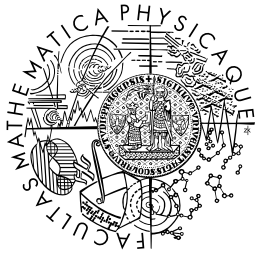


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

ABSTRACT OF DOCTORAL THESIS

Rudolf Rosa

Discovering the structure of natural language sentences by semi-supervised methods

Institute of Formal and Applied Linguistics

Supervisor:	doc. Ing. Zdeněk Žabokrtský, Ph.D.
Study programme:	Informatics
Study branch:	Mathematical Linguistics

Beroun 2018

The results of this thesis were achieved in the period of a doctoral study at the Faculty of Mathematics and Physics, Charles University in years 2013-2018.

Student: Mgr. Rudolf Rosa

Supervisor: doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25, 118 00 Prague 1

Department: Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25, 118 00 Prague 1

Opponents: prof. Jörg Tiedemann, Ph.D.
Department of Digital Humanities
Faculty of Arts
University of Helsinki
P.O. Box 24 (Unioninkatu 40 B), FI-00014 Helsinki, Finland

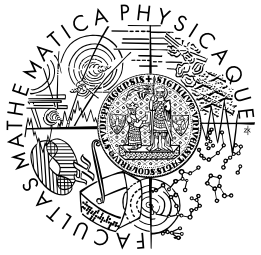
doc. RNDr. Aleš Horák, Ph.D.
Department of Machine Learning and Data Processing
Faculty of Informatics
Masaryk University
Botanická 554/68a, 602 00 Brno

The thesis defence will take place on June 14, 2018 at 10:00 a.m. in front of a committee for thesis defences in the branch Mathematical Linguistics at the Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, Prague 1, room S1.

Chairman of Academic Council: doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25, 118 00 Prague 1

The thesis can be viewed at the Study Department of Doctoral Studies of the Faculty of Mathematics and Physics, Charles University, Ke Karlovu 3, Prague 2.

This abstract was distributed on May 31, 2018.



MATEMATICKO-FYZIKÁLNÍ
FAKULTA
Univerzita Karlova

AUTOREFERÁT DISERTAČNÍ PRÁCE

Rudolf Rosa

Odhalování struktury vět přirozeného jazyka pomocí částečně řízených metod

Ústav formální a aplikované lingvistiky

Školitel:	doc. Ing. Zdeněk Žabokrtský, Ph.D.
Studijní program:	Informatika
Studijní obor:	Matematická lingvistika

Beroun 2018

Disertační práce byla vypracována na základě výsledků získaných během doktorského studia na Matematicko-fyzikální fakultě Univerzity Karlovy v letech 2013-2018.

Doktorand:	Mgr. Rudolf Rosa
Školitel:	doc. Ing. Zdeněk Žabokrtský, Ph.D. Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské nám. 25, 118 00 Praha 1
Školicí pracoviště:	Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské nám. 25, 118 00 Praha 1
Oponenti:	prof. Jörg Tiedemann, Ph.D. Digitaalisten ihmistieteiden osasto Humanistinen tiedekunta Helsingin yliopisto PL 24 (Unioninkatu 40 B), 00014 Helsinki, Suomi
	doc. RNDr. Aleš Horák, Ph.D. Katedra strojového učení a zpracování dat Fakulta informatiky Masarykova univerzita Botanická 554/68a, 602 00 Brno

Obhajoba disertační práce se koná dne 14. června 2018 v 10:00 před komisí pro obhajoby disertačních prací v oboru Matematická lingvistika na Matematicko-fyzikální fakultě UK, Malostranské nám. 25, Praha 1, v místnosti S1.

Předseda RDSO:	doc. Ing. Zdeněk Žabokrtský, Ph.D. Ústav formální a aplikované lingvistiky Matematicko-fyzikální fakulta Univerzita Karlova Malostranské nám. 25, 118 00 Praha 1

S disertační prací je možno se seznámit na studijním oddělení Matematicko-fyzikální fakulty UK, Ke Karlovu 3, Praha 2.

Autoreferát byl rozeslán dne 31. května 2018.

Contents

Introduction

The topic of this thesis is automatic linguistic analysis of written text, specifically syntactic dependency parsing, and, to some extent, Part of Speech (POS) tagging.

In the classical supervised parsing (Chapter 2), a parser is trained on a syntactically annotated corpus, i.e. a treebank. To achieve a reasonable parsing accuracy, the treebank should contain thousands or tens of thousands of manually annotated sentences. However, such treebanks are expensive to create, and are thus available only for a few dozen languages; currently, less than 80 languages have at least a tiny treebank available. This renders approximately 99% of the world's languages *under-resourced* in terms of parsing resources, as the classical fully supervised approach to parsing cannot be applied for these languages.

This situation constitutes the main motivation for our work. While treebanks are not available for those languages, there is a belief that most or all languages in the world are similar to each other to some extent. Therefore, annotated resources for resource-rich languages might be utilized to learn knowledge useful for analyzing other languages, especially similar ones. Moreover, even if no treebank is available for a language, we might still exploit other resources. We discuss the datasets potentially useful for parsing in Chapter 1.

One possible approach to use is the cross-lingual transfer of a delexicalized parser, which we introduce in Chapter 3. Here, the idea is that even if two languages differ in their lexicon, they might not differ that much in their grammar. Therefore, a parser trained without any lexical features on a treebank for a resource-rich *source* language (i.e. a delexicalized parser) might be applicable to a resource-poor *target* language. As the delexicalized parser transfer approach has been repeatedly shown to perform well, we take it as the basis for our research, and extend it in several ways.

There is already a wide range of resource-rich languages, which can be used as source languages in the parser transfer. However, automatically choosing the optimal source language is an important yet non-trivial task. Moreover, a clever combination of multiple sources might be an even better approach to take. We address both of these issues in Chapter 4, where we introduce our language similarity measure, $KL^{-4}_{cpos^3}$, and we port a monolingual multi-source parser combination method into the cross-lingual setting.

By delexicalizing the parser, we are losing accuracy. Fortunately, existing parallel text corpora can be utilized to lexicalize the cross-lingual parsing, either directly through word alignment links, or indirectly via Machine Translation (MT). In Chapter 5, we take the latter approach, investigating the potential of word-based MT approaches and their advantages over phrase-based MT systems.

A problem we have been leaving unaddressed so far is the fact that we typically need to provide the parsers with a morphological annotation of the input sentences, at least in terms of POS tags. However, as we cannot reasonably assume to have supervised POS taggers available for all under-resourced target languages, we need to apply cross-lingual approaches even for tagging, which we investigate in Chapter 6.

We conclude the thesis by summarizing our findings in the form of step-by-step instructions for parsing an under-resourced language.

1. Datasets for Parsing

In this chapter, we deal with the data that we use in cross-lingual parsing.

The key resource for data-driven dependency parsing are dependency treebanks, i.e. corpora of sentences annotated with syntactic trees. In the case of resource-rich target languages for which large treebanks are available, the task of parsing then consists of training an off-the-shelf parser on the treebank and applying it to the texts in the target language.

However, in our scenario, we assume the target languages to be resource-poor, with no annotated data available. Our approaches are thus based on exploitation of treebanks for different source languages, and transfer of the knowledge learned from those treebanks into the target languages. Unfortunately, treebanks tend to use a wide range of different styles of both morphological and syntactic annotation, which poses significant problems to any cross-lingual processing – we need the treebanks to be *harmonized*, i.e. to be annotated in an as much similar way as possible. While the harmonization of syntactic annotation is obviously crucial for cross-lingual parsing, we need the morphological annotation to be harmonized as well, as it constitutes a very important input feature for parsing (this is especially true for the POS tags). We explicitly list the harmonized treebank datasets or their subsets that we use in our experiments in Section 1.1.

Another very important resource for any cross-lingual processing are parallel corpora, i.e. texts in one (source) language accompanied by their human-devised translations in another (target) language. These enable us to transfer annotation or knowledge from the source language into the target language, typically either by means of projection over word alignment on the parallel data, or by training an MT system on the parallel data. Fortunately, parallel data are "natural resources", available in the wild for harvesting and subsequent construction of parallel corpora. We discuss parallel corpora in Section 1.2, with a focus on parallel data that are typically available for under-resourced languages.

However, we ideally want to make use of any relevant resources, also including e.g. linguistic catalogues (Section 1.3).

1.1 Treebank datasets used in our experiments

As our research was done over the course of several years, during which a lot changed in the field of treebank collections, we did not keep our dataset fixed throughout the whole time. For the earlier experiments, we used the Stanfordized HamleDT 2.0 treebanks (Section 1.1.1); later on, we switched to using Universal Dependencies v1.4 (Section 1.1.2).

1.1.1 HamleDT 2.0 dataset

When the research for this thesis commenced, the HamleDT 2.0 collection [Rosa et al., 2014, Zeman et al., 2014] was by far the largest as well as most harmonized existing treebank collection and thus the logical, or probably even the only reasonable, choice of dataset. Our work was thus among the first ones to be applied to a really large harmonized treebank collection. HamleDT 2.0 featured

30 treebanks in 30 languages in both the Prague Dependencies and the Universal Stanford Dependencies annotation style – and although unfortunately only some of them were freely available to the public, we had access to all of them, giving us a great advantage then. Thus, the experiments done early on in our research are performed and evaluated using the Stanfordized HamleDT 2.0 dataset.

The Stanfordized treebanks are annotated with a set of 33 dependency relation labels inspired by the Universal Stanford Dependencies (USD) of de Marneffe et al. [2014] and the Google Stanford Dependencies (GSD) of McDonald et al. [2013], and with the 12 Universal Part of Speech Tagset (UPT) tags as defined by Petrov et al. [2012]. As we initially focused solely on parsing, we use the gold-standard UPT tags in all our experiments conducted on the HamleDT dataset. The treebanks also contain fine-grained Interset morphological annotations [Zeman, 2008], but we did not use these in our experiments.

1.1.2 Universal Dependencies 1.4 subset

In our more recent experiments, we switched from the HamleDT 2.0 dataset to Universal Dependencies (UD) 1.4 [Nivre et al., 2016], as this was the newest UD release available at the time of the switch, featuring 64 treebanks for 47 languages.

The key pillars of UD treebanks annotation are:

- Universal Part of Speech (UPOS) tags, based on UPT,
- Universal morphological features, based on the Interset,
- Universal dependency structure and universal dependency labels, based primarily on Stanford Dependencies (SD), but including notions from USD, HamleDT, and GSD.

Detailed annotation style descriptions, with a large number of practical examples in many languages, are maintained online.[1]

The first set of 10 UD-harmonized treebanks, UD v1.0, was released in January 2015 [Nivre et al., 2015]. A new version of the collection is released every 6 months, adding both conversions of existing treebanks (done manually, semi-automatically, or automatically, potentially with checks and post-corrections), as well as new treebanks annotated in the UD style from scratch. At the time of writing, the latest release is UD v2.1 [Nivre et al., 2017], containing 102 treebanks for 60 languages; the annotation style was partially modified in the transition to the version 2.0.[2] Practically all of the UD treebanks are easily available for download under permissive licences. Thanks to all of the aforementioned characteristics, UD annotation style and datasets have quickly become the current *de facto* standard for most of the work on treebanking, dependency parsing, as well as POS tagging, both monolingual and cross-lingual.

1.2 Parallel corpora

A parallel text corpus is a resource consisting of a text in one language and its translation in another language. Parallel texts are "natural resources", produced by human translators and published for various reasons – we can often easily

[1] `http://universaldependencies.org/guidelines.html`

[2] `http://universaldependencies.org/v2/summary.html`

Corpus	Available languages easily	potentially	Typical number of sentences
OpenSubtitles	62	78	5M – 30M
Watchtower	135	300	100k – 150k
Bible	100	1200/4000	10k – 30k
UDHR	400	544	60 – 70

Table 1.1: Overview of some parallel and multiparallel corpora, with the number of languages for which it is easily or at least potentially available, and a typical size in number of sentences.

get religious texts, international laws, film subtitles, etc. Parallel corpora are often freely available for download, or can be compiled from parallel data harvested from the internet. Still, for under-resourced languages, even the amount of available parallel data is usually lower than for resource-rich languages.

Parallel corpora are typically used in Natural Language Processing (NLP) to train Statistical Machine Translation (SMT) systems, which can be useful for many tasks, including cross-lingual parsing. In the cross-lingual projection approach, parallel data are even used directly to project annotations from its one side to the other side, without using an SMT system.

In many cases, translations of the same texts are available in multiple languages; such resources are usually referred to as multiparallel corpora, and can be even more useful for cross-lingual processing.

Moreover, many parallel corpora can be downloaded from linguistic repositories, such as the OPUS collection of Tiedemann [2012],[3] which publish them in a preprocessed format, usually including sentence segmentation and sentence alignment, and often also tokenization.

In Table 1.1, we present an overview of parallel corpora which are available for a large number of languages (in fact, all of the listed corpora are actually multiparallel, at least to some extent).

In our experiments, we have only used the first two, OpenSubtitles and Watchtower Corpus (WTC). However, we also list the other two, Bible and Universal Declaration of Human Rights (UDHR), as they are available for an even larger number of languages than the first two, thus broadening the potential scope of cross-lingual parsing methods.

1.2.1 OpenSubtitles

The OpenSubtitles corpora are film and TV series subtitles and their translations provided by volunteers through the OpenSubtitles web portal.[4] While the translations are of varying quality, they have been repeatedly successfully used by many researchers. The data are typically sufficiently large, making it possible to train high-quality SMT systems, while also being available in a respectable number of languages. Unfortunately, these are mostly resource-rich languages; for resource-poor languages, little or no data are often available in this corpus, which gravely limits the usefulness of this dataset in the intended use case of cross-lingual parsing.

[3] `http://opus.nlpl.eu/`

[4] `http://www.opensubtitles.org/`

Nevertheless, we employed the OpenSubtitles data in some of our experiments, in particular using the OpenSubtitles2016 version, published by Lison and Tiedemann [2016]. We report the sizes of the parallel data which we used together with the particular languages in Section 1.1.2. We always split of the first 10,000 sentences from the dataset as development data, used for tuning the MT systems, and the last 10,000 sentences as test data, used to intrinsically evaluate the quality of the MT systems.

1.2.2 Watchtower

Agić et al. [2016] introduced a much more realistic resource for cross-lingual parsing: the Watchtower Corpus (WTC). It consists of texts of the Watchtower magazine, published by Jehovah's Witnesses via the Watch Tower Bible and Tract Society of Pennsylvania in a large number of languages, including many under-resourced ones. The texts are available on the Watchtower Online website,[5] from which they were scraped by Agić et al. [2016] and compiled into the WTC.

The WTC contains texts in 135 languages. However, it seems that many more languages are available on the Watchtower website; at the time of writing, it advertises texts in 301 languages, which suggests that the scope of the corpus (and, subsequently, of the presented cross-lingual methods) could still be extended.

For each language, the corpus contains at least 27,000 and no more than 167,000 sentences; the average number of sentences is 116,000, the median is 127,000. These are thus drastically smaller data than the OpenSubtitles, inevitably leading to considerably worse results. However, in line with Agić et al. [2016], we believe this to be a much more realistic setting for under-resourced languages, leading to more plausible estimates of the parsing accuracies – for real under-resourced languages, really large parallel corpora are typically simply not available. We thus use OpenSubtitles for several rather exploratory experiments, but ultimately apply WTC in our final setups.

On the plus side, the WTC data are massively multiparallel, as they consist of translations of the same texts. The texts in WTC are tokenized on punctuation symbols by a trivial tokenizer. This means that languages which do not separate words by spaces, such as Japanese, are not properly tokenized; the results which we report for Japanese thus suffer from this, but we find it useful to investigate what the results are under such settings. The texts are also segmented into sentences using a similar approach, with one sentence per line. The average number of tokens in an English sentence is 16.5 in WTC.

1.3 Linguistic catalogues

The World Atlas of Language Structures (WALS) of Dryer and Haspelmath [2013] is one of the most well-known and respectable sources of information about world's languages. It is a manually curated database, gathering typological information about a wide range of languages and organized in a structured way, and is freely available both for online browsing and for download. This makes it a very valuable resource for any work focusing on a wider range of languages.

[5] `https://wol.jw.org/`

2. Dependency Parsing

In computational linguistics, parsing, or syntactic analysis, is the act of revealing the structural (syntactic) relations between words in a sentence, and presenting them in the form of a graph, usually an ordered rooted parse tree. Syntactic parsing is a classical NLP task, with the tool that performs this task being called a parser. The input to a parser is typically a tokenized and morphologically annotated sentence, and the output is a syntactic parse tree of the sentence.

In this thesis, we focus on the dependency parsing paradigm, which has become the *de facto* standard in recent years, especially in the multilingual setting, with large collections of harmonized treebanks being available for dozens of languages.

Specifically, we have used two parsers throughout our work. In earlier experiments, we use the MSTperl parser [Rosa, 2015], which is a representative of the graph-based parsers. In later experiments, we use the UDPipe/Parsito parser [Straka et al., 2016], which is transition-based.

2.1 Parser evaluation

Dependency parsers are typically evaluated using two measures: Unlabelled Attachment Score (UAS), and Labelled Attachment Score (LAS). Both of these measures are simply token-level accuracies, taking into account all tokens in the test data, and giving each token an equal weight in the evaluation.

Unlabelled Attachment Score (UAS) [Eisner, 1996] only takes the tree structure into account, ignoring the dependency relation labels. Each token is considered correctly parsed if it is assigned the correct head; otherwise, it is considered to be parsed incorrectly. UAS is simply the proportion of correctly parsed tokens – e.g. if there are 10 tokens in the test data, and the parser attaches 8 of them to correct parents, its UAS is 80%.

Labelled Attachment Score (LAS) assesses both the assigned head as well as the assigned dependency relation label. While theoretically, the label expresses the relation between the head node and the dependent node, in practice, it is generally treated as belonging to the dependency node; thus, each node is assigned exactly one head and one dependency relation label. LAS then considers a node correctly parsed if it is assigned both the correct head and the correct dependency relation label; all incorrectly parsed nodes are treated equally, i.e. it does not matter whether only the head is incorrect, only the label is incorrect, or both are incorrect. As could be expected, the LAS of a parser output is the proportion of correctly parsed nodes under these criteria. LAS is typically used as the standard evaluation measure for dependency parsing, including e.g. parsing shared tasks [Buchholz and Marsi, 2006, Zeman et al., 2017].

3. Delexicalized Parser Transfer

In this chapter, we introduce our base approach to cross-lingual parsing, the single-source delexicalized parser transfer.

3.1 Delexicalized parsing

A typical syntactic parser is *lexicalized*, i.e. it uses the individual word forms as input features. Usually, it also takes in POS tags as a useful abstraction over the individual words, which helps it generalize over rare words and rare contexts. However, the lexical features bind the parser tightly to the vocabulary appearing in the training data. This may already pose problems in monolingual parsing, e.g. if the training data is very small. However, it becomes a fundamental obstacle in cross-lingual parsing, where we intend to apply the parser to a different language, which, unless handled somehow, is bound to render the learned lexical features mostly or completely useless.

One possible way out is to remove the lexical features from the parser, using only the POS tags (and potentially other morphological features, such as case and gender), thus obtaining a *delexicalized* parser.

In the simplest case of delexicalized parsing, only coarse-grained POS tags, such as UPOS, are used as the input features. The morphological lemmas obviously need to be removed, since these are lexical features. With fine-grained morphological features, such as case, number, gender, or tense, the situation is less clear – by default, we remove all of these features, since we found they mostly do not transfer well cross-lingually.

The delexicalization is inevitably a lossy procedure. For some sentences, their syntactic structure can be easily determined even without the lexical information, just based on the POS tags. In other cases, stripping the lexical information introduces ambiguity, as the same sequence of POS tags can have multiple syntactic analyses.

3.2 Delexicalized parser transfer

In this thesis, we take the single-source delexicalized parser transfer as our base approach to cross-lingual parsing, upon which we build our methods. The method was introduced by Zeman and Resnik [2008], who trained a delexicalized parser on a Danish treebank and evaluated it on a Swedish one. They note that while the lexicons of these two languages will most probably differ significantly even if they are very close, they may share both many morphological as well as syntactic properties, which motivates their approach.

While the intended use-case is the syntactic analysis of an under-resourced target language using a resource-rich source language, the authors take the usual approach of simulating this situation by evaluating on a treebank for a resource-rich language – while there is a Swedish treebank available, its syntactic annotation is only used by the authors to be able to evaluate their method. This is an approach we also follow in our work.

4. Using Multiple Sources

Multiple potential source treebanks for resource-rich languages are usually available, and it is non-trivial to select the best one for a given target language. While mostly ignored at first, the problem became rather clear with more and more treebanks becoming available, and researchers in cross-lingual parsing have made various attempts at solving it, including both methods of selecting one best source to use, as well as combining multiple sources.

The key part of this chapter, as well as of the whole thesis, is our attempt at solving this problem, using a designated language similarity measure, and a refurbished method for parser combination.

In Section 4.1, we introduce KL_{cpos^3}, our language similarity measure based on Kullback-Leibler divergence of probability distributions of coarse POS tag trigrams, estimated from POS-tagged corpora for the source and target languages. The measure has been designed and tuned specifically for multilingual delexicalized parser transfer, to be used both to select the most similar source language for a given target language, as well as to assign weights to multiple source languages in a multi-source combination (Section 4.2.1).

In Section 4.3, we show that in a single-source setting, KL_{cpos^3} often succeeds in selecting the best available source treebank for a given target language, or, in many other cases, selects a different but competitive one. In the multi-source parse tree combination approach, KL_{cpos^3} is often able to appropriately weight the available source treebanks, so that their weighted combination outperforms an unweighted one in many cases as well as on average.

Interestingly, KL_{cpos^3} has also been shown to perform well in various modifications of the original setting, for which it was not originally designed or tuned. It stays accurate when computed on cross-lingually induced POS instead of gold ones, as independently confirmed by Agić [2017], who also shows it to outperform other language similarity measures in that setting. It is also successful in lexicalized parsing instead of delexicalized (Chapter 5), and even when applied to cross-lingual POS tagging instead of parsing (Chapter 6).

Thus, eventually, our ultimate best-performing setup successfully uses KL_{cpos^3} at several places. We therefore consider KL_{cpos^3} to be the key component of our approach, and the most important invention presented in this thesis.

Some parts of this chapter are adapted from [Rosa and Žabokrtský, 2015a] and [Rosa and Žabokrtský, 2015b].

4.1 KL_{cpos^3} language similarity measure

In this section, we present KL_{cpos^3}, our language similarity measure based on KL divergence [Kullback and Leibler, 1951] of POS tag trigram distributions in tagged corpora.

It has been designed for multi-source cross-lingual delexicalized parsing, both for source treebank selection in single-source parser transfer, which has been described in Section 3.2, and for source treebank weighting in multi-source transfer, which will be described in Section 4.2.1.

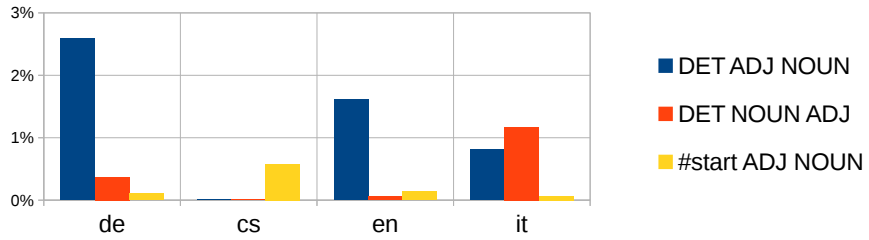


Figure 4.1: Example of estimated probability distributions of several selected POS trigrams in four languages.

4.1.1 The formula

The measure is based on comparing estimated probability distributions of POS sequences that appear in the source and target languages. This is motivated by the fact that POS tags constitute a key feature for delexicalized parsing.

The probability distributions are estimated as relative frequencies of POS tag trigrams in the treebank training sections:

$$\hat{P}(cpos_{i-1}, cpos_i, cpos_{i+1}) = \frac{\text{count}(cpos_{i-1}, cpos_i, cpos_{i+1})}{\sum_{\forall cpos_{a,b,c}} \text{count}(cpos_a, cpos_b, cpos_c)}; \tag{4.1}$$

we use a special value for $cpos_{i-1}$ or $cpos_{i+1}$ if $cpos_i$ appears at sentence beginning or end.

See Figure 4.1 for an example of the estimated probability distributions of three POS tag trigrams in four languages. It can be seen that, at least as far as these particular tag sequences are concerned, the English and German languages are quite close to each other, while both Italian and Czech are quite distant from any of the languages.

The particular tag sequences used in this example correspond to the way noun phrases are formed in these languages. Italian has a slight preference of the adjectives to follow the nouns they modify, although they also often precede them; however, in all the other languages, the adjective-noun POS bigram is much more frequent than the noun-adjective one. Furthermore, except for Czech, all of the languages typically start a noun phrase with a determiner, so it is common for a determiner to precede an adjective, and rare for the adjective to immediately start a sentence; in Czech, where determiners are rare, this is the other way round.

Intuitively, when an automatic parser is applied to analyze the structure of a noun phrase in a given target language, one should use a parser trained on a source language that structures noun phrases similarly, so that it can produce the correct analysis. While we assume not to have syntactically annotated data for the target language, Figure 4.1 shows that already morphologically annotated data can suggest a lot about the syntax of the language. This is the motivation behind estimating language similarity from probability distributions of POS tag n-grams.

Furthermore, if we were to analyze English noun phrases by either an Italian or a Czech parser, we expect Italian to be a better choice, since the *DET-ADJ-NOUN* sequence is well known to it, while this is not true for the Czech parser.

Moreover, even though the Italian parser also expects to see the inversely ordered *DET-NOUN-ADJ* sequences on the input, which are rare in English, this might not matter much, since this simply means that the ability of the parser to analyze such sequences will remain unexploited when applied to English. This would however pose a problem in the other direction, using an English parser to analyze Italian noun phrases, since the *DET-NOUN-ADJ* sequence would presumably confuse the parser greatly, as it is not used to encountering it, and therefore presumably unable to handle it correctly. This motivates our use of KL divergence in a particular direction.

We first represent here the general formula for KL divergence from Q to P, $D_{\mathrm{KL}}(P||Q)$, where P and Q are two discrete probability distributions, P being the true or expected distribution, and Q being an approximation or model of P used instead of P:

$$D_{\mathrm{KL}}(P||Q) = \sum_{\forall x} P(x) \cdot \log \frac{P(x)}{Q(x)}\,, \tag{4.2}$$

with the value of the addend defined as 0 if $P(x) = 0$. The value of KL divergence is a non-negative number; the more divergent (dissimilar) the distributions, the higher its value.

In our setting, we estimate the distance of a source language to a target language as the KL divergence of the POS trigram probability distributions, $D_{\mathrm{KL}}(\hat{P}_{tgt}||\hat{P}_{src})$:

$$KL_{cpos^3}(tgt, src) = \sum_{\forall cpos^3 \in tgt} \hat{P}_{tgt}(cpos^3) \cdot \log \frac{\hat{P}_{tgt}(cpos^3)}{\hat{P}_{src}(cpos^3)}\,, \tag{4.3}$$

where $cpos^3$ is a POS tag trigram.

For the KL divergence to be well-defined, we must ensure that the estimated probability of each target trigram is non-zero in source. For this, we employ a simple "add 1" smoothing approach in the source trigrams probability estimation (4.1): for each target trigram unseen in the source data, we set its source count to 1.

The KL divergence is non-symmetric; $D_{\mathrm{KL}}(P||Q)$ expresses the amount of information lost when a probability distribution Q is used to approximate the true distribution P. Thus, in our setting, we use $D_{\mathrm{KL}}(\hat{P}_{tgt}||\hat{P}_{src})$, as this intuitively corresponds to trying to minimize the error caused by using a source parser as an approximation of a target parser (we are approximating the target language by the source language).

4.1.2 KL_{cpos^3} for source selection

In the single-source parser transfer, the delexicalized parser is trained on a single source treebank, and applied to the target corpus. The problem thus reduces to selecting a source treebank that will lead to a high performance on the target language.

In this case, we compute the KL_{cpos^3} distance of the target corpus to each of the source treebanks and choose the closest source treebank to use for the transfer.

4.1.3 $KL^{-4}_{cpos^3}$ for source weighting

In multi-source transfer, multiple (or all) available source treebanks are used for parser training, possibly weighted by similarity to the target language. In this case, we need to appropriately weight the contribution of each of the sources.

To convert KL_{cpos^3} from a negative measure of language similarity to a positive source parser weight for the multi-source tree combination method, we need to find a way of inverting it. However, as the results were unsatisfactory with simply using the inverted value $(1/x)$, we did some further tuning, empirically finding the fourth power of the inverted value $(1/x^4)$ to work well. Thus, the contribution of each of the sources gets weighted by $KL^{-4}_{cpos^3}(tgt, src)$.

We deal with multi-source transfer in Section 4.2.

4.2 Multi-source combination methods

To the best of our knowledge, the idea of combining multiple source languages for analyzing one target languages was introduced by McDonald et al. [2011]. The authors used a simple treebank concatenation method, combining all available source treebanks into one multilingual treebank, and using it to train one multilingual delexicalized parser. This method does not assign explicit weights to individual source languages; each source language is implicitly weighted by the size of its treebank, regardless of the target language. We take this method as a baseline approach.

Our work on cross-lingual parsing in a multi-source setting rests on two pillars. The first one, the KL_{cpos^3} language similarity measure, was presented in Section 4.1, allowing us to estimate how appropriate each available source language is for processing a given target language, i.e. for training a delexicalized parser on the source language and applying it to the target language.

However, as was already foreshadowed, it is often the case that there are multiple source languages close enough to the target, and one would then like to learn from all such languages; the hope is that treebanks for other similar languages might provide knowledge which is not available in the treebank for the closest language but is necessary to parse the target language.

And there is yet another, more pragmatic reason to take multiple sources into account. Even though it performs very well, the KL_{cpos^3} measure is far from infallible, often failing to designate the optimal source language. In such cases, we would like to have a method of bringing in other promising sources, trying to alleviate the damage done by not choosing the right source. This is a sort of risk management – we accept the risk of achieving slightly suboptimal accuracies for some languages to avoid massively suboptimal performance for other languages; this is what we tuned the $KL^{-4}_{cpos^3}$ measure for.

For this purpose, we build upon the parse tree combination method of Sagae and Lavie [2006], which we ported to the cross-lingual setting in [Rosa and Žabokrtský, 2015b]; this is our primary method, as it showed best performance in our evaluations, and we will only use this method in further experiments. We also investigated two alternative methods – parser model interpolation [Rosa and Žabokrtský, 2015a] and parse tree projection [Agić et al., 2016] – but have not found them to outperform the parse tree combination.

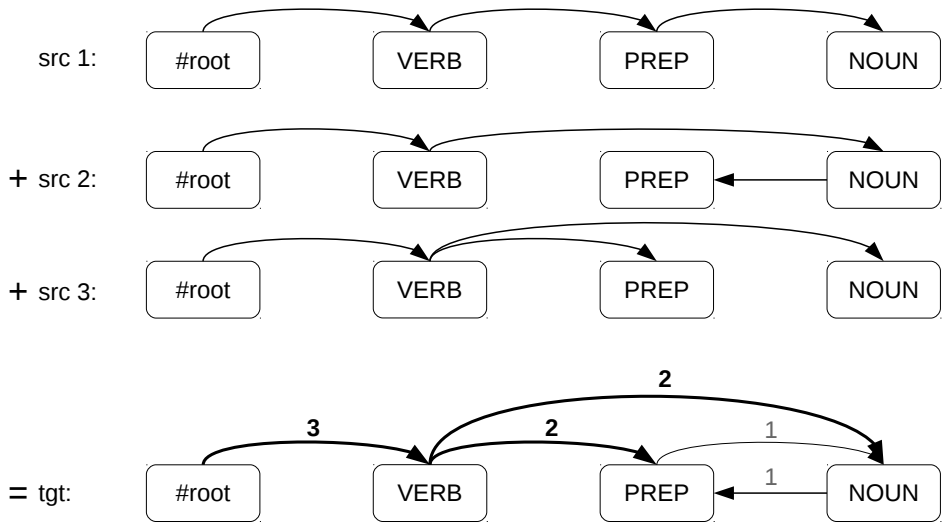


Figure 4.2: Unweighted parse tree combination, combining the parse trees for a delexicalized target sentence (tgt) produced by 3 source parsers (src 1, src 2 and src 3), and selecting the highest scoring dependency tree (MST) as the result (in bold).

4.2.1 Parse tree combination

The multi-source cross-lingual delexicalized parse tree combination method is a simple parser ensembling approach. The original method, which had been devised for a monolingual setting, combines various parsers (i.e. different parsing algorithms), all trained on the same treebank. In our extension to the cross-lingual setting, we only use one dlexicalized parser trained on various source treebanks.

Unweighted tree combination

In our work, we implement the tree combination method in its base unweighted variant in the following way (see also Figure 4.2):

1. Train a delexicalized parser on each source treebank.
2. Apply each of the parsers to the target sentence, obtaining a set of parse trees.
3. Construct a weighted directed graph as a complete graph over all tokens of the target sentence, where each edge is assigned a score equal to the number of parse trees in which it appears (each parse tree contributes by either 0 or 1 to the edge score).
4. Find the final dependency parse tree as the maximum spanning tree over the graph, using the algorithm of Chu and Liu [1965] and Edmonds [1967].

We can also formulate the third step using the following formula for the score w_e that each edge e gets assigned:

$$w_e = \sum_{\forall src} I(e \in tree_{src}) \,, \tag{4.4}$$

where the indicator $I(e \in tree_{src})$ is 1 if the edge e appears in the parse tree produced by the parser trained on the source language *src*, and 0 otherwise.

Weighted tree combination

As we have already noted, the method can be further enhanced by adding weighting. In our case, we use the $KL^{-4}_{cpos^3}$ source-target language similarity estimation; i.e., the same weight is applied to all edges in all parse trees produced by a parser for a given source language.

Thus, in the weighted variant of the method, the third step of the algorithm is modified by each source contributing not with 0 or 1 to the edge score, but with the value of its $KL^{-4}_{cpos^3}$ similarity to the target:

$$w_e = \sum_{\forall src} I(e \in tree_{src}) \cdot KL^{-4}_{cpos^3}(tgt, src)\,. \tag{4.5}$$

4.3 Evaluation

We now use the 18 test target language treebanks from the HamleDT 2.0 dataset to evaluate our methods, as opposed to the 12 development language treebanks which we used for tuning. All the 30 HamleDT 2.0 treebanks, with gold POS tags, were used to compute the $KL^{-4}_{cpos^3}$ similarity for each pair of languages, and to train delexicalized MSTperl parsers. Then, for each target language, the other 29 languages were used as potential sources. In the single-source method, only the delexicalized parser trained on the most similar source language treebank is applied to the target data. In the multi-source approach, all of the source delexicalized parsers are applied to the target data. Their outputs are then combined using the tree combination method, either with the contributions of each source weighted by its $KL^{-4}_{cpos^3}$ similarity to the target, or using equal weights for all sources.

Table 4.1 contains the results of applying the delexicalized parser transfer in several setups to the test target treebanks.

Our baseline is the treebank concatenation method of McDonald et al. [2011], i.e. a single delexicalized parser trained on the concatenation of the 29 source treebanks.

As an upper bound, we report the results of the oracle single-source delexicalized transfer: for each target language, the oracle source parser is the one that achieves the highest UAS on the target treebank test section. In this table, we do not include results of a higher upper bound of a supervised delexicalized parser (trained on the target treebank), which has an average UAS of 68.5%. It was not surpassed by our methods for any target language, although it was reached for Telugu, and approached within 5% for Czech and Latin.

The results show that KL_{cpos^3} performs well both in the selection task and in the weighting task, as both the single-source and the weighted multi-source transfer methods outperform the unweighted tree combination on average, as well as the treebank concatenation baseline. In 8 of 18 cases, KL_{cpos^3} is able to correctly identify the oracle source treebank for the single-source approach. In two of these cases, weighted tree combination further improves upon the result of the single-source transfer, i.e., surpasses the oracle. It also always reaches or

Target language	Treebank concatenation	Single-source oracle		Single-source KL_{cpos^3}			Tree combination w=1	Tree combination w=$KL^{-4}_{cpos^3}$
bn	61.0	te	**66.7**	0.5	**te**	**66.7**	63.2	**66.7**
cs	60.5	sk	**65.8**	0.3	**sk**	**65.8**	60.4	**65.8**
da	**56.2**	en	**55.4**	0.5	sl	42.1	**54.4**	50.3
de	12.6	en	**56.8**	0.7	**en**	**56.8**	27.6	**56.8**
en	12.3	de	**42.6**	0.8	**de**	**42.6**	21.1	**42.6**
eu	**41.2**	da	**42.1**	0.7	tr	29.1	**40.8**	30.6
grc	43.2	et	42.2	1.0	sl	34.0	**44.7**	42.6
la	38.1	grc	40.3	1.2	cs	35.0	**40.3**	39.7
nl	55.0	da	57.9	0.7	**da**	57.9	56.2	**58.7**
pt	62.8	en	64.2	0.2	es	62.7	**67.2**	62.7
ro	44.2	it	**66.4**	1.6	la	30.8	**51.2**	50.0
ru	55.5	sk	57.7	0.9	la	40.4	**57.8**	57.2
sk	52.2	cs	**61.7**	0.2	sl	58.4	**59.6**	58.4
sl	45.9	sk	**53.9**	0.2	**sk**	**53.9**	47.1	**53.9**
sv	45.4	de	**61.6**	0.6	da	49.8	**52.3**	50.8
ta	27.9	hi	**53.5**	1.1	tr	31.1	28.0	**40.0**
te	67.8	bn	**77.4**	0.4	**bn**	**77.4**	68.7	**77.4**
tr	18.8	ta	40.3	0.7	**ta**	40.3	23.2	**41.1**
AVG	44.5		**55.9**	0.7		48.6	48.0	**52.5**
Std.Dev.	16.9		10.8			14.4	15.0	11.8

Table 4.1: Evaluation using UAS on HamleDT 2.0 test target treebanks.

surpasses the single-best transfer as, in principle, it performs a soft weighted n-best transfer. This proves KL_{cpos^3} to be a successful language similarity measure for delexicalized parser transfer, and the weighted multi-source transfer to be a better performing approach than the single-source transfer.

The weighted tree combination is better than its unweighted variant only for half of the target languages, but it is more stable, as indicated by its lower standard deviation, and achieves an average UAS higher by 4.5% absolute. The unweighted tree combination, as well as treebank concatenation, perform especially poorly for English, German, Tamil, and Turkish, which are rich in determiners, unlike the rest of the treebanks: in the treebanks for these four languages, determiners constitute around 5-10% of all tokens, while most other treebanks contain no determiners at all. Thus, in the unweighted method, determiners are parsed rather randomly – UAS of determiner attachment tends to be lower than 5%, which is several times less than for any other POS. In the weighted methods, this is not the case anymore, as for a determiner-rich target language, determiner-rich source languages are given a high weight.

For target languages for which KL_{cpos^3} of the closest source language was lower or equal to its average value of 0.7, the oracle treebank was identified in 7 cases out of 12 and a different but competitive one in 2 cases; when higher than 0.7, an appropriate treebank was only chosen in 1 case out of 6. When KL_{cpos^3} failed to identify the oracle, weighted tree combination was mostly worse than unweighted tree combination. This shows that for distant languages, KL_{cpos^3} does not perform as good as for close languages.

5. Cross-lingual Lexicalization

So far, the basis for our cross-lingual parsing has been the delexicalized parser. However, omitting the lexical information leads to a noticeable drop in parsing accuracy, as the POS tags are often not sufficient by themselves to unambiguously expose the syntactic structure of the sentence. Therefore, in this chapter, we deal with lexicalizing the cross-lingual parsing.

There is in fact a very wide range of directions from which lexicalized cross-lingual parsing can be approached. However, here we only deal with the approach which we eventually chose, based on applying MT methods to the source language treebanks, automatically translating the source treebank into the target language, and then training a rather standard lexicalized parser on it. We explored various setups, and finally settled on the GIZA++ word aligner [Och and Ney, 2003] with intersection alignment and the Moses MT decoder [Koehn et al., 2007] in a monotone word-based setting, i.e. translating each source word to exactly one target word without any reordering.

Such an approach clearly has its limits, as even in close languages, words do not correspond 1:1 (let alone in distant languages) and systematic differences in word order are also common, making the word-based monotone translation sub-optimal in terms of intrinsic translation quality. However, we have found this approach to have two important benefits which seem to outweigh that. First, it makes the annotation transfer extremely simple and thus less noisy. And second, it forces the MT system to produce more literal translations, keeping the structure of the target sentence very similar to the source sentence, which increases the chance of the source annotation to be also valid for the translation.

5.1 Machine translation of source treebank

Tiedemann et al. [2014] introduce the approach of using a full-fledged SMT system to translate the word forms in a source language treebank (i.e. annotated source language sentences) into the target language. In this way, they again obtain a synthetic target-language treebank, which can be used to train a standard lexicalized parser.

Depending on the type of the MT system used, the transfer of the source sentence annotations onto the target language sentences produced by the MT system can be quite simple or quite complex. Tiedemann et al. [2014] explore multiple setups, but eventually decide for a complex SMT system, which requires the use of transfer heuristics similar to those of Hwa et al. [2005], and follow this path in their further works [Tiedemann, 2017]. In our work, on the contrary, we follow the other option which they had explored, using a word-based SMT system, which achieves a lower quality of the translation as measured in BLEU [Papineni et al., 2002], but makes it possible to transfer the source annotation by simply copying it over the 1:1 word alignment, thus avoiding the need for noisy transfer heuristics.

The approach of machine translating the source treebanks has the advantage of directly employing the manually annotated resources, which can be expected to be of high quality with only little noise. Of course, the translations provided by an

MT system are inherently noisy, containing mistranslations, untranslated words, etc., which is bound to introduce noise in the process. On the other hand, the MT outputs can be expected to structurally correspond better to the source sentences than human translations, as they tend to be more literal, which might actually make the annotation transfer more reliable than if human translations were used. Also, additional monolingual target-language texts are trivial to incorporate into this approach, using them to enrich the training data for the target language model.

5.2 Our setup

Our final setup is based on using *word-based* SMT in an otherwise rather standard setting[1] – employing GIZA++ word aligner [Och and Ney, 2003] with intersection alignment symmetrization, phrase table extraction with phrase length fixed to 1, Moses decoder [Koehn et al., 2007] in a monotone setting, and the KenLM language model [Heafield, 2011] with trigrams, trained on the target side of the parallel data.

5.3 Evaluation

In our final evaluation, we move to a more realistic setting, using the larger and more varied UD 1.4 dataset, employing the best cross-lingual tagging setup from Chapter 6 to obtain target POS tags, training the word-based monotone Moses MT system on the smaller out-of-domain WTC multiparallel data, and using $KL^{-4}_{cpos^3}$ to select and weight the top 5 source languages to use for processing each target language (or top 1 in the single-source transfer).

Table 5.1 shows the parsing accuracies in LAS for both delexicalized and lexicalized cross-lingual parser transfer in single-source and multi-source transfer, using either unweighted or weighted parse tree combination. For the single-source transfer, the highest obtainable results (i.e. the oracle) are also presented; however, as we did not train the Moses system for all existing language pairs, the lexicalized oracle may not always be the true oracle – we used the source language of the delexicalized oracle, unless the result for one of the "top 5" languages was higher, in which case we selected this one as the oracle source.

The first thing to notice from the results is that lexicalized parsing outperforms the delexicalized one for all methods and all target languages except for Turkish (and, in the case of the oracle, for Japanese), with the average improvement on all of the setups being around 3 LAS points. While this may not seem to be a lot, we would like to note that the difference between monolingual supervised lexicalized and delexicalized parsers for these languages is 8.8 LAS points on average, with the average lexicalized LAS score being 70.1%. With the cross-lingual parsing LAS being only about half of the supervised, we probably cannot hope to achieve a much larger improvement than 4.5 points on average. Thus, while there still seems to be room for improvement, we seem to already cover most of that gap with our approach.

[1] `http://www.statmt.org/moses/?n=Moses.Baseline`

Target lang.	Single-source transfer				Multi-source tree combination			
	Oracle		Automatic		Unweighted		Weighted	
	delex	lex	delex	lex	delex	lex	delex	lex
da	50.44	55.89	50.44	55.89	51.28	**58.31**	50.75	57.68
el	48.34	**52.31**	43.52	44.78	46.05	**50.72**	45.34	49.41
hu	28.93	30.33	25.96	28.40	30.64	**33.11**	30.62	32.57
id	37.48	40.01	35.32	39.06	37.97	**41.67**	37.32	41.11
ja	**19.89**	18.71	9.33	11.65	15.43	**16.04**	14.21	14.58
kk	11.45	**15.96**	11.45	11.45	12.50	**14.91**	12.65	13.86
lv	35.99	40.74	24.31	28.16	37.64	41.68	39.01	**42.23**
pl	55.93	**58.84**	51.79	54.10	54.03	57.59	54.84	**58.46**
sk	59.41	**65.75**	59.41	**65.75**	49.49	53.81	57.44	62.21
ta	17.50	**18.21**	17.50	**18.21**	10.37	10.45	15.44	16.39
tr	**25.23**	24.45	**25.23**	24.45	21.64	22.07	22.86	22.47
uk	44.40	47.30	44.40	47.30	44.40	46.89	44.81	**48.55**
vi	22.47	25.44	21.83	25.44	25.54	**28.10**	25.32	27.90
AVG	35.19	**38.00**	32.34	34.97	33.61	36.57	34.66	**37.49**
St.Dev.	15.61	17.02	16.25	17.52	15.33	16.92	15.66	17.39

Table 5.1: Evaluation of cross-lingual lexicalization on UD 1.4 treebanks subset with LAS, using WTC data, GIZA++ alignment, word based monotone Moses, cross-lingually induced POS tags, and sources selected using $KL^{-4}_{cpos^3}$. The top 5 sources are used in the tree combination. Best score and best non-oracle score are marked in bold, and reaching the oracle score in the single-source transfer is marked by underlining.

Interestingly, we can see that even though the word-based monotone MT could be expected to be particularly unsuitable for distant language pairs, we cannot make such conclusion from the results. While the absolute improvements in LAS brought by the lexicalization are clearly larger for Indo-European languages than for the non-Indo-European ones (which are all quite solitary in our dataset), the absolute LAS scores themselves are also higher. In relative terms, the lexicalization of the tree combination setup, on average, improves the LAS scores by 8.6% for the Indo-European target languages, and by 6.2% for the non-Indo-European ones, which does not seem to be a crucial difference. Moreover, the *median* relative improvements are even closer to each other: 8.3% for Indo-European and 7.8% for non-Indo-European targets. We thus conclude that the monotone word-based MT seems to be reasonably suitable even for very distant languages, which we find rather surprising.

Even though the weighted tree combination is usually not the best performing approach, it typically gets very close to it, losing on average only 1 LAS point towards the winner, and always less than 3.6 points. In this way, the $KL^{-4}_{cpos^3}$ weighting does perform its job well, evening out the results and thus preventing any huge losses (but also the wins). With the unweighted variant of the tree combination method, as well as with the single-source method, it is much more of a hit-or-miss, losing up to 12 LAS points towards the winner for the former, and up to 14 points for the latter. Thus, even though we acknowledge that it does not usually achieve the best result, we still conclude that the weighted lexicalized tree combination seems to be the best of the methods which we evaluated.

6. Cross-lingual Tagging

POS tagging is a very important prerequisite for syntactic parsing, both monolingual and cross-lingual. In NLP, POS tagging is a standard task on its own, with many applications, and is useful even without subsequent parsing. However, as our focus in this thesis is on parsing, we rather treat POS tagging only as a necessary pre-processing step in the parsing process. We thus do not present any particularly advanced or novel methods in this chapter. Instead, we build on pre-existing approaches of other authors, which we have also already seen applied to cross-lingual parsing in this thesis, and introduce some rather minor modifications and improvements into them. Specifically, we use POS projection over multiparallel data in Section 6.1, and machine translation of source treebanks in Section 6.2, which is mostly identical to the parser lexicalization via MT from Chapter 5.

6.1 Projection over (multi)parallel data

To the best of our knowledge, the first work on cross-lingual tagger induction is that of Yarowsky et al. [2001], who introduce the approach of projecting POS tags over parallel corpora. The main principle of their method of devising a tagger for an under-resourced target language, which we follow even in our work, is as follows:

1. train a POS tagger for a source language,
2. use it to tag the source side of a parallel corpus,
3. word-align the parallel corpus,
4. transfer the source POS tags over the word alignment links onto the target words,
5. train a target tagger on the, now tagged, target side of the parallel corpus.

The setting for the method fits perfectly our use case, as it utilizes just the resources that we assume to have, i.e. a source language treebank and a source-target sentence-aligned parallel corpus.

The method has been revisited and further improved by many authors, some of them devising remarkably sophisticated solutions. For example, Das and Petrov [2011] introduced a graph-based projection approach, constructing bilingual graphs with nodes corresponding to word types and word trigram types to define constraints for an unsupervised tagging model. In a somewhat related approach, Täckström et al. [2013] leveraged both bilingual texts and the Wiktionary lexicon to induce both token-level and type-level constraints, which were then used to provide a partial signal in training a partially observed conditional random field model. A wide range of other works exist in this field.

In our work, we mostly follow the approach of Agić et al. [2015, 2016], who observed the utility of exploiting multiparallel corpora, such as Bible (through Edinburgh Bible Corpus (EBC)) or WTC texts. The crucial advantage of such

resources is the fact that for each target sentence, there are aligned source sentences in multiple languages. These constitute a much more robust source of information, as, for each target word, there are multiple POS tag options based on the various aligned tagged source words. The authors find a simple majority voting mechanism to perform remarkably well, and gain further improvements by incorporating the alignment scores to obtain a weighted voting setup. The authors also show that respectable tagging accuracies can be obtained even with rather small parallel corpora based on religious texts, which, however, can be realistically expected to be available for many under-resourced target languages. This makes their work even more relevant for us.

6.2 Machine-translating the training data

The other approach to cross-lingual tagging that we explore is machine translation of the source treebanks, which is practically the same method as the one we used for cross-lingual parsing in Chapter 5, only applied to POS tagging.

The base approach can be summarized as follows:

1. train an MT system on source-target parallel data,
2. translate the word forms in a source treebank, keeping the POS annotation intact,
3. train a target POS tagger on the resulting synthetic target treebank.

Using MT to translate source training treebanks into the target language for cross-lingual POS tagger induction was introduced by Tiedemann [2014], applying the same approach to tagging and parsing. His setup was very similar to what we use in our work, featuring the GIZA++ word alignment and word-based Moses decoder with the KenLM language model.

In our work, we extended this method to a multi-source setting with source-weighting based on KL_{cpos^3}, in practically the same way as with parsing, and further improved it by using a simple self-training approach. Finally, we combine the translation approach with the projection approach using simple ensembling.

6.3 Influence on parsing

We now show the influence of improvements in the cross-lingual POS tagging to cross-lingual dependency parsing, evaluated within the best parsing setup from Chapter 5 (weighted parse tree combination of 5 closest source parsers, lexicalized using word-based monotone Moses treebank translation).

Table 6.1 compares the LAS achieved in parsing on top of POS tags provided by four of the tagging setups that we evaluated – unweighted combination of the taggers trained on machine translated treebanks for the 7 closest source languages, the weighted variant thereof, the self-training applied on top of the weighted combination, and the ensemble of the translation and projection approaches with self-training on top.

We can see that improvements in tagging accuracy generally tend to lead to corresponding improvements in parsing accuracy, both on average as well as for

Target lang.	Top 7 unw.	Top 7 w.	Retrain	Ensemble
Danish	59.11	**59.51**	59.37	57.68
Greek	47.69	47.74	48.70	**49.41**
Hungarian	26.96	29.95	30.79	**32.57**
Indonesian	37.16	37.88	38.72	**41.11**
Japanese	11.73	10.05	9.20	**14.58**
Kazakh	11.14	12.35	12.95	**13.86**
Latvian	37.53	38.05	41.10	**42.23**
Polish	55.45	56.00	57.51	**58.46**
Slovak	59.74	62.52	**63.46**	62.21
Tamil	14.96	13.06	14.57	**16.39**
Turkish	19.51	18.40	18.96	**22.47**
Ukrainian	51.87	51.04	**54.36**	48.55
Vietnamese	25.95	26.00	26.95	**27.90**
AVG LAS	35.29	35.58	36.66	**37.49**
AVG POSacc	69.55	69.73	71.10	**72.73**

Table 6.1: Cross-lingual parsing LAS based on the tagging setup used – unweighted combination of the top 7 sources, weighted combination of the top 7 sources, self-training using the weighted top 7 combination, and self-training on top of the ensemble setup. The last line lists the average POS tagging accuracy of the underlying cross-lingual tagging setup.

the individual target languages (although there is, as could be expected, some variance in the numbers). The weighted combination is slightly better than the unweighted one on average as well as for 8 of the 13 target languages. Self-training leads to an improvement of all but two targets and adds 1 LAS point on average. And ensembling with self-training leads to a further improvement for 10 of the 13 languages, adding another +0.9 LAS on average (for all the three target languages that experienced a deterioration in LAS, there was also a decrease of tagging accuracy).

On average, the table shows the tagging accuracy rising by 3.2 percentage points from the weakest setup to the strongest one, while the corresponding LAS rises by 2.2 points.

Most importantly, from the results we can see that improvements in tagging accuracy generally tend to lead to improvements in parsing accuracy, justifying the approach of dealing with tagging and parsing independently and then combining the best performing setups to form the final system.

Interestingly, we obtained best results when the parsers were trained on gold POS tags, even though better results are usually achieved by training on POS tags predicted by the tagger that will eventually be used in the inference, as this helps the parsers to know what POS tags they can expect and adapt to that. Moreover, this effect could be expected to be even stronger in the cross-lingual setting, where the inference-stage POS tags are typically considerably different from the gold POS tags. However, in our setup, the machine-translated treebanks which we use for training the parsers are actually quite different from the real target-language texts that both the taggers and the parsers will be eventually applied to, which probably weakens this effect.

Conclusion, or How to parse an under-resourced language

As a conclusion of the thesis, we summarize our findings in the form of a set of instructions for parsing an under-resourced target language for which no annotated data are available.

Get source treebanks

You will need harmonized dependency treebanks for some source languages; ideally for some languages which are close to the target language, but even distant languages can help. The Universal Dependencies treebank collection is currently the best such resource in existence.

Get parallel data

Obtain source-target parallel data for your target language and the source languages for which you have treebanks; multiparallel data are even better. The Watchtower texts seem to be very good for that purpose, and available for many under-resourced languages. Other religious texts are also available in many languages, especially the Bible.

Get monolingual target data

The target side of the parallel data can be used as monolingual target data. For some target languages, other larger monolingual data may be available, such as the Wikipedia texts.

Tokenize the parallel data

If the target language uses word spaces and punctuation, a simple rule-based tokenizer can be used. If not, a specialized tokenizer should be applied. For the source languages, a tokenizer can be trained on the source treebanks.

Align the parallel data

If the parallel data are not sentence-aligned, this has to be performed first; the Hunalign tool can be used for that. For word alignment, there is a range of existing tools, such as FastAlign or GIZA++. The intersection symmetrization should be applied to the produced alignment, as other symmetrizations are too noisy and also more difficult to work with.

Train source part-of-speech taggers

It is recommended to train a tagger that only predicts the coarse POS tags, as more fine-grained morphological labels do not seem to be sufficiently cross-lingual. We were satisfied with using the UDPipe tagger.

POS-tag the parallel data

Use the trained taggers to assign POS tags to the source sides of the parallel data.

Project POS tags over the word alignment

For each target token, determine its POS as the most frequent POS assigned to the source words aligned to it. This will give you initial target POS tags

Measure language similarity

Using the POS-tagged source and target sides of the parallel data, compute the $KL^{-4}_{cpos^3}$ language similarity for each source-target pair.

Train machine translation systems

For several of the source languages which seem to be the most similar to the target language (5 seems like a nice number), train a source→target MT system, such as Moses. Use only word-to-word translation without reordering. Employ a language model, trained on the target data you have.

Translate the source treebanks

Translate the word forms in the source treebanks into the target language. Using a word-based monotone MT system ensures that the annotation can be kept intact.

Train target word embeddings

Apply the word2vec to the target data to obtain target word embeddings. If you use a neural dependency parser in the next step, providing it with the pre-trained embeddings can help adapting it to the target language (but is typically useful even in a monolingual scenario).

Train tagger and parser

Train target taggers and parsers on the translated treebanks. We were satisfied with using UDPipe. Use the gold POS tags for training the parser.

Retag the target data

Retag the target data with the POS taggers trained on the translated treebanks. Determine the the final POS tag for each token by weighted voting, with the weight of the vote of each tagger determined by the $KL^{-4}_{cpos^3}$ similarity of its source language to the target language. Also include the current POS tags, obtained through projection over word alignment, into the voting, with a weight identical to that of the closest source.

Train the final tagger

Train the final tagger on the POS-tagged target data. Training the tagger on real target language data instead of the machine translation outputs makes it better. Also, it is more practical, as it results in a single standard parser that can be directly applied to target language texts, rather than having to perform the multi-source tagger combination each time.

Parse the target data

Tag the target data with the final tagger, and parse it with the dependency parsers trained on the translated treebanks. Score each possible target dependency edge by the sum of the $KL^{-4}_{cpos^3}$ similarities of the sources of all of the parsers that predicted that edge. Find the final parse tree by applying the Maximum Spanning Tree algorithm to the resulting weighted directed graph. For each dependent node, determine the label for the dependency relation to its head node through weighted voting on the labels predicted by all of the parsers, again using $KL^{-4}_{cpos^3}$ weights.

Optional: Train a final parser

A final parser can be trained on the parsed target data. However, in the case of the parser, this seems to actually decrease the accuracy, presumably because this way it gets trained on better target texts but worse POS tags; to achieve the highest accuracy of parsing, this step thus should probably be skipped. On the other hand, having one final parser which can be directly applied to target language sentences instead of a pool of parsers whose outputs need to be combined to get the parse tree may obviously be more handy in practice.

Use the final tagger and parser

Apply the final tagger and parser (or the parser combination) to any tokenized texts in the target language.

Expect low accuracy

The accuracy of the results depends on many factors, especially on the similarity of the available source languages to the target language and on the sizes of the available parallel data and source treebanks. Very roughly, with using the WTC multiparallel data, the accuracy of the POS tagger can be expected to be approximately $(70 \pm 20)\%$, and the labelled attachment accuracy of the parser approximately $(35 \pm 20)\%$. While this may be insufficient for most practical purposes, it is, to the best of our knowledge, about the best you can get.

Bibliography

Željko Agić. Cross-lingual parser selection for low-resource languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W17-0401`.

Željko Agić, Dirk Hovy, and Anders Søgaard. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*. Hrvatska znanstvena bibliografija i MZOS-Svibor, 2015.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Alonso Martínez, Natalie Schluter, and Anders Søgaard. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301, 2016.

Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Yoeng-Jin Chu and Tseng-Hong Liu. On shortest arborescence of a directed graph. *Scientia Sinica*, 14(10):1396, 1965.

Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics, 2011.

Marie-Catherine de Marneffe, Natalia Silveira, Timothy Dozat, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC'14*, Reykjavík, Iceland, 2014. ELRA.

Matthew S. Dryer and Martin Haspelmath, editors. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL `http://wals.info/`.

Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240, 1967.

Jason M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 340–345, Stroudsburg, PA, USA, 1996. ACL. doi: 10.3115/992628.992688. URL `http://dx.doi.org/10.3115/992628.992688`.

Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:11–311, 2005.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume, Proceedings of the Student Research Workshop, Proceedings of Demo and Poster Sessions, Tutorial Abstracts*, pages 177–180, Praha, Czechia, 2007. Univerzita Karlova v Praze, Association for Computational Linguistics. ISBN 978-1-932432-87-9.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86, 1951.

Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, 2016.

Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97, 2013.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal Dependencies 1.0, 2015. URL `http://hdl.handle.net/11234/1-1464`. `http://hdl.handle.net/11234/1-1464`.

Joakim Nivre et al. Universal dependencies 1.4, 2016. URL `http://hdl.handle.net/11234/1-1827`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre et al. Universal dependencies 2.1, 2017. URL `http://hdl.handle.net/11234/1-2515`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Rudolf Rosa. MSTperl parser (2015-05-19), 2015. URL `http://hdl.handle.net/11234/1-1480`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rudolf Rosa and Zdeněk Žabokrtský. MSTParser model interpolation for multi-source delexicalized transfer. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 71–75, Stroudsburg, PA, USA, 2015a. Euskal Herriko Unibertsitatea, Association for Computational Linguistics. ISBN 978-1-941643-98-3.

Rudolf Rosa and Zdeněk Žabokrtský. KL_{cpos^3} – a language similarity measure for delexicalized parser transfer. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Short Papers*, Stroudsburg, PA, USA, 2015b. Association for Computational Linguistics.

Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žabokrtský. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of LREC 2014*, pages 2334–2341, Reykjavík, Iceland, 2014. ELRA. ISBN 978-2-9517408-8-4.

Kenji Sagae and Alon Lavie. Parser combination by reparsing. In *Proceedings of HLT-NAACL*, pages 129–132. ACL, 2006.

Milan Straka, Jan Hajič, and Jana Straková. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Paris, France, May 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12, 2013.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *LREC*, volume 2012, pages 2214–2218, 2012.

Jörg Tiedemann. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1854–1864, August 2014.

Jörg Tiedemann. Cross-lingual dependency parsing for closely related languages – Helsinki's submission to VarDial 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain, 2017.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. Treebank translation for cross-lingual parser induction. In *Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*, 2014.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1072133.1072187. URL `https://doi.org/10.3115/1072133.1072187`.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 213–218, Marrakech, Morocco, 2008. European Language Resources Association. ISBN 2-9517408-4-0.

Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, 2008. Asian Federation of Natural Language Processing, International Institute of Information Technology.

Daniel Zeman, David Mareček, Jan Mašek, Martin Popel, Loganathan Ramasamy, Rudolf Rosa, Jan Štěpánek, and Zdeněk Žabokrtský. HamleDT 2.0, 2014. URL `http://hdl.handle.net/11858/00-097C-0000-0023-9551-4`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, jr. Jan Hajič, Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin,

Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Stroudsburg, PA, USA, 2017. Charles University, Association for Computational Linguistics. ISBN 978-1-945626-70-8. URL `http://www.aclweb.org/anthology/K/K17/K17-3001.pdf`.

List of Publications

Joakim Nivre et al. Universal dependencies 1.4, 2016. URL `http://hdl.handle.net/11234/1-1827`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rudolf Rosa. Automatic post-editing of phrase-based machine translation outputs. Master's thesis, Charles University in Prague, Faculty of Mathematics and Physics, Praha, Czechia, 2013. 5 citations.

Rudolf Rosa. MSTperl parser (2015-05-19), 2015a. URL `http://hdl.handle.net/11234/1-1480`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rudolf Rosa. MSTperl delexicalized parser transfer scripts and configuration files, 2015b. URL `http://hdl.handle.net/11234/1-1485`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rudolf Rosa. Multi-source cross-lingual delexicalized parser transfer: Prague or Stanford? In Eva Hajičová and Joakim Nivre, editors, *Proceedings of the Third International Conference on Dependency Linguistics, Depling 2015*, Uppsala, Sweden, 2015c. Uppsala University, Uppsala University. 5 citations.

Rudolf Rosa. MonoTrans: Statistical machine translation from monolingual data. In Jaroslava Hlaváčová, editor, *Proceedings of the 17th conference ITAT 2017: Slovenskočeský NLP workshop (SloNLP 2017)*, volume 1885 of *CEUR Workshop Proceedings*, pages 201–208, Praha, Czechia, 2017a. ÚFAL MFF UK, CreateSpace Independent Publishing Platform. ISBN 978-1974274741.

Rudolf Rosa. Terminal-based CoNLL-file viewer, v2, 2017b. URL `http://hdl.handle.net/11234/1-2514`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

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 and Zdeněk Žabokrtský. MSTParser model interpolation for multi-source delexicalized transfer. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 71–75, Stroudsburg, PA, USA, 2015a. Euskal Herriko Unibertsitatea, Association for Computational Linguistics. ISBN 978-1-941643-98-3. 3 citations.

Rudolf Rosa and Zdeněk Žabokrtský. KL_{cpos^3} – a language similarity measure for delexicalized parser transfer. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Short Papers*, Stroudsburg, PA, USA, 2015b. Association for Computational Linguistics. 12 citations.

Rudolf Rosa and Zdeněk Žabokrtský. Error analysis of cross-lingual tagging and parsing. In Jan Hajič, editor, *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 106–118, Praha, Czechia, 2017. Univerzita Karlova, Univerzita Karlova. ISBN 978-80-88132-04-2.

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, 2012. ACL. ISBN 978-1-937284-38-1. 3 citations.

Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žabokrtský. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of LREC 2014*, pages 2334–2341, Reykjavík, Iceland, 2014. ELRA. ISBN 978-2-9517408-8-4. 20 citations.

Rudolf Rosa, Daniel Zeman, David Mareček, and Zdeněk Žabokrtský. Slavic forest, Norwegian wood (scripts), 2017a. URL `http://hdl.handle.net/11234/1-1970`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rudolf Rosa, Daniel Zeman, David Mareček, and Zdeněk Žabokrtský. Slavic forest, Norwegian wood. In Preslav Nakov, Marcos Zampieri, Nikola Ljubešić, Jörg Tiedemann, Shervin Malmasi, and Ahmed Ali, editors, *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial4)*, pages 210–219, Stroudsburg, PA, USA, 2017b. Association for Computational Linguistics, Association for Computational Linguistics. ISBN 978-1-945626-43-2. 2 citations.

Daniel Zeman, David Mareček, Jan Mašek, Martin Popel, Loganathan Ramasamy, Rudolf Rosa, Jan Štěpánek, and Zdeněk Žabokrtský. HamleDT 2.0, 2014. URL `http://hdl.handle.net/11858/00-097C-0000-0023-9551-4`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Only publications used in the thesis are listed. Numbers of citations are according to Google Scholar, excluding self-citations by any of the authors.