule usually *lowers* the total amount of fixes performed, as the new analysis often prevents another fix rule from being invoked. Typically this is the "Preposition - noun agreement", as in Example 6.5; see also the corresponding Figure 6.3. Originally, the 'na' preposition incorrectly received the locative morphological case, which, if not fixed, results in an incorrection. If the analysis is corrected, no further fixing is performed, which is correct.

6.2.4 Preposition Without Children

A target preposition with no child nodes is clearly an analysis error. This rule tries to find children for childless prepositions by projecting the children of the aligned source preposition to the target side.

See Example 6.6 and Figure 6.4, where the preposition 's' is childless. However, it is aligned to the preposition 'with' in the source a-tree, which has one

[7]Several infrequent combinations of preposition and morphological case are not considered. These are: 's'+2, 's'+4, 'za'+2, 'v'+4, 'mezi'+4, 'z'+2, 'před'+4, 'o'+4, 'po'+4.

[8]The priority of morphological cases to be selected is as follows: 7, 6, 4, 3, 2, 5, 1

Source:	. . . alluding to the FDP election slogan "We keep our word".
SMT output:	. . . narážky **na** FDP volební **slogan** "držíme slovo".
Gloss:	. . . allusions $\textbf{to}_{\textbf{locative}}$ the FDP election $\textbf{slogan}_{\textbf{accusative}}$ "We keep our word".
Depfix without the fix:	. . . narážky **na** FDP volebním **sloganu** "držíme slovo".
Gloss:	. . . allusions $\textbf{on}_{\textbf{locative}}$ the FDP election $\textbf{slogan}_{\textbf{locative}}$ "We keep our word".
Depfix output:	. . . narážky **na** FDP volební **slogan** "držíme slovo".
Gloss:	. . . allusions $\textbf{to}_{\textbf{accusative}}$ the FDP election $\textbf{slogan}_{\textbf{accusative}}$ "We keep our word".

Example 6.5

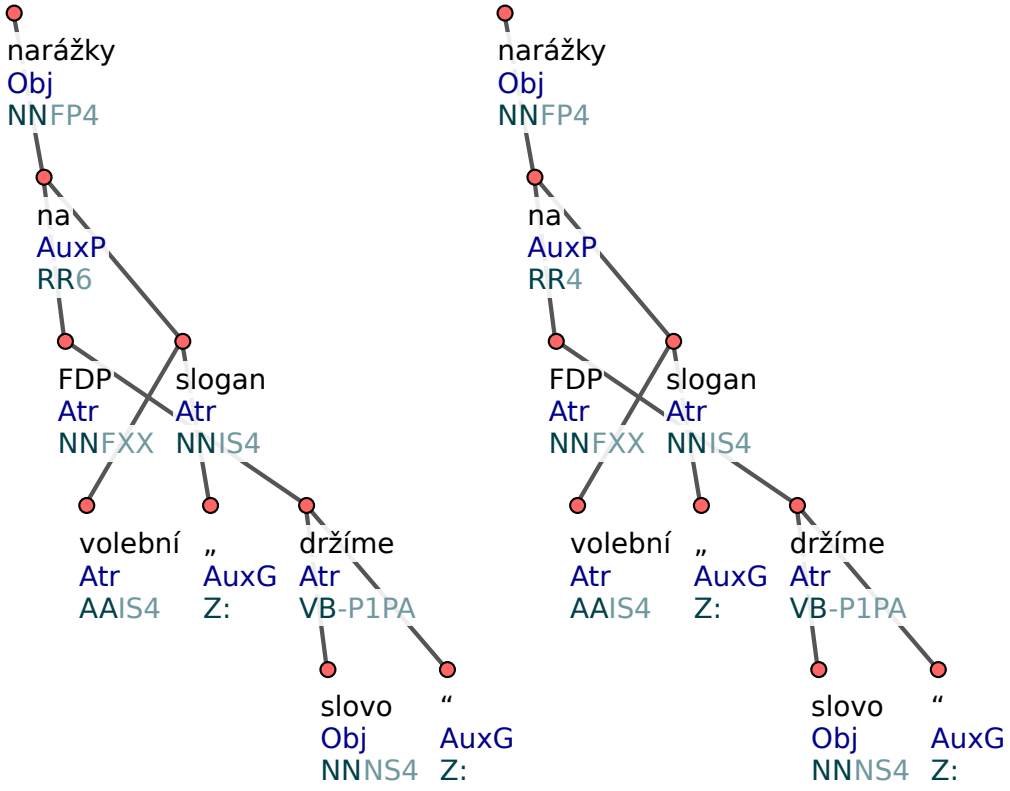


Figure 6.3: Prepositional morphological case

child, the noun ‘teeth’, which in turn is aligned to the noun ‘zuby’ in the target tree. Thus, the noun ‘zuby’ is rehung to become a child of the preposition ‘s’.

A subsequent application of “Preposition - noun agreement” (Section 6.3.1) changes the morphological case of the noun ‘zuby’ from nominative to instrumental, which is the morphological case of the preposition ‘s’, and “Noun - adjective agreement” (Section 6.3.6) then performs the same morphological case change on the adjective ‘zakřivené’ (‘curved’).

Source:	. . . a longish skull with sharp, curved teeth, (. . .) and. . .
SMT output:	. . . dlouhá lebka *s* ostrými, **zakřivené zuby**, (. . .) a. . .
Gloss:	. . . a longish skull $with_{instrumental}$ $\text{sharp}_{\text{instrumental}}$, $\textbf{curved}_{\textbf{nominative}}$ $\textbf{teeth}_{\textbf{nominative}}$, (. . .) and. . .
Depfix output:	. . . dlouhá lebka *s* ostrými, **zakřivenými zuby**, (. . .) **a**. . .
Gloss:	. . . a longish skull $with_{instrumental}$ $\text{sharp}_{\text{instrumental}}$, $\textbf{curved}_{\textbf{instrumental}}$ $\textbf{teeth}_{\textbf{instrumental}}$, (. . .) **and**. . .

Example 6.6

6.3 Agreement Fixes

The agreement fixing rules try to address the errors described in Section 3.3.1. Czech grammar typically requires agreement in morphological gender, number, case and person where applicable. These rules try to enforce the agreement in case it is violated.

The most important features of an agreement fix rule can be described by a simple list of conditions and actions, such as for “Preposition - noun agreement” (Section 6.3.1) (which description we will implicitly refer to in the following description). An agreement fix generally works like that:

1. check whether the child node is of the correct type – e.g. a noun or an adjective
2. check whether the parent node is of the correct type – e.g. a preposition
3. perform some further checks of the nodes – e.g. check that the child node precedes the parent in the sentence, and that the source node aligned to the parent node is a preposition
4. check whether the agreement is violated – e.g. the morphological case of the parent node and the child node differs
5. if passed all the checks, enforce the agreement by projecting some of the morphological attributes from one of the nodes to the other node – e.g. change the morphological case of the child node to the morphological case of the parent node
6. regenerate the word form of the changed node – e.g. regenerate the child node word form

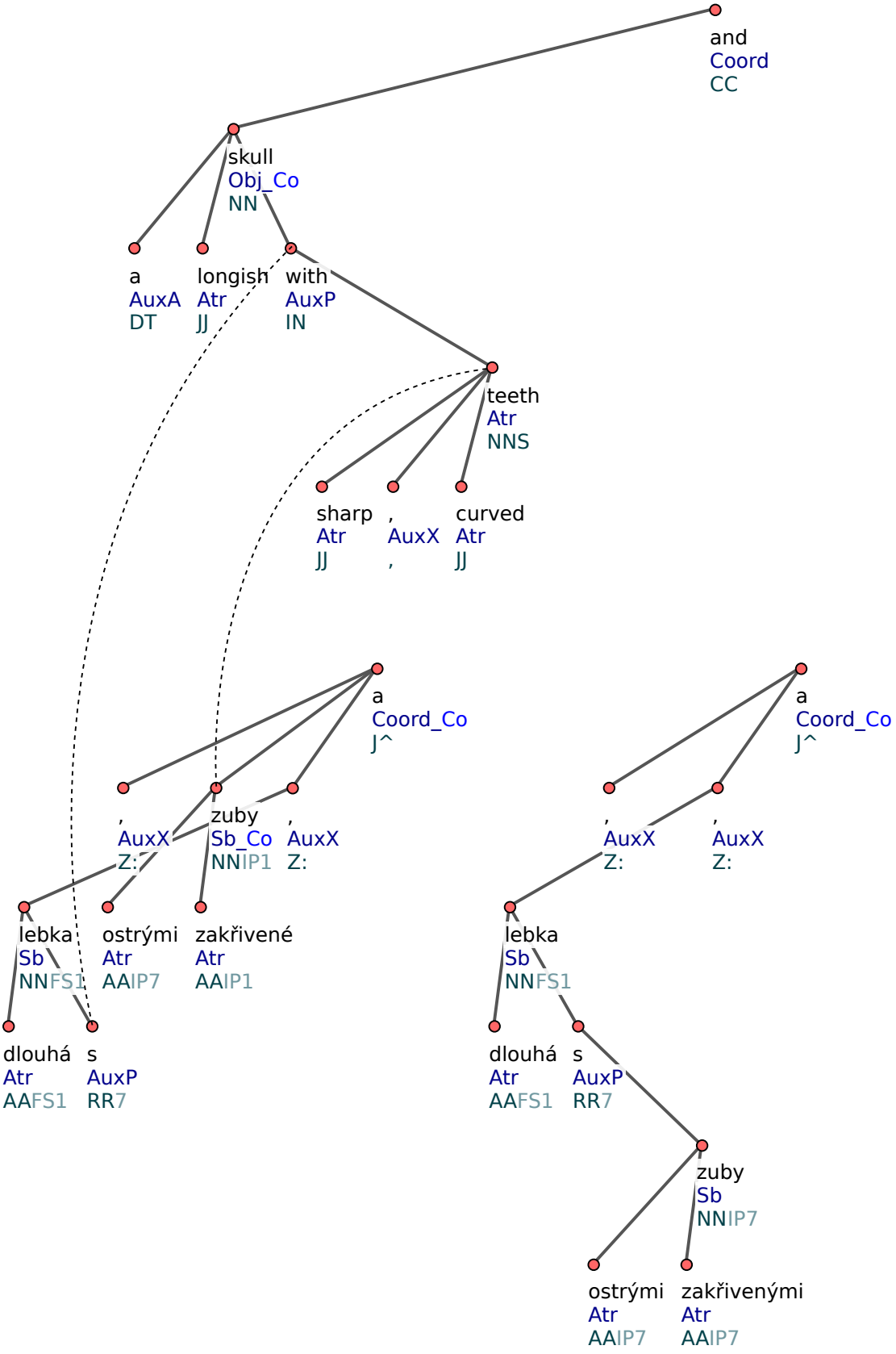


Figure 6.4: Preposition without children

6.3.1 Preposition - Noun Agreement

Conditioning
Child: noun, adjective
Parent: preposition
Checks: child precedes parent, aligned parent is a preposition
Action
Attributes: case
Projected: from parent to child

See Example 6.7 with the preposition 'o' ('about'), which requires its arguments to be in the locative morphological case. Moses produces the first word, 'sportu' ('sport'), correctly inflected, as both the translation model and the language model perform well on short-distance relations; however, the more distant arguments get an incorrect (and rather random) inflection.

The example shows that Depfix is able to correctly handle even coordinated nodes, provided that the analysis was correct. It is sometimes even able to correctly inflect foreign names, such as 'Mandela', provided that both the tagger and the morphological generator are able to handle such word correctly. It, however, also shows that Depfix is unable to correct the incorrect morphological case of the noun 'Nelson', which should also be locative ('Nelsonu'). Our attempt to correct such "noun-noun agreement", although theoretically well grounded, was unsuccessful, as it produced more incorrections than corrections, mainly due to frequent analysis errors.

Source:	It is a story about sport, race relations, and Nelson Mandela.
SMT output:	Je to příběh *o* sportu, rasových **vztahů**, a Nelson **Mandela**.
Gloss:	It is a story $about_{locative}$ $sport_{locative}$, race $\mathbf{relations_{genitive}}$, and $Nelson_{nominative}$ $\mathbf{Mandela_{nominative}}$.
Depfix output:	Je to příběh *o* sportu, rasových **vztazích**, a Nelson **Mandelovi.**
Gloss:	It is a story $about_{locative}$ $sport_{locative}$, race $\mathbf{relations_{locative}}$, and $Nelson_{nominative}$ $\mathbf{Mandela_{locative}}$.

Example 6.7

A case where the child node precedes the parent node (the preposition) is most probably an analysis error, since Czech does not have postpositions, and an attempt to fix that could probably be done, maybe by trying to project the correct tree structure from the English parse tree. However, we leave that for future research.

6.3.2 Subject - Predicate Agreement

Conditioning
Child: subject
Parent: active verb
Checks: child is not 'to' ('it'), aligned child is subject
Action
Attributes: number, person
Projected: from child to parent

The person is only projected if the subject is a pronoun, which do exhibit the person.

See Example 6.8 – with 6 words between the subject and the predicate, an SMT system which does not use any linguistic sentence-structure analysis is unlikely to capture the agreement correctly. Actually, the agreement in morphological number seems to be violated even in the source sentence ('bonuses... exceeds').

Source:	...the total bonuses awarded by the business this year, despite today's announcement, exceeds 20 billion dollars.
SMT output:	...celkové *odměny* udělované byznys letos navzdory dnešní oznámení, **přesahuje** 20 miliard dolarů.
Gloss:	...the total $bonuses_{pl}$ awarded by the business this year despite today's announcement, $\mathbf{exceeds_{sg}}$ 20 billion dollars.
Depfix output:	...celkové *odměny* udělované byznysem letos navzdory dnešní oznámení, **přesahují** 20 miliard dolarů.
Gloss:	...the total $bonuses_{pl}$ awarded by the business this year despite today's announcement, $\mathbf{exceed_{pl}}$ 20 billion dollars.

Example 6.8

6.3.3 Subject - Past Participle Agreement

Conditioning
Child: subject
Parent: past participle
Checks: child precedes parent, child is not 'to' ('it'), aligned child is subject
Action
Attributes: gender, number
Projected: from child to parent

If the child is a member of a coordination structure, the morphological number is set to plural, and the morphological gender is set to masculine animate if there seems to be at least one masculine animate subject, or to masculine inanimate otherwise.

See Example 6.9. Again, the error is probably caused by the number of words between the parent and the child – in this case even including named entities, which are likely to be very rare in the data.

The example also shows the results of application of "Source-aware truecasing" (Section 4.3.1), which helps the reader to correctly identify the named entities – a possible misinterpretation is shown in the gloss ('Pec pod Sněžkou' is a name of a town under the Sněžka mountain).

Source:	... the Horizont Hotel in Pec pod Sněžkou has seen better bookings...
SMT output:	... *hotel* horizont v peci pod sněžkou **zaznamenala** rezervace...
Gloss:	... the $hotel_{masc}$ horizont in the oven under sněžka $\mathbf{saw_{fem}}$ better bookings...
Depfix output:	... *hotel* Horizont v Peci pod Sněžkou **zaznamenal** rezervace...
Gloss:	... the Horizont $Hotel_{masc}$ in Pec pod Sněžkou $\mathbf{saw_{masc}}$ better bookings...

Example 6.9

6.3.4 Passive - Auxiliary 'be' Agreement

Conditioning
Child: auxiliary verb
Parent: passive verb
Checks: parent precedes child
Action
Attributes: gender, number
Projected: from parent to child

The morphological gender is projected only if the auxiliary verb is in the past tense, as other verb tenses do not exhibit morphological gender.

See Example 6.10. First, "Subject - past participle agreement" (Section 6.3.3) fixes 'Účinnost... testován' ('effectiveness... tested'); the long distance between these two words is beyond scope of most SMT systems. Then, passive-auxiliary verb agreement in 'byl testována' ('was tested') is corrected by the fix rule being described.[9]

[9] "Preposition - noun agreement" (Section 6.3.1) is also fixed in the sentence – in 'proti proteáza' ('against protease'), the morphological case of the noun is switched from nominative to dative, as required by the preposition.

Source:	The effectiveness of this series of substances against the HIV protease has been tested...
SMT output:	*Účinnost* této série látek proti HIV proteáza **byl** *testován*...
Gloss:	The $effectiveness_{fem}$ of this series of substances against the HIV protease $\mathbf{was_{masc}}$ $tested_{masc}$...
Depfix output:	*Účinnost* této série látek proti HIV proteáze **byla** *testována*...
Gloss:	The $effectiveness_{fem}$ of this series of substances against the HIV protease $\mathbf{was_{fem}}$ $tested_{fem}$...

Example 6.10

6.3.5 Subject - Auxiliary 'be' Agreement

Conditioning
Child: auxiliary verb
Parent: verb infinitive
Checks: subject of parent is found
Action
Attributes: gender, number
Projected: from subject of parent to child

The morphological gender is projected only if the auxiliary verb is in the past tense, as other verb tenses do not exhibit morphological gender.

If the full verb (the parent node) is in infinitive, the active auxiliary verb 'být', 'be', (the child node) has to be in agreement with the subject (which is also a child of the full verb). See Example 6.11.

Source:	...passengers will no longer be able to take Pendolino to Bratislava.
SMT output:	... *cestující* již **nebude** moci vzít Pendolino do Bratislavy.
Gloss:	... $passengers_{pl}$ will no longer $\mathbf{be_{sg}}$ able to take Pendolino to Bratislava.
Depfix output:	... *cestující* již **nebudou** moci vzít Pendolino do Bratislavy.
Gloss:	... $passengers_{pl}$ will no longer $\mathbf{be_{pl}}$ able to take Pendolino to Bratislava.

Example 6.11

6.3.6 Noun - Adjective Agreement

Conditioning
Child: syntactic adjective
Parent: noun
Checks: child precedes parent
Action
Attributes: gender, number, case
Projected: from parent to child

The child can be any adjective-like word, such as a possessive pronoun or an ordinal numeral:

- adjective (any)
- pronoun: possessive (e.g. 'můj', 'tvůj'), possessive reflexive (e.g. 'svůj'), demonstrative (e.g. 'ten', 'onen'), indefinite (e.g. 'všechnen', 'sám', 'nějaký'), or negative (e.g. 'nijaký', 'žádný')
- numeral: ordinal (e.g. 'pátý', 'šedesátý'), some indefinite (e.g. 'tolikátý'), some interrogative (e.g. 'kolikátý'), and some generic (e.g. 'dvojí', 'desaterý', 'jedny', 'čtvery')

See Example 6.12 with two instances of this fix, one on the demonstrative pronoun 'tato' ('this') and one on the adjective 'polovičatá' ('half-hearted').

Source:	...this half-hearted increase will bear the same fruit...
SMT output:	...**tato polovičatá** *nárůst* bude nést stejné ovoce...
Gloss:	...**this$_{\mathbf{fem}}$ half-hearted$_{\mathbf{fem}}$** *increase$_{masc}$* will bear the same fruit...
Depfix output:	...**tento polovičatý** *nárůst* bude nést stejné ovoce...
Gloss:	...**this$_{\mathbf{masc}}$ half-hearted$_{\mathbf{masc}}$** *increase$_{masc}$* will bear the same fruit...

Example 6.12

6.4 Translation Fixes

The following rules detect and correct structures often mistranslated by Moses but easy to fix on a rule-based basis. Most of them correct the errors that we referred to as Errors in transfer of meaning to morphology in Section 3.3.3.

6.4.1 Missing Reflexive Verbs

Reflexive tantum particles 'se' or 'si' not belonging to any verb or adjective are deleted. This situation usually occurs when the meaning of the source

verb/adjective is lost in translation and only the particle is produced, as described in Section 3.2.1.

Deleting the reflexive particle makes the translation more fluent and grammatical, but it does not help the user in understanding the translation correctly – if the user could have guessed that there was a missing reflexive verb from observing the superfluous reflexive particle, he now has probably lost the only hint. Therefore, we try to find the missing verb in source and add its translation to target, using the simple translator described in Section 6.1.3. Only if we are unsuccessful, we procede with deleting the reflexive particle.

Example 6.13 shows a very successful application of the fix, as not only does the fix improve the sentence, but even the lexical choice made by the simple translator was perfectly correct. However, the translation is still not perfect, as we are only able to generate the infinitive form of the verb by the simple translation model. This could probably be addressed by "Tense translation" (Section 7.3.3), but we have not tried that. (The example also shows a successful "Translation of possessive nouns" (Section 6.4.5) of 'DTP's'.)

Source:	. . . a thousand protesters gathered before the DTP's buildings in Diyarbakir. . .
SMT output:	. . . tisíce demonstrantů *se* před DTP je budovy v Diyarbakiru. . .
Gloss:	. . . thousands of protesters *themselves* before the DTP is buildings in Diyarbakir. . .
Depfix output:	. . . tisíce demonstrantů *se* **shromáždit** před DTP budovami v Diyarbakiru. . .
Gloss:	. . . thousands of protesters **to gather** *themselves* before the DTP buildings in Diyarbakir. . .

Example 6.13

6.4.2 Translation of 'by'

The English preposition 'by' usually marks one of two similar functions, which are translated differently to Czech:

- an author of an object (the parent of 'by' is the object – a noun) is translated using the genitive morphological case, as in Example 6.14
- an actor of an action (the parent of 'by' is the action – a passive verb) is translated using the instrumental morphological case, as in Example 6.15

The function of 'by' is translated only by the morphological case, with no preposition. Thus, if there is a preposition in the Czech sentence that is aligned to 'by', it is removed.

The fix is not performed:

- if the child precedes the parent

- if the child seems to be a named entity
- if the aligned child seems to be a time expression (e.g. 'by tomorrow')
- if the aligned parent is followed by a numeral (e.g. 'by 5 percent')

If 'by' marks the passive actor in the English sentence, but the Czech predicate is active, the child is not switched to the instrumental morphological case– it is labelled as the subject of the Czech predicate instead.

Source:	...the work done by the team of coaches...
SMT output:	...práce **na tým** trenérů...
Gloss:	...the work **on the team** of coaches...
Depfix output:	...práce **týmu** trenérů...
Gloss:	...the work **by the team** of coaches...

Example 6.14

Source:	The timing of his strategy is foiled by his voluntarism.
SMT output:	Načasování jeho strategie je zmařena **jeho voluntarismu**.
Gloss:	The timing of his strategy is foiled **of his voluntarism**.
Depfix output:	Načasování jeho strategie je zmařeno **jeho voluntarizmem**.
Gloss:	The timing of his strategy is foiled **by his voluntarism**.

Example 6.15

6.4.3 Translation of 'of'

The English preposition 'of', when modifying a noun, typically expresses that the first noun (the parent of 'of') somehow belongs to the second noun (the child of 'of'). In Czech, such meaning is expressed by the genitive morphological case.

This fixing rule changes the morphological case of the child node to genitive (see Example 6.16) if all of the following conditions are met:

- the parent of the aligned child is 'of'[10]
- the parent is not a preposition (it would have to be removed by the fix, which proved to make more incorrections than corrections)
- the child does not seem to be a named entity
- the aligned child is not a numeral

[10]Please note that the "parent of the aligned child" is not necessarily the same as the "aligned parent", i.e. the "node aligned to the parent": in the former case, we use a source tree child-parent relation, while in the latter, it is a target tree relation.

Source:	. . . unsustainable deficit level of public finances.
SMT output:	. . . neudržitelná *úroveň schodku* **veřejné finance**.
Gloss:	. . . unsustainable *deficit level* **public finances**.
Depfix output:	. . . neudržitelná *úroveň schodku* **veřejných financí**.
Gloss:	. . . unsustainable *deficit level* **of public finances**.

Example 6.16

6.4.4 Translation of Passive Voice

This rule sets the parent verb to the passive voice, if:

- the parent is a non-passive verb
- the aligned parent can be a past participle: its POS tag is `VBN` or `VBD`
- the child is the auxiliary verb 'být' ('be')
- the aligned child is the auxiliary verb 'be'
- there is no reflexive particle child ('se', 'si'), as the passive can also be expressed by a reflexive

If all the conditions are fulfilled, the parent verb is switched to passive. The verb might have been an infinitive prior to the fix; however, the morphological gender and morphological number will be filled by a subsequent application of "Subject - past participle agreement" (Section 6.3.3), as in Example 6.17.

Source:	Or the dinosaurs were better adjusted. . .
SMT output:	Nebo *dinosauři* byli lépe **přizpůsobit**. . .
Gloss:	Or the $dinosaurs_{masc\ pl}$ were better $\mathbf{adjust_{inf}}$. . .
Depfix output:	Nebo *dinosauři* byli lépe **přizpůsobeni**. . .
Gloss:	Or the $dinosaurs_{masc\ pl}$ were better $\mathbf{adjusted_{masc\ pl}}$. . .

Example 6.17

6.4.5 Translation of Possessive Nouns

English possessive nouns are often misanalyzed by Moses, missidentifying the possessive ending ''s' as a contraction of 'is', and thus translating is as a verb – see Example 6.18.

If the source analysis correctly identifies the ''s' as possessive ending (signalled by the `POS` POS tag), this error can be fixed in two ways:

Source:	David's fish
SMT output:	**David je ryba**
Gloss:	**David is fish**
Depfix output:	**Davidova ryba** OR **ryba Davida**
Gloss:	**David's$_{\text{masc}}$ fish$_{\text{fem}}$** OR **fish of David**

Example 6.18

- If the morphological generator is able to generate the appropriate possessive adjective[11] lemma ('David' – 'Davidův'), we replace the incorrect translation by that adjective in its base form, i.e. identical to the lemma ('David je ryba' – 'Davidův ryba'). The adjective-noun agreement between the possessor ('Davidův') and the possessee ('ryba') might be violated; however, the subsequent "Noun - adjective agreement" (Section 6.3.6) will correct this. See Example 6.19, where the fix is correctly performed, although, unfortunately, the lexical choice made by Moses is wrong.
- If we are unable to generate the possessive adjective, we resort to an alternative translation, setting the possessor to the genitive morphological case and changing the word order so that the possessor follows the possessee, corresponding to the English 'of' possessive construction ('fish of David' – 'ryba Davida'). See Example 6.20, where the fix leads to a significant improvement of the translation.

Source:	... the chancellor's figures...
SMT output:	... **kancléř je postavy**...
Gloss:	... **the chancellor is persons**...
Depfix output:	... **kancléřovy postavy**...
Gloss:	... **the chancellor's persons**...

Example 6.19

In both of these cases, the superfluous verb is deleted and the target dependency tree structure is modified using information from the source dependency tree structure

6.4.6 Translation of Present Continuous

If the source sentence is in a continuous tense, the auxiliary verb 'to be' from the source must not appear in the target output. This rule deletes the auxiliary verb in target and transfers its morphological categories to the main verb, as shown in Example 6.21

The fix is performed only if:

[11] Traditionally, English possessive nouns correspond to Czech possessive adjectives, although this is mainly a matter of terminology.

Source:	. . . Janota's possible continuation in office will be the topic of Friday's meeting.
SMT output:	. . . **Janota je možné pokračování** ve funkci bude tématem páteční schůze.
Gloss:	. . . **Janota is possible continuation** in office will be the topic of Friday's meeting.
Depfix output:	. . . **možné pokračování Janoty** ve funkci bude tématem páteční schůze.
Gloss:	. . . **possible continuation of Janota** in office will be the topic of Friday's meeting.

Example 6.20

- the child is a finite form of the verb 'být' ('be')
- the parent is a verb
- the aligned child is a form of the verb 'be'
- the aligned parent, or the parent of the aligned child, is a gerund and is preceded by the aligned child

Source:	. . . he is accepting the prize as a president whose country. . .
SMT output:	. . . **je akceptovat** cenu jako prezident, jehož země. . .
Gloss:	. . . he **is**$_{\textbf{3rd pers sg pres}}$ **accept**$_{\textbf{inf}}$ the prize as a president, whose country. . .
Depfix output:	. . . **akceptuje** cenu jako prezident, jehož země. . .
Gloss:	. . . he **accepts**$_{\textbf{3rd pers sg pres}}$ the prize as a president, whose country. . .

Example 6.21

6.4.7 Subject Morphological Case

The subject of a Czech sentence typically must be in the nominative morphological case. The fix is performed only if the aligned child is also a subject, or it seems to be the passive actor (its parent is 'by').

Example 6.22 shows a successful application of this rule – the word 'voliče' ('voters') is originally in the accusative morphological case, which typically denotes the object. However, when the morphological case is switched to nominative, it becomes clear that the voters are actually the subject.

"Noun - adjective agreement" (Section 6.3.6) is applied afterwards on the adjective 'švýcarské' ('Swiss'), further improving the translation.

This rule is somewhere in the middle between the categories of analysis fixing rules and translation fixing rules. If the subject form can be a nominative form

Source:	At a time when Swiss *voters* have called for a ban on the construction of minarets...
SMT output:	V době, kdy švýcarské **voliče** vyzvali k zákazu výstavby minaretů...
Gloss:	At a time, when $\text{Swiss}_{\text{acc}}$ $\textbf{voters}_{\textbf{acc}}$ were called for a ban on the construction of minarets...
Depfix output:	V době, kdy švýcarští **voliči** vyzvali k zákazu výstavby minaretů...
Gloss:	At a time, when $\text{Swiss}_{\text{nom}}$ $\textbf{voters}_{\textbf{nom}}$ called for a ban on the construction of minarets...

Example 6.22

in any morphological number, it only corrects the morphological case marker in morphological tag and thus only fixes the analysis. However, if the subject form is definitely not in the nominative morphological case, it changes the morphological case and regenerates the form, ensuring a correct transfer of the "subjectness" of the subject.

6.4.8 Subject Categories Projection

Czech is a pro-drop language, which means that if the source subject is a personal pronoun, it is usually dropped in the target sentence, and the target sentence then does not directly contain the subject (the subject is dropped). However, several morphological categories of the subject – person, morphological gender and morphological number– are still marked on the predicate (in other words, the subject-predicate agreement holds even if the subject is not expressed in the sentence).

Thus, if the subject of the source sentence is a personal pronoun, some of its morphological categories are propagated to the target predicate:

- person
- number (except for 'you', which does not exhibit number)
- gender (only in case of 'he' or 'she', which exhibit the natural gender)[12]

See Example 6.23, where fixing the error significantly changes the meaning of the sentence.

This fix is somewhat complementary to "Subject personal pronouns dropping" (Section 7.3.2), which handles the situation when the subject pronoun was **not** dropped by the SMT system.

This fix could therefore be categorized both as a translation fix and as an agreement fix – it ensures subject-predicate agreement, but it is the agreement between the source subject and the target predicate.

[12]The morphological gender in Czech follows the natural gender whenever it is defined. There are a few exceptions, such as the neuter 'děvče' (a less frequent variant of the feminine 'dívka' – 'a girl'), but they are very rare.

Source:	They claim that *we*'re more expensive than Czech Railways
SMT output:	Tvrdí, že **jsou** dražší než české dráhy
Gloss:	They claim that **they're** more expensive than Czech Railways
Depfix output:	Tvrdí, že **jsme** dražší než české dráhy
Gloss:	They claim that **we're** more expensive than Czech Railways

Example 6.23

6.5 Ordering of the Rules

The order of rule application is important as there are dependencies among the rules – e.g. a rule that changes the morphological case of a noun has to be applied prior to applying a rule that changes the morphological case of an adjective according to the morphological case of that noun. Fortunately, there exists a topological ordering of the rules (according to which we apply the rules).

The rules are applied in the following order:

1. "Translation of possessive nouns" (Section 6.4.5)
2. "Prepositional morphological case" (Section 6.2.3)
3. "Fixing reflexive tantum" (Section 6.2.1)
4. "Translation of passive voice" (Section 6.4.4)
5. "Fixing morphological number of nouns" (Section 4.2.3)
6. "Preposition without children" (Section 6.2.4)
7. "Translation of 'by'" (Section 6.4.2)
8. "Rehanging children of auxiliary verbs" (Section 6.2.2)
9. "Subject morphological case" (Section 6.4.7)
10. "Subject - auxiliary 'be' agreement" (Section 6.3.5)
11. "Translation of present continuous" (Section 6.4.6)
12. "Subject - predicate agreement" (Section 6.3.2)
13. "Subject - past participle agreement" (Section 6.3.3)
14. "Subject categories projection" (Section 6.4.8)
15. "Passive - auxiliary 'be' agreement" (Section 6.3.4)
16. "Preposition - noun agreement" (Section 6.3.1)
17. "Translation of 'of'" (Section 6.4.3)
18. "Noun - adjective agreement" (Section 6.3.6)
19. "Missing reflexive verbs" (Section 6.4.1)

Chapter 7

Tectogrammatical Layer

In this chapter, we move to a deeper-syntax layer, called tectogrammatical layer, or t-layer for short, which provides us with some useful abstractions over the a-layer, allowing us to perform many fixes easier.

First, we describe the t-layer in Section 7.1, focusing on aspects relevant for Depfix, also including a brief overview of our approach to the theory of valency. We then give some details on analyzing the sentences to t-layer in Section 7.2; however, as opposed to a-layer analysis, we made only little modifications to the pre-existing analysis pipeline in Treex.

Next, we describe the fixes performed on t-layer, which are both rule-based (Section 7.3) and statistical (Section 7.4). Although the fixes operate on t-layer, the corrections they make are immediately projected into the a-layer, which then allows us to easily generate the corrected sentence, as there is a 1:1 correspondence between a-nodes and surface tokens. Various changes are made to the a-nodes – they can be deleted, their form or even lemma can be changed (we again use the morphological generator already described in Section 6.1.2), and even new a-nodes can be created.

7.1 Tectogrammatical Trees and Valency

7.1.1 Tectogrammatical Dependency Trees

Tectogrammatical trees, or t-trees, are deep syntactic dependency trees based on the Functional Generative Description (Sgall, 1967). However, in our work, we adopt a simplification defined by Treex (Popel and Žabokrtský, 2010), which omits many important aspects of the original theory, such as the topic-focus articulation or generating of elided nodes.

Each node in a tectogrammatical tree, called a *t-node*, corresponds to one content word, such as a noun, a full verb or an adjective; the node consists of the lemma of the content word, called the *t-lemma*,[1] and several other attributes – in our work, we only use the t-lemmas, the *formemes* (see Section 7.1.2), and several *grammatemes* (see Section 7.1.3). Functional words, such as prepositions

[1]The t-lemma is a copy of the a-layer lemma of the content word. The theory originally supposed that the t-lemmas would be a generalization over the a-layer lemmas in some cases, but this is not fully implemented in Treex; only the lemmas of personal pronouns are replaced by the `#PersPron` t-lemma.

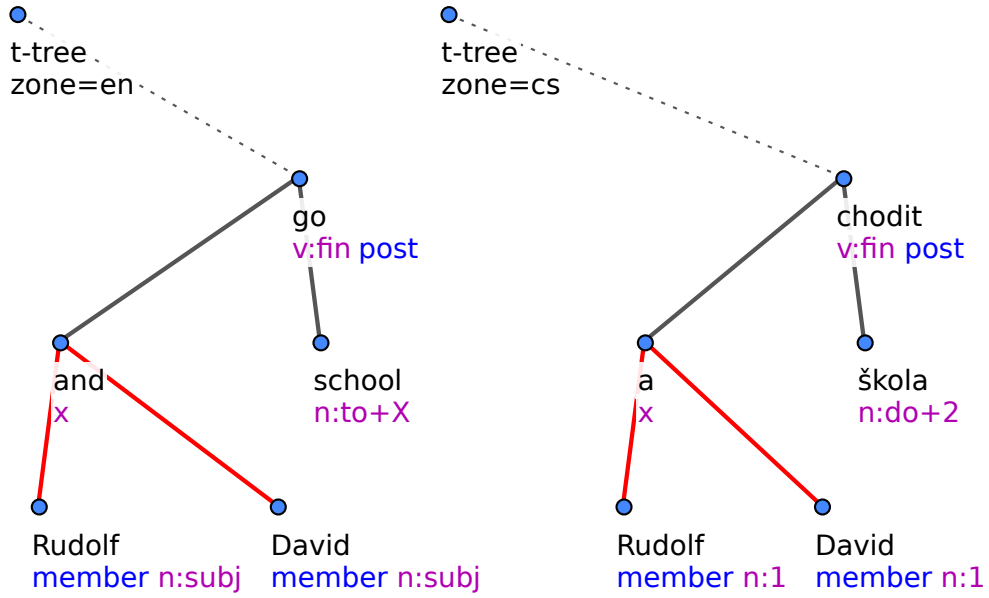


Figure 7.1: A pair of t-trees for the sentence 'Rudolf and David will go to school.' – 'Rudolf a David budou chodit do školy.'.

or auxiliary verbs or punctuation, are not directly present in the tectogrammatical tree, but are represented by the attributes of the respective content nodes.

Figure 7.1 shows a pair of t-trees, which correspond to the sentence that was already used in Section 5.1 to illustrate the a-trees, i.e. 'Rudolf and David will go to school.' – 'Rudolf a David budou chodit do školy.'. (The a-trees for that sentences were shown in Figure 5.1.) For simplicity, the values of grammatemes are not shown, except for the `tense` grammateme for the verbs (`post`).

For ease of use, each t-node contains links to all of the a-nodes that it represents – typically, there is one link to the *lexical node*, which is the content word, and several links to *auxiliary nodes*. This, among other benefits is brings, allows us to perform fixes on t-nodes and then only regenerate the relevant a-nodes.

7.1.2 Formemes

A formeme is a string representation of selected morpho-syntactic features of the content word and selected auxiliary words that belong to the content word, devised to be used as a simple and efficient representation of the node.

A noun formeme, which we are most interested in, consists of the following three parts; please refer back to Figure 7.1 for the examples:

1. The syntactic part-of-speech – `n` for nouns.
2. The preposition if the noun has one (empty otherwise), such as 'to' in `n:to+X` ('to school') or 'do' in `n:do+2` ('do školy').
3. A form specifier.
   - In English, it typically marks the subject or object, as in `n:subj` ('Rudolf', 'David'); other values exist as well, such as `ger` for the

gerund. In case of a noun accompanied by a preposition, the third part is always `X`, as in `n:to+X` ('to school').

- In Czech, it denotes the morphological case of the noun, represented by its number (see Section 4.1.1), as in `n:1` ('Rudolf', 'David') or `n:do+2` ('do školy').

Adjectives and nouns can also have the `adj:attr` and `n:attr` formemes, respectively, meaning that the node is in morphological agreement with its parent. This is especially important in Czech, where this means that the word bears the same morphological case as its parent node. A pair of similar formemes is `adj:poss` and `n:poss`, marking possessive adjectives (in Czech) and possessive nouns (in English).[2] In Depfix, `adj:poss` formemes are treated similarly to `adj:attr` formemes, since an `adj:poss` t-node is also in morphological agreement with its parent.

7.1.3 Grammatemes

The *grammatemes* are t-node attributes that carry similar information as the morphological categories in morphological tag, such as the number, gender, or tense. Usually, there is a simple one-to-one correspondence between the morphological tag categories and the grammateme values, but sometimes, the grammatemes add a deeper level of abstraction.

The grammatemes are well-defined for Czech, as they are based on the theory by Sgall (1967). The definitions of their values on English sentences are rather fuzzy; currently, they are mostly defined by implementation.

In our work, we usually do not use the grammatemes, except for a few cases. Therefore, we only describe the grammatemes that we use.

Negation grammateme

The `negation` grammateme, used in "Negation translation" (Section 7.3.1)), provides a useful abstraction level by marking the negation if either the lexical node or any of the auxiliary nodes are negated (although handling double or even higher-order negation is not resolved yet)

Verb tense grammatemes

In "Tense translation" (Section 7.3.3)), we rely on analysis of the verb tense, which is reflected in a set of several grammatemes. The abstraction level provided by the tectogrammatical layer is crucial here, as the tags only label individual words, whereas verbs often form compound forms. A compound verb form is a chain of words, containing a full verb (the lexical node) and several other words, such as auxiliary verbs, modals and prepositions (the auxiliary nodes). All of these together represent one verb tense, which cannot be found directly in the tags, but is represented on the tectogrammatical layer by attributes of the verb t-node.

We make use of the following verb tense grammatemes:

[2] English possessive nouns correspond to Czech possessive adjectives. The reasons for this difference are probably more historical than linguistic, but the formemes in Treex obey that distinction.

English phrase and its valency frame		Czech phrase and its valency frame	
n:subj go n:up+X	go up the hill	chodit do kopce	n:1 chodit n:do+2
n:subj go n:to+X	go to school	chodit do školy	n:1 chodit n:do+2
n:subj go n:to+X	go to the doctor	chodit k doktorovi	n:1 chodit n:k+3
n:subj go n:to+X	go to concerts	chodit na koncerty	n:1 chodit n:na+4
n:subj go n:for+X	go for a visit	chodit na návštěvu	n:1 chodit n:na+4

Table 7.1: Examples of valency frames of the verb 'go' and 'chodit'.

- the `tense` grammateme, which groups all the existing verb tenses into the three basic categories; its values are `ant` for past, `sim` for present and `post` for future

- the `diathesis` grammateme, which marks passive verb forms

Because of the limitations of grammatemes when applied to English, we devised our own structure to represent English tenses, together with the analysis that populates it – see Section 7.2.1.

7.1.4 Valency

The notion of *valency* (Tesnière and Fourquet, 1959) is semantic, but it is closely linked to syntax. In the theory of valency, each verb has one or more *valency frames*. Each valency frame describes a meaning of the verb, together with arguments (usually nouns) that the verb must or can have, and each of the arguments has one or several fixed forms in which it must appear. These forms can typically be specified by prepositions and morphological cases to be used with the noun, and thus can be easily expressed by formemes.

For example, the verb 'to go' in Figure 7.1 has a valency frame that can be expressed as `n:subj go n:to+X`, meaning that the subject goes to somewhere; the corresponding Czech verb 'chodit' has the frame `n:1 chodit n:do+2` with a similar meaning (`n:1` usually marks the subject, as the subject in Czech typically in the nominative morphological case, and at the same time the nominative is rare for other sentence members). Several examples of the valency of 'go' and 'chodit' are listed in Table 7.1.

In our work, we have extended our scope also to noun-noun valency (Matthews, 1981), i.e. the parent node can be either a verb or a noun, while the arguments are always nouns.[3] Therefore, we use the term *governor* for the parent word which imposes the valency frame, and *argument* for a dependent word which implements the valency frame (typically by bearing a specific preposition and/or morphological case).

[3]Practice has proven this extension to be useful, although the majority of the corrections performed are still of the verb-noun valency type.

present	continuous	conditional	going to
past	perfect	negation	modality
future	passive	infinitive	

Table 7.2: Verb tense flags.

7.2 Analysis

Treex (Popel and Žabokrtský, 2010) provides a scenario to analyze the sentences up to t-layer, with a prerequisite of already having the a-layer. The analysis is rule based.

We found the quality of the existing analysis to be sufficient, with the exception of the analysis of verb tenses, described in Section 7.2.1. We developed an improved version of the analysis, which we use instead of the original one. We describe our improvements in Section 7.2.2.

7.2.1 Original Verb Tense Analysis

The verb tense is represented by a set of grammatemes, as was briefly described in Section 7.1.3.

As the grammatemes were originally designed for Czech, they match the system of Czech verb tenses very well. The grammatemes are filled by a small set of rules, which map the values of morphological tags to the values of the verb grammatemes.

However, the situation with analysis of English tenses is more complicated, not only because the system of English verb tenses is much more complex than that the Czech one, but also because matching it to the system of tense grammatemes is not yet resolved.

Originally, the analysis of English tenses was done partly heuristically and could not capture many complex compound verb forms. Moreover, it did not capture the tenses fully – some distinctions, such as the perfectivity or the continuousness, were not made at all. All of the following tenses were grouped into a single label (`tense=ant`) – that is, if the heuristics worked correctly: present perfect simple, present perfect continuous, past simple, past continuous, past perfect simple, past perfect continuous.

7.2.2 Our Adaptations of the Verb Tense Analysis

We found the coarse approach to verb tense analysis unsuitable for our needs, and therefore developed a full analysis of the English tenses. In our approach, the tense of an English verb form is represented by a set of flags, listed in Table 7.2. The flags are binary, except for the modality flag, which is multiclass – its value is a modality type, such as debitive modality ('must'), possibilitive modality ('can', 'could') or permissive modality ('may', 'might'). The "going to" flag typically marks future, but is considered not to in case of 'were going to'.

The analysis is rule-based. It relies on the underlying analyses to be correct, especially that the compound verb form components were identified correctly – in a tectogrammatical tree, each compound verb form is represented by one t-node, which groups together all the tokens that the compound form consists

of. However, it does tolerate errors in `VBD` (past simple) / `VBN` (past participle) tagging.

The main principle of the analysis is to transcribe the verb forms into a normalized form which uses a set of only 10 different tokens, corresponding to the possible forms of the verbs 'be', 'have', and full verbs. These are able to capture the following flags: past, passive, perfect, continuous.[4]

The other flags, such as future or modality, are triggered by modifiers which are not present in the normalized form, to alleviate the combinatorial complexity of listing all of their possible combinations. They are removed during the transcription, setting the flags immediately. Some of the modifiers trigger multiple flags, such as 'should' which we understand as marking both the hortative modality and the conditionality; this is influenced by the ultimate objective to match the verb forms to their Czech counterparts, as e.g. the best translation of 'he should' – 'měl by' – should be both modal and conditional. Most of the modifiers have only one form and do not carry any other tense information, except for the following: 'have to', 'want to', 'do', 'be going to', 'be able to'. For these, the POS tag has to be carried onto the following word.

1. get all relevant tokens – verbs (`VB.*`), modals (`MD`) and the word 'able'
2. transcribe the tokens, removing modality markers (e.g. 'must' and 'should'), conditional markers (e.g. 'would' and 'should'), future markers (forms of 'will', 'shall' and 'be going to'), and forms of 'do', and setting the corresponding flags triggered by the markers
3. set the flags corresponding to the transcription
4. if the compound form could not be analysed, delete the first token and go back to step 2
5. set the negation flag if 'not' is found among the lemmas of the auxiliary nodes
6. change "present perfect conditional" to past conditional (e.g. 'would have loved')
7. change "present perfect modal" to past modal (e.g. 'must have loved')

The result is a set of flags, which is returned. Some of the flags are then mapped to grammatemes:

- the `tense` grammateme, which reflects the syntactical tense (past/present/future), with an exception of the present perfect tenses which are considered to be past tenses
- the `diathesis` grammateme, which marks the passive

[4]The tokens used are naturalistic: `be, being, were, been; have, having, had; love, loving, loved`. Thus, e.g. the past perfect tense is represented by `had loved`, and the present perfect continuous passive tense is represented by `had been being loved`. This is only for convenience of coding, any other set of tokens could be used.

- the `deontmod` grammateme, which marks presence of a modal verb (the set of values is the same as for the modality flag)
- the `verbmod` grammateme, which distinguishes the indicative, imperative and conditional modality
- the `negation` grammateme, which marks negated verbs

The only omission that we are aware of is an insufficient analysis of infinitives. We rely on preceding analysis steps to correctly identify the infinitives, but we are unsure which flags to assign to them; the system of inifitive forms in Czech is much poorer than that of English, providing little support for our decisions.

Otherwise, the accuracy of the tense analysis is close to 100%; when manually inspecting the results, we only encountered errors that were caused by errors in the preceding analysis steps, such as an auxiliary attached to an incorrect full verb or a mistagged full verb.

Still, it must be noted that the current approach to analysis of both English and Czech verb tenses is still a relaxation of the original idea of the tectogrammatical layer: Sgall (1967) supposed that the attributes of tectogrammatical nodes should capture, among other, the real (semantic or even pragmatic) tense of the verb. At present, this idea is only reflected in the set of values of the `tense` grammateme, which should have reflected the tense of a clause relatively to the tense of the parent clause. The values are `sim` for actions happening simultaneously, `post` for actions happening after and `ant` for actions happening before the "parent action". However, in practice, these values represent the absolute tense – i.e. `sim` means present, `post` means future and `ant` means past.

For many applications, it would be beneficial if the tense identification was able to capture the pragmatic tense. However, this seems to be too hard to do at the moment, since not even the syntactic tense analysis is perfect now. Applications related to machine translation therefore have to implicitly employ the assumption that a syntactic tense A in one language will usually correspond to a syntactic tense B in the other language, no matter what the semantics or the pragmatics of the tense are.

Practice has shown this assumption to be rather reasonable for the English-to-Czech translation. For example, both the English present simple tense and the Czech present tense can express both a repeated action in present, as in 'I paint pictures.' – 'Maluju obrazy.', and an action in future that happens according to a given schedule, as in 'My plane leaves at 8.' – 'Moje letadlo odlétá v 8.'.

7.3 Rule-based Fixes

In Section 3.3.3, we described a range of errors in the transfer of meaning to morphology. Most of the errors were already addressed in Section 6.4. However, three types of the errors – in negation, subject personal pronouns, and verb tense – are not corrected on the analytical layer, as the tectogrammatical layer provides useful abstractions that the fixing rules can make use of. The fixing rules that

correct the errors are "Negation translation" (Section 7.3.1), "Subject personal pronouns dropping" (Section 7.3.2), and "Tense translation" (Section 7.3.3).

7.3.1 Negation Translation

In Section 3.3.3, we described errors in negation, i.e. missing or extra negation, with missing negation being much more frequent, and discussed that these errors are very serious.

In Depfix, we decided to fix only the error of missing negation, for the following reasons:

- It is more frequent in our data, occurring especially on verbs.
- It is safer to make. If we fail to detect that the English sentence is negated, we simply do not make the correction, while if we failed in detecting the negation in English and switched the Czech negative translation into positive, it would be an incorrection.
- False positives may be not harmful. Czech is a language with double negation typically bearing the meaning of a single negation, and thus negating a sentence which already contains a negation may keep its meaning unchanged.
- It is easier to make. To negate something in Czech, it is sufficient to add the 'ne-' prefix to the word that should be negated – it can lead to an incorrect form or an ungrammatical structure, but it is most probably always understandable. On the other hand, a negative Czech sentence often contains several negation markers, and if only some of them are deleted or made positive (which by itself may not be trivial), the polarity of the resulting sentence may be difficult to tell.

To detect the polarity of both English and Czech t-nodes (or clauses), it should, in theory, be sufficient to check the value of the negation grammateme. However, we found out that in practice this is not sufficient, as often the negation grammateme is not set even if the t-node is negated. We therefore devised a set of rules that, using some heuristics, are usually able to correctly detect the polarity of a t-node, both in Czech and in English.

A Czech t-node is considered to be **negated** if at least one of the following conditions is met:

- its negation grammateme is set to `neg1`
- its lemma seems to bear a negative prefix, such as 'ne-' ('not'), 'bez-' ('without'), 'mimo-' ('except'), or 'proti-' ('againts'), and is not found in a list of exceptions, such as 'nebo' ('or'), 'nedávno' ('recently'), 'neutr*' ('neutr*') or 'netopýr' ('bat')
- its formeme contains a negative preposition, such as 'bez' ('without'), 'mimo' ('except'), or 'proti' ('againts')

- there is a negated t-node among its child nodes; this is obviously recursive, but the maximum depth of recursion is set to 1

An English t-node is considered to be **negated** if one of the two following conditions is met:

- its negation grammateme is set to `neg1`
- there is 'no' or 'not' among its child nodes

A t-node is considered to be **positive** if:

- it is not considered to be negated
- neither its parent nor its grandparent is in the same clause as the node and considered to be negated (leads to a lower recall but higher precision)

If a positive Czech node is aligned to a negated English node, we add negation into the Czech sentence. If the parent of the non-negated node is a finite verb, we negate that verb, otherwise we negate the node. For a node that corresponds to a compound verb form, we negate the first of the verbs but skipping the conditional 'by', 'bych'... ('would') as these cannot be negated.

We first try to negate a word by switching on the negation flag in the morphological tag and calling the morphological generator to generate a new form. If this fails, we simply prefix the form with the negative prefix 'ne-'. Moreover, if the lemma is 'muset' ('have to'), we change the lemma to 'smět' ('can'), as the correct negation of 'musím' ('I have to') is not 'nemusím' ('I do not have to'), but 'nesmím' ('I cannot').

See Example 7.1, where the error in the original sentence completely reversed the meaning.

Source:	... he feels that he does not wholly deserve it.
SMT output:	... cítí, že si plně **zaslouží**.
Gloss:	... he feels that he wholly **deserves**.
Depfix output:	... cítí, že si plně **nezaslouží**.
Gloss:	... he feels that he wholly **does not deserve**.

Example 7.1

7.3.2 Subject Personal Pronouns Dropping

If the subject of a Czech sentence is a personal pronoun, it is often dropped – see Example 7.2. The translation is correct even if the pronoun is not dropped, but it is considered to be less natural.

It is worth noting that the dropping is possible because the morphological categories of the subject are reflected on the verb and the subject pronoun

Source:	She lives in Kladno.
SMT output:	**Ona** *bydlí* na Kladně.
Gloss:	**She** $lives_{3rd\ person\ sg}$ in Kladno.
Depfix output:	*Bydlí* na Kladně.
Gloss:	$Lives_{3rd\ person\ sg}$ in Kladno.

Example 7.2

therefore carries no additional information.[5] This fix is thus somewhat complementary to "Subject categories projection" (Section 6.4.8), which tries to enforce the agreement between the predicate and the dropped subject using information about the source subject.

The phenomenon of pronoun dropping exists in many languages and has been studied e.g. by Adams (1987), McShane (1999) or Muller (2006). The authors study the development of pro-dropping, often comparing various languages, and provide explanations for the phenomenon. However, they do not provide a description or a set of rules for pro-dropping that we could easily implement into Depfix– probably because construction of such a set of rules would be hard or impossible to do. For example, Lindseth (1998) notes that "[i]n stylistically unmarked discourse, Czech (...) omits unstressed pronominal subjects". This is probably true, but the level of analysis that we have to our disposal provides us neither with the information whether the discourse is stylistically unmarked, nor whether the pronominal subjects are stressed.

Therefore, we only constructed a heuristic block that tries to decide whether the subject pronoun should be dropped, based partly on our linguistic intuition and mainly on observations on the development data. Therefore, we are unable to provide a sane linguistic grounding for most of the rules. If we encountered a situation in which we were unable to set up a rule that would separate the droppable and undroppable cases, we do not drop the pronoun in any of the cases, since dropping an undroppable pronoun can make the sentence incomprehensible while not dropping a droppable pronoun usually only results in a less natural but still comprehensible sentence.

Our main observation is that the subject pronoun 'to' ('it') is both the most common and the most difficult to handle. For example, it is often part of fixed expressions, which become unnatural or even incomprehensible when the pronoun is removed.

We drop the subject personal pronoun by default. The pronoun is **not** dropped if it fulfills any of the following conditions:

- it is not in the nominative case

[5]The subject and predicate agree in person and number; in the past tense, they also agree in gender for the third person (both in singular and in plural). This implies that the gender information is lost by dropping the third person pronoun if the verb is in the present or future tense.

- it is followed by a comma, as in 'Já, a to je moje zvláštnost, si maltu vždycky míchám sám.' ('I, and this is my special feature, always mix my mortar myself.')
- it is followed by 'sám' ('*self'), as in 'on sám' 'he himself'
- it is followed or preceded by 'nic' ('nothing') or 'vše' ('all'), as in 'Já nic neřekl.' ('I said nothing.') or 'My všichni to slyšeli.' ('All of us heard it.')
- it is preceded by a verb – usually it is the parenting verb – as in 'Mám já to ale krásného ptakopyska!' ('Don't I have a beautiful platypus!')
- it is coordinated, as in 'on nebo ona' ('he or she')

Moreover, if the pronoun is 'to' ('it'), it is also not dropped if any of the following conditions is met:

- it is at the beginning of the sentence (this has many false positives)
- it does not have a source counterpart, which could mean that it was generated thanks to the language model
- its source counterpart is not 'it'; usually this is e.g. 'this' or 'that', which may suggest emphasis
- its parent is either 'být' ('to be') or 'znamenat' ('to mean'), as in 'Zítra to bude v novinách.' ('Tomorrow it will be in the newspaper.') or 'Možná to znamená, že...' ('Maybe it means, that...')

If the pronoun passes all of the checks, it is dropped.

If the main verb follows the pronoun, we also shift the verb into the original position of the pronoun, loosely obeying the rule of the Wackernagel position (Avgustinova and Oliva, 1997) – the personal pronoun was probably in the "first" position and there were probably other words in the "second" position, and removing the pronoun might cause the second position word to be moved into the first position where it probably does not belong. By shifting the verb into the first position instead of the pronoun, we ensure that the first position remains occupied and the second-position words stay where they should be.

See Example 7.3 and Example 7.4. The word in the second position, i.e. the auxiliary verb 'jsem' in the first example or the reflexive particle 'se' in the second, must keep its second position even after the the first-position pronoun is removed, which is fulfilled by moving the full verb ('utekl', 'nedivím').

7.3.3 Tense Translation

It has been shown in Section 3.3.3 that Moses can perform very badly in transferring the verb tense from English to Czech, as a purely statistical approach with little linguistic knowledge fails to handle compound verb forms correctly.

However, we found that a fully rule-based approach to tense transfer is not perfect either. This is mainly due to the current state of tectogrammatical analysis, which is not deep enough in analysing verb tenses, especially with

Source:	I escaped.
SMT output:	**Já** *jsem* utekl.
Gloss:	**I** $do_{1st\ person\ sg}$ escaped.
Depfix output:	Utekl *jsem*.
Gloss:	Escaped $do_{1st\ person\ sg}$.

Example 7.3

Source:	I don't blame them.
SMT output:	**Já** *se* jim nedivím.
Gloss:	**I** *myself* them don't-blame$_{\text{1st person sg}}$.
Depfix output:	Nedivím *se* jim.
Gloss:	Don't-blame$_{\text{1st person sg}}$ *myself* them.

Example 7.4

the English language. Currently, it only captures the syntactical tense, such as present continuous, but it does not provide any deeper analysis, such as whether the present continuous tense expresses an action in present or in future – see Section 7.2.1.

Experiments have shown that in cases of verb forms with high ambiguity in choosing the correct tense for the Czech translation (such as the present perfect tense, which does not exist in Czech and has to be translated either as the past tense or the present tense, based on meaning and context), we probably cannot easily improve the quality of the translation produced by Moses only by employing rules. We would have to resort to a statistical approach to sufficiently address the important features of meaning and context, which, however, Moses is already good at.

The most difficult type of sentences for rule-based corrections seem to be English sentences with tense shifting, such as reported speech, as the tense shifting does not happen in Czech – see Example 7.5. To translate an English sentence with tense shifting correctly, it would be necessary to detect that tense shifting happened, and to perform a reverse tense shift before the translation. However, the English tense shifting is not injective and thus is not easily reversible.

Source:	Peter *told* me I **was singing** well.
Interpretation:	Peter *told* me: "You **are singing** well."
Correct translation:	Petr mi *řekl*, že **zpívám** dobře.
Gloss:	Peter *told* me I **am singing** well.

Example 7.5

Moreover, from our data it seems that the tense shifting rule is treated as optional in current English. See Example 7.6, where both of the interpretations can be correct. It is therefore very risky to perform any rule-based corrections on sentences with tense shifting. We do our best to avoid fixing verbs which might be shifted, especially when the verb is in a past tense, as a present or future tense suggests that the verb is not shifted – except for the past perfect tense, which can be safely considered to express some kind of past, regardless of the shifting.[6]

Indirect speech:	Peter *told* me I **was** singing well.
Possible interpretation:	Peter *told* me: "You **are** singing well."
Possible interpretation:	Peter *told* me: "You **were** singing well."

Example 7.6

The English conditionals are also hard to translate correctly and we therefore try to avoid them as well. It might be possible to first properly analyze the conditional type being used in the sentence and then use a set of rules to choose its best translation, but we leave that for future research.

Based on observed low accuracy when trying to correct the tense in many cases, we do **not** perform the fix if any of these conditions holds:

- there is 'that' among the auxiliary nodes of the source verb t-node and the verb is in a past tense (but not past perfect)
- there is a parenting dicendi verb in a past tense, such as 'said', and the source verb concerned is also in a past tense (but not past perfect)
- either the source or the target verb is a conditional
- there is 'if' or 'when' among the auxiliary nodes or child nodes of the source verb t-node
- the source verb is in a present tense and one (but not both) of the verbs is in passive
- either the source or the target verb is an infinitive
- either the source or the target verb is an imperative

If none of the aforementioned conditions aborts the fix, we try to change the tense of the Czech verb to the tense of the English verb. We use the following mapping:

- English future tenses are mapped to the Czech future tense

[6]We are aware of the marginal case of e.g. '. . . he had said: "She is nice."', which could be shifted to '. . . he had said she had been nice.'. However, we have not observed any such sentence in our data and believe such construction to be extremely rare.

- English present tenses (except for present perfect) are mapped to the Czech present tense
- English past tenses and present perfect tenses are mapped to the Czech past tense
- English conditional is mapped to the Czech conditional
- English passive is mapped to the Czech passive

We provide several examples of the results of the fix rule, changing the tense from future to present (Example 7.7), from present to future (Example 7.8),[7] from past to present (Example 7.9), from present to past (Example 7.10), and from past to future (Example 7.11).

Source:	. . . you need time and steady nerves.
SMT output:	. . . **budete potřebovat** čas a pevné nervy.
Gloss:	. . . you **will need** time and steady nerves.
Depfix output:	. . . **potřebujete** čas a pevné nervy.
Gloss:	. . . you **need** time and steady nerves.

Example 7.7

Source:	This will bring problems for whoever is in office. . .
SMT output:	To **přináší** problémy pro každého, kdo je v kanceláři. . .
Gloss:	This **brings** problems for anyone who is in office. . .
Depfix output:	To **bude přinášet** problémy pro každého, kdo je v kanceláři. . .
Gloss:	This **will bring** problems for anyone who is in office. . .

Example 7.8

Source:	The generals are defending themselves. . .
SMT output:	Generálové se **bránili**. . .
Gloss:	The generals **were defending** themselves. . .
Depfix output:	Generálové se **brání**. . .
Gloss:	The generals **are defending** themselves. . .

Example 7.9

[7] Although the result of the fix is correct, it would be more natural to generate the one-word future form 'přinese' instead of the compound form bude přinášet.

Source:	Amnesty also cited the case of a former detainee...
SMT output:	Amnesty rovněž **cituje** případ bývalého vězně...
Gloss:	Amnesty also **cites** the case of a former detainee...
Depfix output:	Amnesty rovněž **citoval** případ bývalého vězně...
Gloss:	Amnesty also **cited** the case of a former detainee...

Example 7.10

Source:	...the direct service from Prague - Letohrad will be cut dramatically.
SMT output:	...přímé spojení z Prahy - letohrad se dramaticky **snížil**.
Gloss:	...the direct service from Prague - Letohrad **was lowered** dramatically.
Depfix output:	...přímé spojení z Prahy - Letohrad se dramaticky **sníží**.
Gloss:	...the direct service from Prague - Letohrad **will be lowered** dramatically.

Example 7.11

7.4 Statistical Fixes

In this section, we describe a statistical approach to correcting errors in the verb-noun and noun-noun valency, which we described in Section 3.3.2.

Our approach is to use deep linguistic analysis to automatically determine the structure of each sentence, and to detect and correct valency errors using a simple statistical valency model.

7.4.1 Evaluation of Existing SPE Approaches

First, we decided to evaluate the utility of the approach of Béchara et al. (2011) for the English-Czech language pair. The evaluation was performed by Aleš Tamchyna; however, its description has not yet been published. Therefore, we cite his approach and results here.

We used 1 million sentence pairs from CzEng 1.0 (Bojar et al., 2012b), a large English-Czech parallel corpus. Identically to the paper, we split the training data into 10 parts, trained 10 systems (each on nine tenths of the data) and used them to translate the remaining part. The second step was then trained on the concatenation of these translations and the target side of CzEng. We also implemented the *contextual* variant of SPE where words in the intermediate language are annotated with corresponding source words if the alignment strength is greater than a given threshold. We limited ourselves to the threshold value 0.8, for which the best results are reported in the paper. We tuned all systems on the dataset of WMT_{11} (Callison-Burch et al., 2011) and evaluated on the WMT_{12} dataset (Callison-Burch et al., 2012).

Table 7.3 summarizes our results. The reported confidence intervals were estimated using bootstrap resampling (Koehn, 2004). SPE did not lead to any improvements of BLEU in our experiments. In fact, SPE even slightly decreased the score (but the difference is statistically insignificant in all cases).

We conclude that this method does not improve English-Czech translation, possibly because our training data is too large for this method to bring any benefit. We therefore proceed with a more complex approach which relies on deep linguistic knowledge.

7.4.2 Valency Models

To be able to detect and correct valency errors, we created statistical valency models. We model the conditional probability of the argument formeme based on several features of the governor-argument pair. We decided to use the following two models:

$$P(f_{arg}|l_{gov}, f_{al.arg}) \tag{7.1}$$

$$P(f_{arg}|l_{gov}, l_{arg}, f_{al.arg}) \tag{7.2}$$

where:

- f_{arg} is the formeme of the Czech argument
- l_{gov} is the lemma of the Czech governor
- l_{arg} is the lemma of the Czech argument
- $f_{al.arg}$ is the formeme of the English argument aligned to the Czech argument

The input is first processed by the model (7.1), which performs more general fixes, in situations where the $(l_{gov}, f_{al.arg})$ pair rather unambiguously defines the valency frame required.

Then model (7.2) is applied, correcting some errors of the model (7.1), in cases where the argument requires a different valency frame than is usual for the $(l_{gov}, f_{al.arg})$ pair, and making some more fixes in cases where the correct valency frame required for the $(l_{gov}, f_{al.arg})$ pair was too ambiguous to make a correction according to model (7.1), but the decision can be made once information about l_{arg} is added.

We computed the models on the full training set of CzEng 1.0 (Bojar et al., 2012b) (roughly 15 million sentences), and smoothed the estimated probabilities with add-one smoothing.

Direction	Baseline	SPE	Context SPE
en→cs	**10.85±0.47**	10.70±0.44	10.73±0.49
cs→en	**17.20±0.53**	17.11±0.52	17.18±0.54

Table 7.3: Results of SPE for English-Czech.

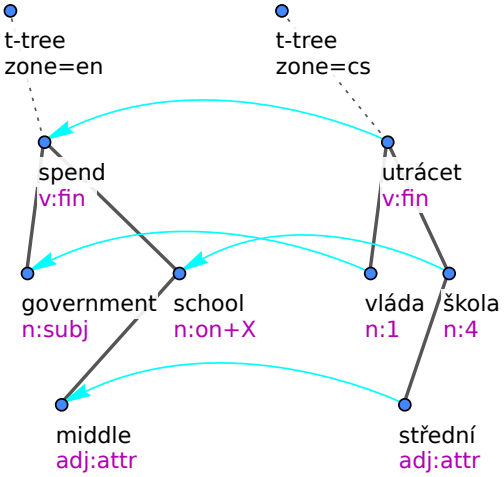


Figure 7.2: Tectogrammatical trees for the sentence 'The government spends on the middle schools.' – 'Vláda utrácí střední školy.'; only lemmas and formemes of the nodes are shown.

7.4.3 Correcting Valency Errors

The fixing pipeline consists of several steps:

1. improbable argument formemes are replaced with correct formemes according to the valency model
2. the words are regenerated according to the new formemes
3. the regenerating continues recursively to children of regenerated nodes if they are in morphological agreement with their parents; this is marked by the `*:attr` or `*:poss` formeme, see Section 7.1.2

To decide whether the formeme of the argument is correct, we query the valency model for all possible formemes and their probabilities. If an alternative formeme probability exceeds a fixed threshold (see Section 7.4.4), we assume that the original formeme is incorrect, and we use the alternative formeme instead.

Source:	The government spends on the middle schools.
SMT output:	Vláda *utrácí* **střední školy**.
Gloss:	The government **destroys the middle schools**.
Depfix output:	Vláda *utrácí* **za střední školy**.
Gloss:	The government **spends on the middle schools**.

Example 7.12

Consider Example 7.12, with the sentence that was presented in Section 3.3.2. The corresponding t-trees are shown in Figure 7.2. When processing the 'utrácí'-'školy' ('spends'-'schools') pair, we query the model (7.2) and get the following probabilities:

- P(n:4 | utrácet, škola, n:on+X) = 0.07
  (the original formeme)
- P(n:za+4 | utrácet, škola, n:on+X) = 0.89
  (the most probable formeme)

The threshold for this change type is 0.86, is exceeded by the `n:za+4` formeme and thus the change is performed: 'školy' is replaced by 'za školy'.

7.4.4 Correction Types and Thresholds

We distinguish four types of changes:

- Changing the argument morphological case only, keeping the preposition intact if there is one, such as changing `n:1` to `n:2` on 'pokrytí'-'škoda' in Example 7.13 (note that this is an example of noun-noun valency)
- Changing the preposition, as in Example 7.14, where `n:podle+2` is changed to `n:v+6` on 'být'-'dosah'
- Adding a new preposition, as in Example 7.12, where `n:4` is changed to `n:za+4` on 'utrácet'-'škola'
- Removing the preposition, as in Example 7.15, where `n:na+4` is changed to `n:2` on 'vzdát'-'plán'

Source:	The budget is almost three billion shot to **cover various damages** owed to the victims of communism and crime...
SMT output:	Rozpočet je téměř tři miliardy na *pokrytí* **různé škody** vůči obětem komunismu a zločinu...
Gloss:	The budget is almost three billion for the *covering* **various damages**$_{\text{nominative}}$ owed to the victims of communism and crime...
Depfix output:	Rozpočet je téměř tři miliardy na *pokrytí* **různých škod** vůči obětem komunismu a zločinu...
Gloss:	The budget is almost three billion for the *covering* **of various damages**$_{\text{genitive}}$ owed to the victims of communism and crime...

Example 7.13

We set the thresholds differently for different types of changes. The values of the thresholds that we used are listed in Table 7.4 and were roughly estimated using automatic evaluation and then fine-tuned manually.

For some combinations of a change type and a model, as in case of the preposition removing, we never perform a fix because we observed that it nearly never improves the translation. E.g., if a verb-noun pair can be correct both with and without a preposition, the preposition-less variant is usually much

Source:	. . . if the world record was in my reach. . .
SMT output:	. . . pokud světový rekord *byl* **podle mého dosahu**. . .
Gloss:	. . . if the world record *was* **according to my reach**$_{\text{genitive}}$. . .
Depfix output:	. . . pokud světový rekord *byl* **v mém dosahu**. . .
Gloss:	. . . if the world record *was* **in my reach**$_{\text{locative}}$. . .

Example 7.14

Source:	We can't give up on our production plan. . .
SMT output:	Nemůžeme se *vzdát* **na náš výrobní plán**. . .
Gloss:	We can't *give up* **our production plan**$_{\text{accusative}}$. . .
Depfix output:	Nemůžeme se *vzdát* **našeho výrobního plánu**. . .
Gloss:	We can't *give up* **on our production plan**$_{\text{genitive}}$. . .

Example 7.15

Correction type	Thresholds for models	
	(7.1)	(7.2)
Changing the argument morphological case only	0.55	–
Changing the preposition	0.90	0.84
Adding a new preposition	–	0.86
Removing the preposition	–	–

Table 7.4: Deepfix thresholds

more frequent than the prepositional variant (and thus is assigned a much higher probability by the model), but the preposition often bears a meaning that is lost by removing it. This is demonstrated in Example 7.16, where the removal of the preposition inverses the meaning of the sentence.

Source:	This year, women were awarded the Nobel Prize in all fields except physics
SMT output:	Letos byly ženy laureát Nobelovy ceny ve všech *oblastech* **kromě fyziky**
Gloss:	This year, women were awarded the Nobel Prize in all *fields* **except physics**
Depfix output:	Letos byly ženy laureát Nobelovy ceny ve všech *oblastech* **fyziky**
Gloss:	This year, women were awarded the Nobel Prize in all *fields* **of physics**$_{\text{genitive}}$

Example 7.16

7.4.5 Choosing the Models

In this section, we detail the various valency model definitions that we tried to use, discuss their performance and explain why we finally chose the models described in Section 7.4.2.

A first simple model

We first tried to predict the formeme of the argument (f_{arg}) by a simple model, conditioned only on the lemma of the governor (l_{gov}) and the lemma of the argument (l_{arg}):

$$P(f_{arg}|l_{gov}, l_{arg}) \tag{7.3}$$

However, we observed that also the argument type distinction is necessary, at least to distinguish the subject from the object. In Czech, the subject of a sentence is nearly always in the nominative morphological case (to be represented by the `n:1` formeme), while the object is nearly never in the nominative morphological case. Assigning or not assigning the nominative morphological case erroneously could lead to much confusion in the meaning of the sentence – due to the free order of Czech, the morphological case is one of the key features that enable a reader to recognize which word is the subject and which is the object.

In practice, not distinguishing the argument type usually lead to Depfix not performing any fix, since both the subject formeme (`n:1`) and the object formeme (usually `n:4`) are often approximately equally frequent, none of them surpassing the threshold. In case of infrequent words, the fix was sometimes performed, but its result was rather random.

Adding argument type information

A natural way to represent the argument type would be its analytical function, denoted a_{arg} in (7.4):

$$P(f_{arg}|l_{gov}, l_{arg}, a_{arg}) \tag{7.4}$$

However, the subject-object distinction is often hard to make for the automatic tools when the sentence, as in our case, is erroneous.

As a substitute heuristic, we tried to use the attachment direction instead, denoted d_{arg} in (7.5):

$$P(f_{arg}|l_{gov}, l_{arg}, d_{arg}) \tag{7.5}$$

In English, this would be a very good approximation, as the subject is nearly always a left constituent to the verb and the object a right one. However, in Czech, known to have a highly free word-order, the attachment direction is merely a heuristic. Although we found that it performs rather impressively for automatic translations from English, as opposed to natural Czech texts, it was not sufficient for our needs.

Therefore, we decided to use information from the English sentence instead. The automatic tools have better performance on the English sentence, because it comes from humans, not from an SMT system. We believe the English word aligned to the Czech argument to be the corresponding English argument, and use information about that in our model.

We tried to use the formeme of the English argument, denoted $f_{al.arg}$ in (7.6):

$$P(f_{arg}|l_{gov}, l_{arg}, f_{al.arg}) \tag{7.6}$$

This model constituted a significant performance improvement, as it not only enables us to distinguish subject from objects, but it also provides us with more fine-grained argument type specification. For example, in the sentence 'Martin met Mark on the hill at 5 o'clock.' – 'Martin potkal Marka na kopci v 5 hodin.', we predict the formemes for each of the four arguments ('Martin', 'Mark', 'the hill', '5 o'clock') distinctly, as they are assigned different formemes on the English side.

We therefore selected model (7.6) to be used as one of the final models.

A more general model

By performing a more detailed analysis of the performance of the chosen model (7.6), we realized that by conditioning our model on such the multi-valued formeme, data sparseness became an important issue: approximately 63% of the development data governor-argument pairs were unseen in the training data. In such cases, a valency error could be neither detected nor corrected. This made us explore possibilities of generalizing the model (but keeping the argument type information).

We introduced a more general model (7.7), which does not use information about the argument lemma l_{arg}, as we estimated it to be the least significant piece of information in the model:

$$P(f_{arg}|l_{gov}, f_{al.arg}) \tag{7.7}$$

Using the generalized model lead to a large increase of recall, interestingly with no significant decrease of precision. Therefore, we selected that model as well to be used as one of our final models.

The best results were obtained when both of the selected models were used one after another, as was already described in Section 7.4.2.

7.4.6 Future Work

Our approach to statistical post-editing of SMT is, to the best of our knowledge, unique. Although we manage to improve the quality of the SMT output, the improvement is rather modest, as much more research could still be done in this area.

As it was mentioned, complex formemes are typically less frequent than simple ones, even if they are perfectly correct. This fact made us not perform preposition deletion, as a prepositionless formeme is nearly always much more frequent than a formeme with a preposition, but the preposition often (correctly) conveys additional meaning which is lost by deleting it.[8] We would therefore like the scores assigned by our models to reflect that observation, which would hopefully enable us to perform more corrections, and/or to perform them more reliably. We believe that the frequency of a formeme is approximately inversely proportionate to its complexity, which probably could be approximated by the length of the formeme string; or, the absolute unconditioned frequency of the formeme could be taken into account. However, we have not explored this possibilities.

The set of thresholds we use does not account for many things; among other, it accounts neither for the score of the formeme to be replaced (it is at most 1 - the score of the other formeme, but it makes no difference whether it is really high or really low), nor for the absolute counts of occurrences of the formemes in the training data (e.g. formemes occurring only once are simply eliminated as they cannot surpass most of the thresholds because of the add-one smoothing being in effect; however, they can surpass the 0.55 threshold). We believe that the thresholds should be more fine-grained in these respects, but we were unable to set up such a set of thresholds that would lead to a better performance.

In our approach, we disregard multiword prepositions. This is done only for simplicity and, in our opinion, support for multiword prepositions should be added. However, they are very rare in our data and therefore no substantial improvement of performance is expected.

We also disregard coordinated nodes. We realized that special treatment is necessary, as the coordinated nodes should usually (but not always) have the same formemes. Moreover, the errors in underlying analyses are very frequent, making many of the changes performed by Depfix wrong – very often, the preposition is not correctly linked to the node or nodes it belongs to, and there are other issues

[8] We also encountered issues with correctly identifying the preposition to be deleted, as the prepositions can be vocalized – e.g. 'v' ('in') can become 've' for easier pronunciation, as in 've vaně' ('in the bath tub') – but the formemes contain the unvocalized forms. As the vocalization is always perfomed by adding an 'e', we search for the preposition using a regular expression: `^(the preposition)e?$`. Still, this can lead to errors.

as well. We believe that handling these cases is rather difficult, but the ability to handle them correctly would significantly improve the overall performance. It should also be reminded that SMT systems often make errors in translation of coordinations, which means that by disregarding them in this part of Depfix, many errors remain uncorrected.

And last, as in most parts of Depfix, a better treatment of named entities would be beneficial. Our approach is not to fix any words that seem to be named entities (the form contains an uppercase letter), as the number of successful corrections on them was similar to the number of incorrections.

Chapter 8

Evaluation

This chapter evaluates the Depfix system. We first describe the approach that we took to the evaluations in Section 8.1. Section 8.2 contains both a manual and an automatic evaluation of the final Depfix system, the automatic being performed on a set of 13 MT systems. Section 8.3 evaluates the performance of the individual fixes of Depfix on Moses SMT system. In Section 8.4, we evaluate our reimplementation if the MST Parser that we use in Depfix, including an evaluation of the adaptations of the parser that we introduced.

8.1 Evaluation Methodology

We describe the datasets that we used to evaluate the performance of Depfix and detail both the automatic and manual evaluation methods.

8.1.1 Evaluation Datasets

We evaluated Depfix on several datasets – WMT_{10} (Callison-Burch et al., 2010), which is our development dataset, WMT_{11} (Callison-Burch et al., 2011) and WMT_{12} (Callison-Burch et al., 2012). The data in the datasets are taken from the news domain and were translated by human translators. The datasets were used for evaluation in the Translation task of the Workshop on Statistical Machine Translation (WMT).

Human post-editations

The improvements brought by Depfix measured in automatic metrics scores are typically very low. We believe that this is partly caused by the reference translations, which are often very different from the source sentences and thus very different from the outputs of the SMT systems; sometimes the reference translations are even erroneous. In such cases, the outputs of Depfix often do not get any closer to of further from the reference translation, as measured by the automatic metric, and we are then unable to evaluate many of the changes performed by Depfix properly.

We therefore created an alternative reference translation of the WMT_{11} dataset by post-editing the Moses translations, i.e. performing human post-editation instead of automatic post-editation.

Dataset	Senteces
WMT_{10}	2489
WMT_{10}'	2034
WMT_{11}	3003
HPE_{11}	6006
WMT_{12}	3003

Table 8.1: Sizes of the evaluation datasets.

We divided the dataset into six parts of equal sizes. Each of these parts, consisting of the source sentences and their Moses translations (but without the reference translations), was then given to two of our six post-editors, who were instructed to transform the outputs of Moses into correct translations by a minimal number of edits. This resulted in a set of 12 post-editations of the 6 parts, i.e. 2 for each part.

However, we discovered that the work of the post-editors was often far from perfect, leaving many uncorrected errors and even occasionally introducing new ones. Therefore, we redistributed the 12 post-edited set back to the post-editors, this time with the task to post-edit their colleagues' outputs.

The final dataset, further referred to as HPE_{11}, consists of the reduplicated WMT_{11} source sentences, reduplicated outputs of Moses, and a reference made by concatenating the 12 double-post-edited parts.

Overview

We detail the sizes of the datasets in Table 8.1. WMT_{10}' is a subset of WMT_{10} that was used to evaluate the final systems in WMT 2011 (Callison-Burch et al., 2011). The average sentence length is 21 words for the source sentences, 17 words for their Moses translations, and 18 words for both the reference translations and the human post-editations.

8.1.2 Manual evaluation

Manual evaluation of Depfix performance is performed by two independent annotators, who task is to annotate every sentence in a dataset that was changed by Depfix as a correction, an incorrection, or an indefinite change. The indefinite change can have several meanings, such as a change with no effect on the translation quality, a pair of a correction and an incorrection in one sentence, or a translation that is completely incomprehensible both before and after the change; for simplicity, we decided not to distinguish these explicitly, as we expect the agreement on these subtype to be very low.

For each sentence, the annotator can see the source sentence, the reference translation, and a pair of candidate translations, one being the output of Moses and the other being its post-editation. The differences in the translations are highlighted, as the sentences are often long but contain only a few changes, which was found to make the task much more difficult for the annotators. The candidate translations are given in a random order, so that the annotator does not know which sentence is post-edited by Depfix and which one is not. Moreover, the

quadruples of the sentences are also given in a random order to account for the inhomogeneities of the datasets.

The task of the annotator is to mark the better one of the two candidate translations if possible. If this is the Depfix output, we assume that the change performed by Depfix was a corrections, if it is the unchanged output of Moses, we assume that Depfix performed an incorrection. If the annotator does not mark any sentence as being better than the other, the Depfix change on the sentence is indefinite.

8.1.3 Automatic Evaluation

For automatic evaluations, we use NIST (Doddington, 2002) and BLEU (Papineni et al., 2002) translation quality metrics. The metrics are quite similar. They both measure the quality of the translation by comparing the individual words and word n-grams in a reference translation and the translation produced by the MT system. They then compute a score based on the level of match between the source and target, and the differences in lengths of the translations.

BLEU is the de-facto standard metric for SMT, and many SMT systems are tuned to maximize their BLEU score. However, NIST proved to be more fine-grained and better correlated with human judgement when evaluating the improvements of individual parts of Depfix, which usually constitute only a tiny change to the overall score. Therefore, we use BLEU to evaluate only the whole system, not its individual parts.

8.1.4 Development Manual Evaluation

During the development of Depfix, it was naturally necessary to continually evaluate the modifications of the Depfix setup. We discuss here the possibilities of an evaluation to be used, and the final approach that we took.

When evaluating small changes to the Depfix setup, automatic evaluation, described in Section 8.1.3, can only serve as an indication of a probable increase or decrease in performance, as both the small size of the development dataset (WMT_{10}) and the small number of differences in Depfix outputs caused by a small change to its setup make them very unreliable. We used both NIST and BLEU for such indication, with NIST proving to be better, but BLEU serving as a secondary indication – for example, if we observed a small increase in NIST but a big decrease in BLEU (which is quite the opposite of our usual observations), we tried to evaluate several similar setups as well to try to find out which metric to trust more in such case.

Unfortunately, a proper manual evaluation described in Section 8.1.2 is also unsuitable for everyday evaluations, as it would require to have a pool of annotators available all the time, and it is also not straightforward to link its results to the individual parts of Depfix that are responsible for them. We performed such evaluation only a few times during the development of Depfix; the results of these evaluations can be found in (Mareček et al., 2011) and (Rosa et al., 2012b).

Therefore, quick manual evaluations on small randomly selected subsets of the dataset where used throughout the development process as the primary

Annotator	Evaluated	Changed	-	+	0	Precision	Recall
D	300	160	36	94	30	58.8%	31.3%
V	1050	579	116	336	127	58.0%	32.0%
Total	1350	739	152	430	157	58.2%	31.9%

Table 8.2: Manual evaluation of Depfix performance on a subset of 1350 sentences from WMT_{12}

indication. They were performed by the developer of Depfix himself, as they provide immediate feedback on the exact effect of the modifications to the setup of Depfix. Before finalizing each of the modifications, such evaluation was performed on a larger part of the dataset for higher reliability.

8.2 Evaluation of the Whole Depfix System

We evaluated the whole Depfix system both manually (Section 8.2.1 and automatically (Section 8.2.2).

8.2.1 Manual Evaluation

We performed manual evaluation of the performance of Depfix on the Moses translations of a randomly selected subset of WMT_{12}. A total of 739 changed sentences were evaluated jointly by two annotators; a subset of 84 sentences was annotated by both of the annotators for the sake of evaluation of their agreement.

The results of the evaluation are shown in Table 8.2. The table shows the number of sentences that were selected for evaluation, the size of the subset of these sentences that were changed by Depfix, which were then manually evaluated, and their division into the three categories of corrections (+), incorrections (-), and indefinite (0) changes.

The following formulas define the precision (8.1) and recall (8.2) of Depfix, which are also listed in the table:

$$precision = \frac{corrected}{changed} \tag{8.1}$$

$$recall = \frac{corrected}{evaluated} \tag{8.2}$$

The results show that Depfix post-editing improves the quality of the outputs of Moses. The majority of the changes are positive, the precision of Depfix reaching 58%. Moreover, the recall of Depfix is around 32%, i.e. it is able to correct an error in approximately every third sentence produced by Moses.

The inter-annotator agreement was measured on a subset of 150 sentences from the WMT_{12} dataset, out of which 84 were changed by Depfix. These sentences were annotated by both of the annotators, and their annotations were then compared. The results are shown in Table 8.3.

If we disregard sentences that at least one of the annotators was unable to evaluate ("indefinite"), the inter-annotator agreement reaches 93%, as there are only 4 sentences out of 56 where the annotators took the exactly opposite decisions.

D/V	improved	worsened	indefinite
improved	36	2	8
worsened	2	16	2
indefinite	10	0	8

Table 8.3: Inter-annotator agreement matrix of manual evaluation of Depfix.

If we take all of the 84 sentences into account, i.e. also requiring the annotators to agree on whether the difference in translation quality is distinguishable, their agreement drops to 71% because of 20 sentences where one of the annotators was able to denote one of the translations as being better while the other could not.

We believe this level of agreement to be high enough for the results of the manual evaluation to be considered trustworthy. The size of the paralelly annotated dataset is rather small, but the resulting values of inter-annotator agreement are consistent with the values we measured on much larger datasets in previous evaluations of Depfix by the same annotators, such as 500 parallely annotated sentences in (Rosa et al., 2012b).

8.2.2 Automatic Evaluation

We evaluated the performance of Depfix on outputs of many English-to-Czech MT systems, which took part in the WMT shared tasks (Callison-Burch et al., 2010, 2011, 2012). All the data were acquired from the websites of the workshops, which be found on `http://www.statmt.org/`; the outputs of the systems thus come from the respective years of the workshops. We also evaluated Depfix on newer outputs of Moses and Google Translate, coming from autumn 2012. We computed NIST and BLEU scores for each of the setups, and evaluated statistical significance of the improvement in BLEU, using bootstrap resampling.

The results are shown in Table 8.4. For each system and dataset, the table lists the number of sentences changed by Depfix, and NIST and BLEU scores for the output of the system both before ("base") and after ("Depfix") being processed by Depfix; for better readability, the scores are multiplied by 100. Most of the differences in BLEU are statistically significant on the 0.05 level of significance; those that are not are marked by "!". The table also includes averages of scores from the three years of the WMT. The averages do not include the scores of current versions of Moses and Google Translate (first and last system in the table) as they are newer than the other systems; the outputs of CU Bojar and Google Translate that come from the WMT datasets are included in the average.

It should be noted that, as we have shown in (Mareček et al., 2011), the automatic evaluation does not always correlate with human evaluation, potentially even inverting the sign of the change. However, we do not have a capacity for large-scale manual evaluation, and the automatic evaluation thus has to be used as an approximation instead.

The results clearly show that, although tuned for outputs of Moses, Depfix can improve the results of the majority of existing English-to-Czech MT systems; for one of the setups (SFU 2010), the improvement is even above 1 BLEU point. Only the outputs of TectoMT are consistently made worse by Depfix processing; however, this is to be expected, as the design of both TectoMT and Depfix

SMT system	Data-set	Chgd sents	NIST score x 100			BLEU score x 100		
			base	Depfix	Δ	base	Depfix	Δ
Moses	WMT_{10}	1473	544.24	559.98	+15.73	15.66	16.08	+0.42
	WMT_{11}	1753	572.56	582.36	+9.79	16.39	16.61	+0.22
	HPE_{11}	3506	973.84	991.33	+17.48	43.88	44.71	+0.82
	WMT_{12}	1669	526.27	533.63	+7.35	13.81	13.85	! +0.04
CU Bojar	WMT_{10}'	1252	547.28	557.32	+10.03	15.85	16.19	+0.33
	WMT_{11}	1914	568.79	582.60	+13.80	16.35	16.83	+0.47
	WMT_{12}	1658	532.10	538.13	+6.02	14.19	14.26	! +0.07
CU Tamchyna	WMT_{11}	1902	561.66	575.64	+13.98	15.86	16.32	+0.46
	WMT_{12}	1624	530.12	535.73	+5.61	14.01	14.04	! +0.02
CU TectoMT	WMT_{10}'	671	524.76	523.66	-1.10	12.83	12.76	! -0.07
	WMT_{11}	1026	553.71	551.46	-2.24	13.60	13.50	-0.10
	WMT_{12}	942	522.78	522.54	-0.24	11.99	11.97	! -0.02
CU Zeman	WMT_{10}'	1425	496.08	510.13	+14.05	12.33	12.95	+0.61
	WMT_{11}	2351	539.91	557.91	+18.00	14.08	14.81	+0.73
	WMT_{12}	2216	494.36	505.79	+11.43	12.10	12.44	+0.34
UEDIN	WMT_{10}'	1392	543.38	560.31	+16.92	15.91	16.69	+0.78
	WMT_{11}	1973	587.66	602.93	+15.27	17.30	17.94	+0.64
	WMT_{12}	1806	560.32	569.16	+8.84	15.54	15.78	+0.23
JHU	WMT_{11}	2181	588.47	600.73	+12.25	16.92	17.35	+0.42
	WMT_{12}	2257	514.39	524.41	+10.02	13.06	13.38	+0.32
SFU	WMT_{10}'	1581	468.39	491.34	+22.95	11.43	12.49	+1.05
	WMT_{12}	2074	494.81	506.58	+11.77	12.03	12.45	+0.41
DCU	WMT_{10}'	1519	495.13	514.34	+19.20	13.36	13.95	+0.59
Potsdam	WMT_{10}'	1326	474.16	490.84	+16.68	12.34	12.92	+0.58
KOC	WMT_{10}'	1637	453.07	471.84	+18.77	11.74	12.34	+0.59
EuroTrans	WMT_{10}'	1289	443.53	457.50	+13.97	10.10	10.46	+0.35
	WMT_{11}	1809	446.44	451.46	+5.01	9.30	9.51	+0.21
	WMT_{12}	1824	428.09	436.50	+8.41	8.79	8.95	+0.15
Bing	WMT_{10}'	1496	465.18	483.41	+18.23	11.81	12.59	+0.78
	WMT_{12}	1955	542.38	551.29	+8.91	13.86	14.24	+0.37
Google Translate	WMT_{10}'	1312	555.88	568.16	+12.27	16.57	17.16	+0.59
	WMT_{11}	1894	621.42	632.23	+10.81	19.73	19.97	+0.23
	WMT_{12}	1737	572.48	576.23	+3.74	16.22	16.22	! 0.00
Google Translate 2012	WMT_{10}	1516	582.59	590.30	+7.71	17.66	18.01	+0.35
	WMT_{11}	1796	624.77	630.48	+5.71	19.37	19.44	! +0.06
	WMT_{12}	1762	572.75	576.66	+3.91	16.22	16.24	! +0.01
Average	WMT_{10}'	1355	496.99	511.71	+14.72	13.12	13.68	+0.56
	WMT_{11}	1881	558.51	569.37	+10.86	15.39	15.78	+0.38
	WMT_{12}	1809	519.18	526.64	+7.45	13.18	13.37	+0.19

Table 8.4: Automatic evaluation of the whole Depfix system.

are similar, often even sharing the same analysis and generation tools that are available in Treex.

Sadly, for the best performing system, Google Translate, the improvements by Depfix are often very small and statistically insignificant. We believe that this is caused by the fact that the quality of Google Translate outputs is already very high and there is little remaining to be fixed. Moreover, Google Translate seems to use a powerful language model, which minimizes the amount of grammatical errors in its output; the downside of such approach is that it tends to generate fluent and grammatical outputs that have a different meaning than the source sentence, but this usually cannot be detected by Depfix.

A comparison of the average number of changed sentences with the sizes of the dataset shows that on average, Depfix changes over 60% of the sentences, with an average improvement of about 0.3 or 0.4 BLEU point. The fact that the improvement is lower on the later dataset may suggest that the MT systems are improving over time, making less error that can be fixed by Depfix. However, as the absolute scores for the WMT_{12} dataset are lower than the scores for WMT_{11} dataset, it might also suggest that either some datasets are harder to translate, or that the quality of the reference translations in some datasets is lower.

The absolute scores from the evaluation on the HPE_{11} dataset are naturally much higher, as the post-editations are necessarily much closer to the output of Moses than the reference translation. However, the increase of the absolute scores and of the improvement by Depfix are proportionate, which presumably confirms consistence and sanity of the results reported.

Please note that all of the scores listed in the table are case-insensitive and thus do not account for Depfix corrections that only change the casing of the words.

8.3 Evaluation of Individual Parts of Depfix

For an even finer insight on the performance of individual parts of Depfix, we evaluate the utility of each of the individual parts of Depfix, i.e. each fix rule an each type of a statistical fix. As it is hard to evaluate each part of Depfix intrinsically, we estimate the performance of a part by removing it from the Depfix system, and comparing the outputs of such modified system to the outputs of the whole system.

We use two indicators of a performance of a part of Depfix: the number of sentences affected by the removal of the part, listed in Table 8.5, and the negative decrease of NIST after removing the part, listed in Table 8.6. As was already explained in Section 8.1, the difference in NIST score has to be taken only as an approximate indication of the performance of the rule, especially in case of score differences very close to 0.

As the standard NIST score is case-insensitive, the score differences shown for "Source-aware truecasing" and "Sentence-initial capitalization" were computed using its case-sensitive variant as implemented in the MTrics evaluation tool (Kos, 2008). This is marked by asterisks in the table.

The tables show the largest numbers of changes are made by three of the agreement fixing rules, namely "Noun - adjective agreement" (Section 6.3.6), "Preposition - noun agreement" (Section 6.3.1) and "Subject - past participle

Fix	Affected sentences in dataset		
	WMT_{10}	WMT_{11}	WMT_{12}
M-layer analysis fixes			
Tokenization projection	179	162	99
Fixing morphological number of nouns	174	198	161
Adding missing alignment links	37	56	40
M-layer translation fixes			
Source-aware truecasing	197	309	277
Vocalization of prepositions	42	33	32
Sentence-initial capitalization	60	42	56
A-layer analysis fixes			
Fixing reflexive tantum	14	13	13
Rehanging children of auxiliary verbs	18	6	10
Prepositional morphological case	23	31	27
Preposition without children	17	23	19
A-layer agreement fixes			
Preposition - noun agreement	304	336	304
Subject - predicate agreement	67	84	50
Subject - past participle agreement	273	283	283
Passive - auxiliary 'be' agreement	23	19	28
Subject - auxiliary 'be' agreement	11	6	5
Noun - adjective agreement	416	496	407
A-layer translation fixes			
Missing reflexive verbs	34	45	63
Translation of 'by'	64	92	83
Translation of 'of'	65	95	76
Translation of passive voice	17	24	26
Translation of possessive nouns	113	68	73
Translation of present continuous	31	18	31
Subject morphological case	120	159	144
Subject categories projection	18	24	36
T-layer rule-based fixes			
Negation translation	80	102	104
Subject personal pronouns dropping	73	79	94
Tense translation	155	186	216
T-layer statistical fixes, general model			
Changing the argument case	261	268	268
Changing the preposition	19	31	33
T-layer statistical fixes, specific model			
Changing the preposition	20	37	26
Adding a new preposition	12	27	25

Table 8.5: Impact of the individual fixes – number of affected sentences.

Fix	NIST increase on dataset, x100			
	WMT_{10}	WMT_{11}	WMT_{12}	HPE_{11}
M-layer analysis fixes				
Tokenization projection	4.49	3.80	0.99	6.84
Fixing morphological number of nouns	0.62	0.70	0.67	0.04
Adding missing alignment links	0.07	-0.08	0.01	-0.19
M-layer translation fixes				
Source-aware truecasing*	4.30	4.80	4.00	4.00
Vocalization of prepositions	0.21	0.14	0.20	-0.02
Sentence-initial capitalization*	0.20	0.20	-0.10	0.00
A-layer analysis fixes				
Fixing reflexive tantum	-0.01	0.02	0.21	0.11
Rehanging children of auxiliary verbs	-0.01	0.01	-0.09	0.03
Prepositional morphological case	-0.04	0.32	0.16	0.32
Preposition without children	0.22	-0.10	0.05	-0.10
A-layer agreement fixes				
Preposition - noun agreement	3.37	1.73	1.17	4.40
Subject - predicate agreement	-0.11	-0.50	-0.11	-0.71
Subject - past participle agreement	0.48	0.60	0.03	2.11
Passive - auxiliary 'be' agreement	0.08	0.05	0.04	0.15
Subject - auxiliary 'be' agreement	0.01	-0.05	0.01	-0.02
Noun - adjective agreement	1.82	1.82	1.46	3.96
A-layer translation fixes				
Missing reflexive verbs	0.21	0.00	0.12	0.44
Translation of 'by'	-0.02	-0.27	0.09	0.30
Translation of 'of'	0.52	0.56	0.54	0.22
Translation of passive voice	0.02	-0.07	-0.09	0.14
Translation of possessive nouns	1.63	0.44	0.47	1.34
Translation of present continuous	0.40	-0.01	0.06	-0.04
Subject morphological case	0.42	0.47	0.68	0.11
Subject categories projection	0.16	0.10	0.26	0.08
T-layer rule-based fixes				
Negation translation	0.08	0.26	0.13	0.27
Subject personal pronouns dropping	0.51	0.26	-0.03	0.06
Tense translation	0.02	0.16	0.06	-1.03
T-layer statistical fixes, general model				
Changing the argument case	0.67	-0.40	-0.08	-0.26
Changing the preposition	0.23	0.11	0.48	0.11
T-layer statistical fixes, specific model				
Changing the preposition	0.22	0.00	0.03	-0.06
Adding a new preposition	0.02	0.05	0.02	-0.49

Table 8.6: Impact of the individual fixes – NIST scores.

agreement" (Section 6.3.3). As could be expected, these rules bring the highest increases of NIST at the same time. However, the benefit of "Subject - past participle agreement" (Section 6.3.3) is only marked when HPE_{11} is used as the reference.

One of the largest gains in NIST score is thanks to "Tokenization projection", which at the same time fixes one of the least serious errors. This is something to be considered when automatically evaluating some data by metrics such as BLEU or NIST. Another of the largest gains is from "Source-aware truecasing", which also fixes errors that are typically not very serious. However, please note that the standard MT evaluation metrics are case-insensitive, which does not motivate the researchers to produce correctly cased outputs.

Changing the argument morphological case by the statistical fixing (Section 7.4) also changes a large number of sentences. However, it also leads to an accordingly large decrease of NIST score on all but the development dataset. Generally, the results suggest that the statistical corrections are over-tuned for the development data (WMT_{10}), as they show very little increases or decreases on the other datasets in most cases.

The largest decrease of NIST is reported for the "Subject - predicate agreement" (Section 6.3.2), although it only changes an average number of sentences. This suggests that the rule is all wrong, with most or all of its changes to the sentences being incorrections.

"Tense translation" (Section 7.3.3) gets penalized a lot when HPE_{11} is taken as the reference translation. This may indicate that it makes inappropriate changes of tenses, probably due to a simplistic approach to mapping English tenses to Czech tenses. However, the result could also mean that the post-editors did not pay enough attention to errors in verb tenses, as these often do not stand out as much as errors in agreement or in different lexical choice – they both approximately correctly convey the meaning of the source sentence and do not make the target sentence ungrammatical.

On the other hand, the scores of agreement fixes are much higher on HPE_{11} than on the other datasets. This is probably because the post-editors were told to try to perform minimal necessary post-edits; thus, in case of agreement violation, they would typically only correct the agreement if no further errors were present. On the other hand, the WMT reference translation might use the same morphological categories as Depfix enforces, but with different lexical choices, thus not awarding any points to Depfix for many corrections. The post-editor typically would not change the lexical choice for a synonymous one, he would only do such a change if the original lexical choice were unacceptable. It seems that this is one thing that Depfix can already do similarly to people.

Several of the rules, such as "Subject - auxiliary 'be' agreement" (Section 6.3.5) or "Translation of passive voice" (Section 6.4.4), have a small effect on the translations, measured both in numbers of changed sentences and in differences in NIST score. However, during the development of Depfix, their effect seemed to be positive, although small, and we therefore included them in the final system nevertheless.

Parser setup	UAS	LAS
MSTA parser	83.82%	77.08%
base, PDT	84.05%	77.34%

Table 8.7: Evaluation of the base parser

8.4 Evaluation of the Parser and Labeller

In this section, we evaluate our adaptations of the MST Parser (McDonald et al., 2005), which we described in Chapter 5. In Section 8.4.1, we evaluate the base monolingual parser and labeller. We then follow by Section 8.4.2, which evaluates the various adaptations of the parser for Depfix. We already presented evaluations of some of the setups in (Rosa et al., 2012a) and (Rosa and Mareček, 2012).

8.4.1 The Base Parser

For the base monolingual parser, described in Section 5.3, we use the Prague Dependency Treebank 2.0 Hajič et al. (2006) for training and testing; we refer to the parser as "base, PDT". We use 68500 sentences as training data, 4500 sentences as development data and 4500 sentences as test data. We compare its performance to the MSTA parser, which was described in Section 5.2. The results are presented in Table 8.7. The table shows two scores for each of the parser setups. The unlabelled attachment score (UAS) is the percentage of nodes that got their parent node assigned correctly, i.e. it only evaluates the performance of the unlabelled parser (Section 5.3.1). The labelled attachment score is then the percentage of nodes that both get their parent assigned correctly and the edge to the parent was labelled by the correct analytical function (see Section 5.1), i.e. it jointly evaluates both the unlabelled parser and the second-stage labeller (Section 5.3.2).

The results show that we have been successful in reimplementing both the parser and the labeller. As was described in Section 5.3, our implementation of the MST Parser is simplified in several aspects. We believe that we managed to achieve better performance than MSTA parser thanks to our feature set, which was carefully tuned for parsing of Czech sentences.

The performance of the labeller was evaluated in (Rosa and Mareček, 2012), coming to a conclusion that it reaches state-of-the-art performance for Czech.

8.4.2 The Modified Parser

We used the Prague Czech-English Dependency Treebank (PCEDT) 2.0 (Hajič et al., 2012) as the training data for our parser. PCEDT 2.0 is a parallel treebank created from the Penn Treebank (Marcus et al., 1993) and its translation into Czech by human translators. The dependency trees on the English side were converted from the manually annotated phrase-structure trees in Penn Treebank, the Czech trees were created automatically using the MSTA parser. Words of the Czech and English sentences were aligned by GIZA++ (Och and Ney, 2003).

We evaluate the extensions to the parser using NIST, which we used during the development as the main indicator of performance; we sometimes used BLEU as well, but it usually differed only very slightly, which we believe to be insignificant.

Parser setup	NIST improvement x100			
	WMT_{10}	WMT_{11}	HPE_{11}	WMT_{12}
MSTA parser	12.84	7.85	12.04	5.79
PCEDT, base	12.00	7.96	12.12	6.32
+ worsening the training data	12.87	8.34	13.01	6.93
+ adding parallel information	14.93	9.44	15.15	7.45
+ trimming the lemmas	14.98	9.92	15.51	7.16
+ adding large-scale information	14.91	9.37	16.25	7.44
+ manually boosting feature weights	15.73	9.79	17.48	7.35

Table 8.8: Evaluation of the parser modifications as improvements of Depfix performance, measured in NIST.

We were often unable to quickly compare the setups of the parser by manual evaluation, as most of the differences in Depfix outputs when using two different parsers were rather random, not following any observable pattern.

We do not use the attachment scores for evaluation, as we can only compute them on the test data which is taken from PCEDT or PDT, but this tells us little about the performance the parser will have on SMT outputs.

We evaluate the steps that lead to our final setup in Table 8.8. We take our base monolingual parser, trained on PCEDT, for the baseline; for comparison, the performance of MSTA parseris also listed. We then continue by additively modifying the parser, performing adaptations that were described Chapter 5; thus, the last line in the table details the performance of our final setup.

1. the base monolingual parser (Section 5.3), this time on PCEDT
2. worsening the training data (Section 5.4)
3. adding parallel information (Section 5.5)
4. trimming the lemmas (Section 4.2.4)
5. adding large-scale information (Section 5.6)
6. manually boosting feature weights (Section 5.5.3)

The benefit of worsening the training data and adding parallel information was already confirmed by manual evaluation in (Rosa et al., 2012a).

The evaluation confirms that all of the modifications of the parser seem to have a positive effect on Depfix performance, leading to an overall increase of the NIST score between 1 and 3 hundredths of a point for the WMT datasets, and even about 0.04 for the HPE_{11} dataset.

The largest gain in NIST score is achieved by adding the aligned features. This is probably caused by the fact that this modification directly provides additional information about a possibly correct output of the parser, while the other extensions provide only indirect indications.

Worsening the training data did not bring such a large improvement, but it is far from negligible either and is consistently positive. As the approach we took in Section 5.4 is rather simple, we believe that further research of making the

training data even more similar to the SMT outputs could bring an even larger improvement.

Manually boosting the weight of the aligned edge feature led to a similar improvement – not immense, but respectful and usually positive. Considering the fact that we probably took the simplest and most coarse approach possible, we believe that much more could be gained by this approach.

The effect of adding large-scale information is not as clearly positive as the other extensions, although the largest decrease is on WMT_{11} dataset and can be probably disproved by the opposite effect on the HPE_{11} dataset. Still, as we already noted in Section 5.6, we believe that more research in this area is necessary.

The effect of trimming the lemmas is not impressive, but we took that step mainly because the large-scale information features perform better when the lemmas are trimmed; thus, we take the small increase in NIST score on most datasets after trimming the lemmas as a positive by-product.

Chapter 9

Conclusion

In this thesis, we presented a complex automatic post-editing system, Depfix, which is able to correct various types of errors in English-to-Czech statistical machine translation.

Statistical machine translation has become the state-of-the-art approach for machine translation, despite (or thanks to?) the fact that it employs little linguistic knowledge, which it replaces by machine learning on large-scale data sets. However, this approach leads to many grammatical errors in the outputs, which lower the quality of the translation.

Depfix is a system which brings linguistic knowledge back into machine translation. However, as incorporating linguistic knowledge directly into statistical translation systems seems to be a rather difficult tasks, Depfix takes a different approach. We do not try to directly modify the machine translation system. Depfix only takes its output, a Czech sentence, together with the source English sentence, performs a deep linguistic analysis of the sentences, and tries to find and correct various types of errors in the translation, with a focus on errors that are severe, common and/or easy to fix.

Depfix relies on a range of existing natural language processing tools, such as taggers and parsers, which allow it to explore the structure of each sentence on several layers of linguistic abstraction. We use the Treex framework, which incorporates many such tools, especially for Czech and English. However, we had to adapt the Czech language analysis pipeline for our task, as most of the tools show a decreased performance on outputs of statistical machine translation. This is especially due to the errors present in the sentences, substantially different from errors found in texts produced by speakers of the language, for analysis of which the tools were primarily designed. Our main contribution in tool adaptation is the reimplementation of the Maximum spanning tree parser, with several modifications that significantly improve its accuracy on outputs of English-to-Czech statistical machine translation, such as the incorporation of parallel features into the feature set.

The post-editing itself is performed by numerous blocks, called fixes, that use the linguistic analysis to identify various, usually grammatical, errors. A prominent error type that Depfix focuses on are errors in morphological agreement. Morphological agreement is an important phenomenon of Czech language, governed by a set of strict and explicit grammatical rules, but is often violated in statistical machine translation outputs. However, with proper

linguistic analysis, it is very easy to spot and fix, the correct word form being generated by a morphological generator.

The majority of the fixes are rule-based (although, due to the unreliability of the analyses and to some extent also to vagueness in Czech grammar, they often have to employ heuristics to increase their precision). However, Depfix also contains a fully statistical component, which tries to correct valency errors. This is a challenging task both for statistical translation systems and for rule-based post-editing. The level of analysis that typical SMT systems employ is too shallow to capture valency if the valency members are not adjacent. A rule-based post-editing system can explore the sentence structure easily, but the rules that govern valency are probably best described by valency lexicons, which are currently only being developed and will probably not be available in the near future in a form and extent suitable for employment in Depfix. Our approach therefore is to build a set of valency models from a large-scale parallel corpus, automatically analyzed up to the deep-syntactical layer, which enables us to combine extensive linguistic knowledge with statistical evidence to both detect and correct a number of errors in valency.

We performed a throughout evaluation of Depfix, both automatic and manual, evaluating both the whole system and its individual parts. The evaluation proved that post-editing of statistical machine translations by Depfix significantly improves their quality.

We believe the research path of employing linguistic knowledge in automatic post-editing of statistical machine translation, which we pioneer in our work, to be a reasonable direction to follow in future.

As shown by our modified parser, the performance of Depfix can be significantly improved by adapting the analysis tools for the specific task of processing erroneous data when having their human-provided English translation at disposal. We believe that adapting the morphological tagger in a similar way would bring similar improvements; moreover, it is possible that we still do not exploit the information from the English sentence to the maximum possible extent.

In the rule-based fixes, we often resorted to simplifications and heuristics that proved to perform well enough to constitute a positive effect, but there are many places where the rules could still be fine-tuned to achieve a higher fixing precision and/or recall; such places are often explicitly mentioned in the descriptions of the rules. Moreover, many common errors remain unaddressed in our work, as we tried to focus only on the most serious and the most frequent ones, but at the same time we avoided fixing errors that our approach did not seem to apply well to. Some of them might be fixed by rules quite similar to the ones that we already use, while other will require a new, different approach.

And last, we are convinced that our methodology is by no means limited to English-to-Czech translation. For many languages, there exist high-performance processing tools, able to provide similar linguistic analyses as the tools we use, but relevant for the respective languages. These could be used to provide ground for developing a similar rule-based and/or statistical post-editing system, which we expect would also be able to improve the quality of machine translation between the respective languages.

Bibliography

Marianne Adams. From old french to the theory of pro-drop. *Natural Language & Linguistic Theory*, 5(1):1–32, 1987.

Tania Avgustinova and Karel Oliva. On the nature of the wackernagel position in czech. *Formale Slavistik*, 7:25–57, 1997.

H. Béchara, Y. Ma, and J. van Genabith. Statistical post-editing for a statistical mt system. *MT Summit XIII*, pages 308–315, 2011.

O. Bojar, B. Jawaid, and A. Kamran. Probes in a taxonomy of factored phrase-based models. In *Proceedings of the Seventh Workshop on Statistical Machine Translation, Montreal, Canada, June. Association for Computational Linguistics. Submitted*, 2012a.

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC2012*, Istanbul, Turkey, May 2012b. ELRA, European Language Resources Association. In print.

Ondřej Bojar. Analyzing Error Types in English-Czech Machine Translation. *Prague Bulletin of Mathematical Linguistics*, 95:63–76, March 2011. ISSN 0032-6585.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=176313.176316`.

Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

David Burkett, John Blitzer, and Dan Klein. Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135. Association for Computational Linguistics, 2010.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. Findings of the 2010 joint workshop on statistical

machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.statmt.org/wmt10/pdf/WMT03.pdf.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2103.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3102.

X. Carreras. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, volume 7, pages 957–961, 2007.

Wenliang Chen, Jun'ichi Kazama, and Kentaro Torisawa. Bitext dependency parsing with bilingual subtree constraints. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 21–29. Association for Computational Linguistics, 2010.

Wenliang Chen, Jun'ichi Kazama, Min Zhang, Yoshimasa Tsuruoka, Yujie Zhang, Yiou Wang, Kentaro Torisawa, and Haizhou Li. Smt helps bitext dependency parsing. In *EMNLP*, pages 73–83. ACL, 2011. ISBN 978-1-937284-11-4. URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2011.html#ChenKZTZWTL11.

Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14(1396-1400):270, 1965.

Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. A statistical parser for Czech. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 505–512, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. ISBN 1-55860-609-3. doi: http://dx.doi.org/10.3115/1034678.1034754. URL http://dx.doi.org/10.3115/1034678.1034754.

K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991, 2003.

G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240, 1967.

Andreas Eisele, Christian Federmann, Hans Uszkoreit, Hervé Saint-Amand, Martin Kay, Michael Jellinghaus, Sabine Hunsicker, Teresa Herrmann, and Yu Chen. Hybrid machine translation architectures within and beyond the euromatrix project. In *Proceedings of the 12th annual conference of the European Association for Machine Translation (EAMT 2008)*, pages 27–34, 2008.

Jason M Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics, 1996.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

Mark Fishel, Ondřej Bojar, and Maja Popović. Terra: a collection of translation error-annotated corpora. In *Proceedings of the 8th International Conference on Language Resources and Evaluation LREC'2012*, pages 7–14, Istanbul, Turkey, 2012.

Jennifer Foster, Joachim Wagner, and Josef Van Genabith. Adapting a WSJ-trained parser to grammatically noisy text. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 221–224. Association for Computational Linguistics, 2008.

Kevin Gimpel and Shay Cohen. Discriminative online algorithms for sequence labeling- a comparative study, 2007.

Jan Hajič. *Disambiguation of rich inflection: computational morphology of Czech*. Karolinum, 2004.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. Prague dependency treebank 2.0, 2006.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics, 2009.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Announcing prague czech-english dependency treebank

2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, İstanbul, Turkey, 2012. ELRA, European Language Resources Association. ISBN 978-2-9517408-7-7.

Jan Hajič. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Prague Karolinum, Charles University Press, 1998.

Martin Wittorff Haulrich. *Data-Driven Bitext Dependency Parsing and Alignment*. PhD thesis, Copenhagen Business School, 2012.

Liang Huang, Wenbin Jiang, and Qun Liu. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1222–1231. Association for Computational Linguistics, 2009.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11:311–325, September 2005. ISSN 1351-3249. doi: 10.1017/S1351324905003840. URL `http://dl.acm.org/citation.cfm?id=1088141.1088144`.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395, Barcelona, Spain, 2004.

Philipp Koehn. A process study of computer-aided translation. *Machine Translation*, 23:241–263, 2009a. ISSN 0922-6567. doi: 10.1007/s10590-010-9076-3. URL `http://dx.doi.org/10.1007/s10590-010-9076-3`.

Philipp Koehn. A web-based interactive computer aided translation tool. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 17–20, Suntec, Singapore, August 2009b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P09/P09-4005`.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-2045`.

Kamil Kos. Adaptation of new machine translation metrics for Czech. Bachelor's thesis, Charles University in Prague, 2008.

Philippe Langlais, George Foster, and Guy Lapalme. Transtype: a computer-aided translation typing system. In *Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems - Volume 5*, NAACL-ANLP-EMTS '00, pages 46–51, Stroudsburg, PA, USA, 2000.

Association for Computational Linguistics. doi: 10.3115/1117586.1117593. URL http://dx.doi.org/10.3115/1117586.1117593.

Martina Lindseth. *Null-subject properties of Slavic languages: with special reference to Russian, Czech and Sorbian*, volume 361. Otto Sagner, 1998.

Chi-kiu Lo and Dekai Wu. MEANT: an inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 220–229, 2011.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.*, 19: 313–330, June 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972475.

David Mareček, Rudolf Rosa, Petra Galuščáková, and Ondřej Bojar. Two-step translation with grammatical post-processing. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 426–432. Association for Computational Linguistics, 2011.

Peter H Matthews. Syntax. cambridge textbooks in linguistics. *Cambridge University Press*, 69:75, 1981.

R. McDonald and F. Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, volume 6, pages 81–88, 2006.

R. McDonald, K. Crammer, and F. Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98. Association for Computational Linguistics, 2005.

Ryan McDonald. *Discriminative learning and spanning tree algorithms for dependency parsing*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2006. AAI3225503.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, 2005.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 216–220, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1596276.1596317.

Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics, 2011.

Marjorie McShane. The ellipsis of accusative direct objects in russian, polish and czech. *Journal of Slavic linguistics*, 7(1):45–88, 1999.

Gereon Muller. Pro-drop and impoverishment. In *Form, structure, and grammar: A Festschrift presented to Günther Grewendorf on occasion of his 60th birthday*, volume 63, page 93. Akademie Verlag, 2006.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL 2007 Shared Task. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, June 2007.

Václav Novák and Zdeněk Žabokrtský. Feature engineering in maximum spanning tree dependency parser. In Václav Matoušek and Pavel Mautner, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 10th I nternational Conference on Text, Speech and Dialogue*, Lecture Notes in Computer Science, pages 92–98, Pilsen, Czech Republic, 2007. Springer Science+Business Media Deutschland GmbH. ISBN 978-3-540-74627-0.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

K. Oflazer and I.D. El-Kahlout. Exploring different representational units in english-to-turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32. Association for Computational Linguistics, 2007.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, 2002.

Martin Popel and Zdeněk Žabokrtský. TectoMT: modular NLP framework. In *Proceedings of the 7th international conference on Advances in natural language processing*, IceTAL'10, pages 293–304, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14769-0, 978-3-642-14769-2. URL `http://portal.acm.org/citation.cfm?id=1884371.1884406`.

Rudolf Rosa and David Mareček. Dependency relations labeller for Czech. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue: 15th International Conference, TSD 2012. Proceedings*, number 7499 in Lecture Notes in Computer Science, pages 256–263, Berlin / Heidelberg, 2012. Masarykova univerzita v Brně, Springer Verlag. ISBN 978-3-642-32789-6.

Rudolf Rosa, Ondřej Dušek, David Mareček, and Martin Popel. Using parallel features in parsing of machine-translated sentences for correction of grammatical errors. In *Proceedings of Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6), ACL*, pages 39–48, Jeju, Korea, 2012a. ACL, ACL. ISBN 978-1-937284-38-1.

Rudolf Rosa, David Mareček, and Ondřej Dušek. DEPFIX: A system for automatic correction of Czech MT outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Montréal, Canada, 2012b. Association for Computational Linguistics. ISBN 978-1-937284-20-6.

B. Rosenfeld, R. Feldman, and M. Fresko. A systematic cross-comparison of sequence classifiers. *SDM 2006*, pages 563–567, 2006.

Petr Sgall. *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic, 1967.

Michel Simard, Cyril Goutte, and Pierre Isabelle. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York, April 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1064`.

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing, ACL 2007*, pages 67–74, Praha, 2007.

Lucien Tesnière and Jean Fourquet. *Eléments de syntaxe structurale*. Éditions Klincksieck, Paris, 1959.

David Vilar, Evgeny Matusov, Saša Hasan, Richard Zens, and Hermann Ney. Statistical machine translation of european parliamentary speeches. In *Proceedings of MT Summit X*, pages 259–266, 2005.

David Vilar, Jia Xu, Luis Fernando d'Haro, and Hermann Ney. Error analysis of statistical machine translation output. In *Proceedings of LREC*, pages 697–702, 2006.

William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association)*, pages 354–359, 1990. URL `http://www.amstat.org/sections/srms/Proceedings/papers/1990_056.pdf`.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Deyi Xiong, Min Zhang, and Haizhou Li. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 604–611. Association for Computational Linguistics, 2010.

Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. *NLP for Less Privileged Languages*, page 35, 2008.

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 55–63. Association for Computational Linguistics, 2009.

List of Terms and Abbreviations

MT machine translation

SMT statistical machine translation

SPE statistical post-editing

source a sentence in English which is being translated to Czech

target a sentence in Czech which is a machine translation of an English (source) sentence

fix an attempt of Depfix to correct a specific error; also the piece of code that makes such attempt

correction a successful fix

incorrection an unsuccessful fix

NLP natural language processing

m-layer morphological layer

a-layer analytical alyer

t-layer tectogrammatical layer

POS part of speech

morphological tag part-of-speech tag used in Czech

POS tag part-of-speech tag used in English, corresponding to the Penn Treebank tagset (Marcus et al., 1993)

MST maximum spanning tree

PMI pointwise mutual information

PDT Prague Dependency Treebank

PCEDT Prague Czech-English Dependency Treebank

UAS unlabelled attachment score

LAS labelled attachment score

NIST the NIST translation quality metric by Doddington (2002)

BLEU the BLEU translation quality metric by Papineni et al. (2002)

Attachments

A Examples of Depfix outputs

We provide a set of examples from our development data set, WMT_{10}. The examples are accompanied by reference translations and the changes are highlighted; however, glosses are not provided.

Examples 9.1, 9.2, 9.3, and 9.4 talk about president Obama receiving a Nobel Peace Prize, examples 9.5, 9.6, and 9.7 are concerned with a lack of snow in Czech mountains, and examples 9.8, 9.9, and 9.10 are dealing with cutting of financial bonuses of managers.

Source:	First, he will visit the Nobel Institute, where he will have his first meeting with the five committee members who selected him from 172 people and 33 organisations.
SMT output:	Zaprvé, že navštíví Nobelův institut, kde bude mít své první setkání s pěti členy výboru, kteří ho **vybrala** ze 172 lidí a 33 organizací.
Depfix output:	Zaprvé, že navštíví Nobelův institut, kde bude mít své první setkání s pěti členy výboru, kteří ho **vybrali** ze 172 lidí a 33 organizací.
Reference:	Nejprve zavítá do Nobelova institutu, kde se vůbec poprvé setká s pěti členy výboru, který ho v říjnu vybrali ze 172 lidí a 33 organizací.

Example 9.1

Source:	Then, in the afternoon, the visit will culminate in a grand ceremony, at which Obama will receive the prestigious award.
SMT output:	A pak, **odpoledne,** návštěva vyvrcholí **velkolepý obřad,** při němž Obama obdrží prestižní ocenění.
Depfix output:	A pak, **odpolednem,** návštěva vyvrcholí **velkolepým obřadem,** při němž Obama obdrží prestižní ocenění.
Reference:	Odpoledne potom návštěva vyvrcholí slavnostní ceremonií, na které Obama převezme prestižní ocenění.

Example 9.2

Source:	The president does not want to skirt around this topic, as he realises that he is accepting the prize as a president whose country is currently at war in two countries.
SMT output:	Prezident nechce obejít toto téma, protože si uvědomí, že **je akceptovat** cenu jako prezident, jehož země je v současné době ve válce ve dvou zemích.
Depfix output:	Prezident nechce obejít toto téma, protože si uvědomí, že **akceptuje** cenu jako prezident, jehož země je v současné době ve válce ve dvou zemích.
Reference:	Prezident se tomuto tématu nechce vyhnout, protože si uvědomuje, že cenu přebírá jako prezident, který v současnosti válčí ve dvou zemích.

Example 9.3

Source:	He has said, for example, that he feels that he does not wholly deserve it.
SMT output:	Řekl například, že cítí, že si plně **zaslouží.**
Depfix output:	Řekl například, že cítí, že si plně **nezaslouží.**
Reference:	Uvedl například, že má pocit, že si ji plně nezaslouží.

Example 9.4

Source:	The lack of snow is putting people off booking ski holidays in hotels and guest houses.
SMT output:	Nedostatek sněhu **je přimět** lidi z rezervace lyžařské dovolené v hotely a penziony.
Depfix output:	Nedostatek sněhu **přiměje** lidi z rezervace lyžařské dovolené v hotely a penziony.
Reference:	Chybějící sníh odrazuje lidi i od objednávání lyžařských pobytů v hotelech a penzionech.

Example 9.5

Source:	Hoteliers in Špindlerův Mlýn are so far just offering week-long stays for New Year's Eve.
SMT output:	Hoteliéři ve Špindlerově **mlýně jsou** zatím jen **nabízí** týdenní pobyty na Silvestra.
Depfix output:	Hoteliéři ve Špindlerově **Mlýně** zatím jen **nabízejí** týdenní pobyty na Silvestra.
Reference:	Špindlerovští hoteliéři zatím nabízejí na Silvestra převážně jen týdenní pobyty.

Example 9.6

Source:	What we're interested in is Christmas week, which hasn't sold out yet.
SMT output:	To, co nás zajímá, je vánoční týden, který dosud **prodal.**
Depfix output:	To, co nás zajímá, je vánoční týden, který dosud **neprodal.**
Reference:	Teď nás nejvíc zajímá vánoční týden, který zatím vyprodán není.

Example 9.7

Source:	Instead of money, the group of 30 top managers will receive shares, which cannot be sold for another five years.
SMT output:	Místo peněz, skupina 30 **vrcholoví manažeři** obdrží akcie, které nemohou být prodány za dalších pět let.
Depfix output:	Místo peněz, skupina 30 **vrcholových manažerů** obdrží akcie, které nemohou být prodány za dalších pět let.
Reference:	Skupina 30 předních manažerů dostane namísto peněz akcie, které bude možné prodat nejdříve za pět let.

Example 9.8

Source:	According to Reuters, Goldman Sachs is leading the effort to link Wall Street bonuses with long-term performance.
SMT output:	Podle agentury Reuters, Goldman Sachs je v čele úsilí o propojení Wall Street **bonusy** s dlouhodobou **výkonnost.**
Depfix output:	Podle agentury Reuters, Goldman Sachs je v čele úsilí o propojení Wall Street **bonusů** s dlouhodobou **výkonností.**
Reference:	Goldman Sachs se tak podle agentury Reuters staví do čela snahy o spojení odměn na Wall Street s dlouhodobou výkonnosti.

Example 9.9

Source:	High bonuses in the banking sector, what with the economic crisis, have become a political hot potato.
SMT output:	Vysoké odměny v bankovním sektoru, co s hospodářskou krizí, se **stal politický horký brambor.**
Depfix output:	Vysoké odměny v bankovním sektoru, co s hospodářskou krizí, se **staly politickým horkým bramborem.**
Reference:	Vysoké odměny v bankovním sektoru se s příchodem finanční krize staly v řadě zemí žhavým politickým tématem.

Example 9.10

B Scenarios

This attachment lists the Treex scenarios of the Depfix pipeline, in the order in which they are applied to the input. All of the scenarios are stored in the Treex repository in the `treex/devel/depfix/scenarios/` folder.

B.1 Analyses and Fixing on M-layer

```
Util::SetGlobal language=en
W2A::EN::Tokenize
W2A::EN::NormalizeForms
W2A::EN::FixTokenization
W2A::EN::TagMorce
W2A::EN::FixTags
W2A::EN::Lemmatize

Util::SetGlobal language=cs
W2W::ProjectTokenization source_language=en
W2A::CS::Tokenize
W2A::CS::TagFeaturama lemmatize=1
W2A::CS::FixMorphoErrors

Align::A::AlignMGiza dir_or_sym=intersection from_language=en
    to_language=cs model_from_share=en-cs
    tmp_dir=/COMP.TMP cpu_cores=1
Align::AddMissingLinks layer=a language=en target_language=cs
    alignment_type=intersection

Util::SetGlobal language=en
A2N::EN::StanfordNamedEntities model=ner-eng-ie.crf-3-all2008.ser.gz
A2N::EN::DistinguishPersonalNames
```

B.2 Parsing to A-layer

```
Util::SetGlobal language=en
W2A::EN::ParseMST model=conll_mcd_order2_0.01.model
W2A::EN::SetIsMemberFromDeprel
W2A::EN::RehangConllToPdtStyle
W2A::EN::FixNominalGroups
W2A::EN::FixIsMember
W2A::EN::FixAtree
W2A::EN::FixMultiwordPrepAndConj
W2A::EN::FixDicendiVerbs
W2A::EN::SetAfunAuxCPCoord
W2A::EN::SetAfun

Util::SetGlobal language=cs
W2A::CS::ParseMSTperl model_name=boost_model_025
    use_aligned_tree=1 alignment_language=en
```

```
    alignment_type=intersection alignment_is_backwards=1
W2A::CS::LabelMIRA model_name=pcedt_wors_para
    use_aligned_tree=1 alignment_language=en
    alignment_type=intersection alignment_is_backwards=1
A2A::GuessIsMember
```

B.3 Fixing on A-layer

```
A2A::CopyAtree source_language=cs language=cs selector=T
Align::AlignSameSentence language=cs to_selector=T
Align::AlignForward language=en

Util::SetGlobal language=cs selector=T
A2N::CS::SimpleRuleNER

Util::SetGlobal language=cs selector=T
    source_language=en dont_try_switch_number=1
A2A::CS::FixPOS dont_try_switch_number=0 magic=POSadj
A2A::CS::FixPrepositionalCase
W2A::CS::FixReflexiveTantum
A2A::CS::FixPassive
A2A::CS::FixNounNumber
A2A::CS::FixPrepositionWithoutChildren
A2A::CS::FixBy dont_try_switch_number=0
A2A::CS::FixAuxVChildren
A2A::CS::FixSubject
A2A::CS::FixVerbAuxBeAgreement
A2A::CS::FixPresentContinuous
A2A::CS::FixSubjectPredicateAgreement
A2A::CS::FixSubjectPastParticipleAgreement
A2A::CS::FixVerbByEnSubject
A2A::CS::FixPassiveAuxBeAgreement dont_try_switch_number=0
A2A::CS::FixPrepositionNounAgreement dont_try_switch_number=0
A2A::CS::FixOf dont_try_switch_number=0
A2A::CS::FixNounAdjectiveAgreement
A2A::CS::FixAuxT
```

B.4 Analysis to T-layer

```
Util::SetGlobal language=cs selector=T
A2T::CS::MarkEdgesToCollapse
A2T::BuildTtree
A2T::RehangUnaryCoordConj
A2T::SetIsMember
A2T::CS::SetCoapFunctors
A2T::FixIsMember
A2T::MarkParentheses
A2T::CS::DistribCoordAux
```

```
A2T::CS::MarkClauseHeads
A2T::CS::MarkRelClauseHeads
A2T::CS::MarkRelClauseCoref
A2T::DeleteChildlessPunctuation
A2T::CS::FixTlemmas
A2T::CS::FixNumerals
A2T::SetNodetype
A2T::CS::SetFormeme use_version=2 fix_prep=0
A2T::CS::SetDiathesis
A2T::CS::SetFunctors
A2T::CS::SetMissingFunctors
A2T::SetNodetype
A2T::FixAtomicNodes
A2T::CS::SetGrammatemes
A2T::CS::MarkReflexivePassiveGen
A2T::CS::AddPersPron
A2T::CS::MarkReflpronCoref

Util::SetGlobal language=en selector=
W2A::FixQuotes
A2T::EN::MarkEdgesToCollapse
A2T::EN::MarkEdgesToCollapseNeg
A2T::BuildTtree
A2T::SetIsMember
A2T::EN::MoveAuxFromCoordToMembers
A2T::EN::FixTlemmas
A2T::EN::SetCoapFunctors
A2T::EN::FixEitherOr
A2T::EN::FixHowPlusAdjective
A2T::FixIsMember
A2T::EN::MarkClauseHeads
A2T::EN::SetFunctors
A2T::EN::MarkInfin
A2T::EN::MarkRelClauseHeads
A2T::EN::MarkRelClauseCoref
A2T::EN::MarkDspRoot
A2T::MarkParentheses
A2T::SetNodetype
A2T::EN::SetTense
A2T::EN::SetGrammatemes
A2T::SetSentmod
A2T::EN::SetFormeme
A2T::EN::RehangSharedAttr
A2T::EN::SetVoice
A2T::EN::FixImperatives
A2T::EN::SetIsNameOfPerson
A2T::EN::SetGenderOfPerson
A2T::EN::AddCorAct
```

```
T2T::SetClauseNumber
A2T::EN::FixRelClauseNoRelPron
A2T::EN::FindTextCoref
```

B.5 Fixing on T-layer

```
Align::T::CopyAlignmentFromAlayer language=en
    to_language=cs to_selector=T
Align::ReverseAlignment language=en layer=t align_type=cs2en_int

Util::SetGlobal language=cs selector=T src_alignment_type=cs2en_int
T2T::CS2CS::PrecomputeNodeInfo
T2T::CS2CS::DropSubjPersProns
T2T::CS2CS::FixTense
T2T::CS2CS::FixNegation

Util::SetGlobal language=cs selector=T src_alignment_type=cs2en_int
Util::SetGlobal lower_threshold=1 lower_threshold_en=1 upper_threshold=1
Util::SetGlobal model_from_share=
    czeng10_ptlemma_syntpos_enformeme_formeme_147MW.model
Util::SetGlobal model_format=ptlemma_syntpos_enformeme_formeme
T2T::CS2CS::FixInfrequentNouns        upper_threshold_en=0.55 magic=no1
T2T::CS2CS::FixInfrequentPrepositions upper_threshold_en=0.90

Util::SetGlobal language=cs selector=T src_alignment_type=cs2en_int
Util::SetGlobal lower_threshold=1 lower_threshold_en=1 upper_threshold=1
Util::SetGlobal model_from_share=
    czeng10_new_tlemma_ptlemma_syntpos_enformeme_formeme_147MW.model
Util::SetGlobal model_format=tlemma_ptlemma_syntpos_enformeme_formeme
T2T::CS2CS::FixInfrequentPrepositions upper_threshold_en=0.84
T2T::CS2CS::AddFrequentPrepositions   upper_threshold_en=0.86

Util::SetGlobal language=cs selector=T src_alignment_type=cs2en_int
Util::Eval tnode='delete $tnode->wild->{deepfix_info};'
Util::Eval anode='delete $anode->wild->{deepfix_info};'
```

B.6 M-layer Translation Fixes

```
Util::SetGlobal language=cs selector=T source_language=en
A2A::CS::VocalizePrepos
A2A::CS::FixCasing
A2A::CS::FixFirstWordCapitalization
```

B.7 Detokenization

```
Util::SetGlobal language=cs selector=T
A2W::Detokenize
A2W::CS::DetokenizeUsingRules
```

```
A2W::CS::DetokenizeDashes
Util::Eval zone='print $zone->sentence . "\n";'
```

C The Feature Set for the Parser and Labeller

We present here our final feature set for the parser and the labeller, which were described in Chapter 5.

In Section C.1, we list the base feature set, with the features introduced in Section 5.3.3.

Section C.2 details our final feature set, with features described in Section 5.5 and Section 5.6. We do not repeat the listing of the base feature set as it only differers in changing `lemma` to `trunc_lemma`, i.e. cutting off the tails from the lemmas.

C.1 Base Feature Set

Parser

```
- COARSE_TAG
- COARSE_TAG|coarse_tag

- LEMMA
- LEMMA|lemma
- LEMMA|COARSE_TAG
- LEMMA|COARSE_TAG|coarse_tag
- LEMMA|COARSE_TAG|lemma
- LEMMA|lemma|coarse_tag
- COARSE_TAG|lemma|coarse_tag
- LEMMA|COARSE_TAG|lemma|coarse_tag

- FORM
- FORM|form
- FORM|COARSE_TAG
- FORM|COARSE_TAG|coarse_tag
- FORM|COARSE_TAG|form
- FORM|form|coarse_tag
- COARSE_TAG|form|coarse_tag
- FORM|COARSE_TAG|form|coarse_tag

- PRECEDING(coarse_tag)|COARSE_TAG|coarse_tag|following(coarse_tag)
- PRECEDING(coarse_tag)|COARSE_TAG|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|FOLLOWING(coarse_tag)|coarse_tag|following(coarse_tag)
- COARSE_TAG|FOLLOWING(coarse_tag)|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|coarse_tag|following(coarse_tag)
- COARSE_TAG|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|FOLLOWING(coarse_tag)|coarse_tag
- PRECEDING(coarse_tag)|COARSE_TAG|coarse_tag
- 1.coarse_tag|between(coarse_tag)|2.coarse_tag

- distance(ord)|COARSE_TAG
- distance(ord)|coarse_tag
- distance(ord)|COARSE_TAG|coarse_tag
```

```
- distance(ord)|LEMMA
- distance(ord)|lemma
- distance(ord)|LEMMA|lemma
- distance(ord)|LEMMA|COARSE_TAG
- distance(ord)|lemma|coarse_tag
- distance(ord)|LEMMA|COARSE_TAG|coarse_tag
- distance(ord)|LEMMA|COARSE_TAG|lemma
- distance(ord)|LEMMA|lemma|coarse_tag
- distance(ord)|COARSE_TAG|lemma|coarse_tag
- distance(ord)|LEMMA|COARSE_TAG|lemma|coarse_tag

- distance(ord)|FORM
- distance(ord)|form
- distance(ord)|FORM|form
- distance(ord)|FORM|COARSE_TAG
- distance(ord)|form|coarse_tag
- distance(ord)|FORM|COARSE_TAG|coarse_tag
- distance(ord)|FORM|COARSE_TAG|form
- distance(ord)|FORM|form|coarse_tag
- distance(ord)|COARSE_TAG|form|coarse_tag
- distance(ord)|FORM|COARSE_TAG|form|coarse_tag

- distance(ord)|PRECEDING(coarse_tag)|COARSE_TAG
      |coarse_tag|following(coarse_tag)
- distance(ord)|PRECEDING(coarse_tag)|COARSE_TAG
      |preceding(coarse_tag)|coarse_tag
- distance(ord)|COARSE_TAG|FOLLOWING(coarse_tag)
      |coarse_tag|following(coarse_tag)
- distance(ord)|COARSE_TAG|FOLLOWING(coarse_tag)
      |preceding(coarse_tag)|coarse_tag
- distance(ord)|COARSE_TAG|coarse_tag|following(coarse_tag)
- distance(ord)|COARSE_TAG|preceding(coarse_tag)|coarse_tag
- distance(ord)|COARSE_TAG|FOLLOWING(coarse_tag)|coarse_tag
- distance(ord)|PRECEDING(coarse_tag)|COARSE_TAG|coarse_tag
- distance(ord)|1.coarse_tag|between(coarse_tag)|2.coarse_tag
```

Labeller

```
- COARSE_TAG
- coarse_tag
- COARSE_TAG|coarse_tag

- LEMMA
- lemma
- LEMMA|lemma
- LEMMA|COARSE_TAG
- lemma|coarse_tag
- LEMMA|COARSE_TAG|coarse_tag
```

```
- LEMMA|COARSE_TAG|lemma
- LEMMA|lemma|coarse_tag
- COARSE_TAG|lemma|coarse_tag
- LEMMA|COARSE_TAG|lemma|coarse_tag

- FORM
- form
- FORM|form
- FORM|COARSE_TAG
- form|coarse_tag
- FORM|COARSE_TAG|coarse_tag
- FORM|COARSE_TAG|form
- FORM|form|coarse_tag
- COARSE_TAG|form|coarse_tag
- FORM|COARSE_TAG|form|coarse_tag

- PRECEDING(coarse_tag)|COARSE_TAG|coarse_tag|following(coarse_tag)
- PRECEDING(coarse_tag)|COARSE_TAG|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|FOLLOWING(coarse_tag)|coarse_tag|following(coarse_tag)
- COARSE_TAG|FOLLOWING(coarse_tag)|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|coarse_tag|following(coarse_tag)
- COARSE_TAG|preceding(coarse_tag)|coarse_tag
- COARSE_TAG|FOLLOWING(coarse_tag)|coarse_tag
- PRECEDING(coarse_tag)|COARSE_TAG|coarse_tag
- 1.coarse_tag|between(coarse_tag)|2.coarse_tag

- isfirstchild()|coarse_tag
- islastchild()|coarse_tag
- isfirstchild()|COARSE_TAG
- islastchild()|COARSE_TAG
- isfirstchild()|coarse_tag|COARSE_TAG
- islastchild()|coarse_tag|COARSE_TAG

- childno()|coarse_tag
- CHILDNO()|coarse_tag
- childno()|COARSE_TAG
- CHILDNO()|COARSE_TAG
- childno()|coarse_tag|COARSE_TAG
- CHILDNO()|coarse_tag|COARSE_TAG

- LABEL()
- coarse_tag|LABEL()
- COARSE_TAG|LABEL()
- coarse_tag|COARSE_TAG|LABEL()

- l.label()
- coarse_tag|l.label()
- COARSE_TAG|l.label()
```

```
- coarse_tag|COARSE_TAG|l.label()
- LABEL()|l.label()

- G.attdir()|coarse_tag
- G.attdir()|G.coarse_tag
- G.attdir()|coarse_tag|COARSE_TAG
- G.attdir()|COARSE_TAG|G.coarse_tag
- G.attdir()|coarse_tag|G.coarse_tag
- G.attdir()|coarse_tag|COARSE_TAG|G.coarse_tag

- coarse_tag|COARSE_TAG|G.coarse_tag
- coarse_tag|G.coarse_tag
- COARSE_TAG|G.coarse_tag

- G.label()
- LABEL()|G.label()

- r.coarse_tag
- coarse_tag|r.coarse_tag
- COARSE_TAG|r.coarse_tag
- coarse_tag|COARSE_TAG|r.coarse_tag
```

C.2 Extended Feature Set

Parser

```
- COARSE_TAG|aligned_afun
- COARSE_TAG|coarse_tag|aligned_afun
- TRUNC_LEMMA|aligned_afun
- TRUNC_LEMMA|trunc_lemma|aligned_afun

- distance(ord)|COARSE_TAG|aligned_afun
- distance(ord)|coarse_tag|aligned_afun
- distance(ord)|COARSE_TAG|coarse_tag|aligned_afun
- distance(ord)|TRUNC_LEMMA|aligned_afun
- distance(ord)|trunc_lemma|aligned_afun
- distance(ord)|TRUNC_LEMMA|trunc_lemma|aligned_afun

- aligned_edge
- COARSE_TAG|aligned_edge
- coarse_tag|aligned_edge
- COARSE_TAG|coarse_tag|aligned_edge
- TRUNC_LEMMA|aligned_edge
- trunc_lemma|aligned_edge
- TRUNC_LEMMA|trunc_lemma|aligned_edge

- distance(ord)|COARSE_TAG|aligned_edge
- distance(ord)|coarse_tag|aligned_edge
- distance(ord)|COARSE_TAG|coarse_tag|aligned_edge
```

- distance(ord)|TRUNC_LEMMA|aligned_edge
- distance(ord)|trunc_lemma|aligned_edge
- distance(ord)|TRUNC_LEMMA|trunc_lemma|aligned_edge

- COARSE_TAG|aligned_tag
- COARSE_TAG|coarse_tag|aligned_tag
- TRUNC_LEMMA|aligned_tag
- TRUNC_LEMMA|trunc_lemma|aligned_tag

- distance(ord)|COARSE_TAG|aligned_tag
- distance(ord)|coarse_tag|aligned_tag
- distance(ord)|COARSE_TAG|coarse_tag|aligned_tag
- distance(ord)|TRUNC_LEMMA|aligned_tag
- distance(ord)|trunc_lemma|aligned_tag
- distance(ord)|TRUNC_LEMMA|trunc_lemma|aligned_tag

- pmibucketed(trunc_lemma)
- distance(ord)|pmibucketed(trunc_lemma)

Labeller

- COARSE_TAG|aligned_afun
- coarse_tag|aligned_afun
- COARSE_TAG|coarse_tag|aligned_afun
- TRUNC_LEMMA|aligned_afun
- trunc_lemma|aligned_afun
- TRUNC_LEMMA|trunc_lemma|aligned_afun

- COARSE_TAG|aligned_edge
- coarse_tag|aligned_edge
- COARSE_TAG|coarse_tag|aligned_edge
- TRUNC_LEMMA|aligned_edge
- trunc_lemma|aligned_edge
- TRUNC_LEMMA|trunc_lemma|aligned_edge

- COARSE_TAG|aligned_tag
- coarse_tag|aligned_tag
- COARSE_TAG|coarse_tag|aligned_tag
- TRUNC_LEMMA|aligned_tag
- trunc_lemma|aligned_tag
- TRUNC_LEMMA|trunc_lemma|aligned_tag