OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Cross-lingual Natural Language Processing with Linguistic Typology Knowledge

Chinmay Choudhary
15232114

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisor: Dr. Colm O'Riordan

Internal Examiner: Dr. Bharathi Raja Chakravarthi
External Examinar: Dr. Edoardo M. Ponti

School of Computer Science
University of Galway
Galway, Ireland
July, 2023

Acknowledgements

Undertaking this PhD project at *University of Galway, Ireland* has been a life-changing experience for me. It would not have been possible to complete this PhD project without the guidance and support of numerous people during the course of my PhD.

First of all, I would like to give a big thank you to my supervisor **Dr. Colm O'Riordan** for all his support and encouragement during the course of my PhD. Without his guidance and constant feedback this PhD would not have been achievable. I would specifically like to thank him for being extremely supportive and helpful during the period of COVID lockdown.

I would also like to thank **Dr. Frank Glavin** for his valuable guidance specifically during the early days of my PhD, when I was choosing my thesis topic.

I highly appreciate the **Hardiman Research Scholarship** provided by the University of Galway, Ireland to support my PhD for the entire duration of four years. I would like to thank **Dr. Lucy Byrnes**, Dean of Graduate Studies as well as **Ms Sandra Donohue**, Graduate Studies Administrator Dean of Graduate Studies Office for helping out with my application processes, and finally realising of the funding opportunity for me.

I wish to thank **Dr Seamus Hill**, **Dr Josephine Griffith** and **Dr Owen Molloy** as members of my *Graduate Research Council* for evaluating my progress every year and providing with feedback, encouragement and valuable suggestions for the future course of the project.

I greatly appreciate **Daniel Kelly** and **Raymond Conlin** as the fellow researchers in my lab for valuable reviews, suggestions and feedbacks. Our free-time coffee discussion and our weekly meetups to discuss each other's progress so far have been extremely valuable for me during the course of my PhD.

I would also like to thank **Prof. Paul Buitelaar** and **Dr. John McCrae** as well as the entire NLP research group at the *Insight SFI Research Centre for Data Analytics* for providing me frequent opportunities to present my work to the group

and receive valuable feedbacks.

My thanks also goes out to my team-head **Micheal Mackey**, my supervisor **Abhishekh Khanna** as well as my colleagues at the *Software Development Networking* group at the *Huawei Ireland Research Center* for not only providing me with the opportunity to do research internship and gain industry experience during the course of my PhD but also to provide valuable suggestions on my PhD research project.

I am also very grateful to *William Johnson*, the CTO of *Rugby Smarts* and *Neil Haran*, the founder and CEO of *Kappture.co.uk*, as well as my colleagues at *Rugby Smarts* and *Kappture.co.uk* for providing me with the opportunity to gain part-time industry experience during the course of my PhD, as well as providing suggestions about my PhD project.

I am indebted to all to my teachers at University of Galway and University of Edinburgh, who gave me very strong foundation in the field of NLP and IT. Furthermore, I have been seeking advice/feedback from these faculty from time to time.

Finally, I would also like to say a heartfelt thank you to my Mother, Father and my Fiance for always believing in me and encouraging me to pursue my PhD.

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Chinmay Choudhary
15232114

Abstract

State-of-the-art approaches to most Natural Language Processing (NLP) tasks have achieved near human performance. This recent progress has positively impacted millions of lives and businesses around the world. However, these approaches are neural-network based supervised approaches that require large manually annotated datasets to be trained on. Such datasets are available in only a handful (less than 1%) of high-resource languages. Hence, most of the world's population is still excluded from the benefits of NLP.

The most promising class of approaches proposed by researchers to address this issue of data-sparsity in low-resource languages is *Cross-lingual Model Transfer* approaches. These approaches typically involve training a neural-network model using a high-resource language called *Source language* and adapting it to a low-resource language called *Target language* using cross-lingual/multilingual word-representations. Although these Cross-lingual Model Transfer approaches sufficiently outperform all other types of approaches to various NLP tasks for low-resource languages (such as Cross-lingual Data-transfer approaches, Unsupervised approaches etc.), still they significantly under-perform fully supervised approaches trained on abundant data. In this work we utilised the linguistic typology knowledge available in various open-source typology databases to improve the performances of state-of-the-art Cross-lingual Model Transfer approaches to four key intermediate NLP tasks namely *Constituency Parsing*, *Dependency Parsing*, *Enhanced Dependency Parsing* and *Semantic Role Labelling*.

Linguistic typology is the field of linguistics that aims to study and classify all the world's languages based on their syntactic, semantic and phonological properties. There are numerous publicly available typology databases such as WALS, URIEL, ValPal etc. that provide a taxonomy of typological features and their possible values as well distinct feature-value for each language. These databases are created by the contributions of numerous linguistics over the decades, primarily to study the similarities and distinctions among world's languages. However, in this work we argue that this typology knowledge can also be utilised by the CLT models to improve their performance. Thus, we propose and evaluate novel cross-lingual approaches to numerous NLP tasks that utilise typology knowledge in this work. We also propose and evaluate various frameworks to inject the typology knowledge available in various open-source databases into the modern neural-network architectures.

Contents

List of Figures

List of Publications

1. **Multitasking End-to-end BERT based Cross-lingual Dependency Parser.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at EACL 2023 (Under Review)

2. **Cross-lingual Semantic Role Labelling with the Valpal database knowledge.** In Proceedings of THE 3RD WORKSHOP ON KNOWLEDGE EXTRACTION AND INTEGRATION FOR DEEP LEARNING ARCHITECTURES (DEEPLIO) AT ACL 2022

3. **Universal Recurrent Neural Network Grammar.** In Proceedings Of 33RD ANNUAL CONFERENCE ON COMPUTATIONAL LiNGUISTICS AND SPEECH PROCESSING (ROCLING) 2021

4. **Improving the Performance of UDify with Linguistic Typology Knowledge.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at NAACL 2021

5. **End-to-end mBERT based Seq2seq Enhanced Dependency Parser with Linguistic Typology knowledge.** In Proceedings of SPECIAL INTEREST GROUP ON NATURAL LANGUAGE PARSING (SIGPARSE) AT ACL 2021

6. **NUIG: Multitasking Self-attention based approach to SigTyp 2020 Shared Task.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at EMNLP 2020

Chapter 1

Introduction

The field of Natural Language Processing (NLP) has shown tremendous progress in the last few years. This is due to the significant advancement made in the field of deep-learning/neural-networks which has made it possible to statistically model complex linguistic rules and patterns, related to various sophisticated computational linguistic tasks. In fact, the state-of-the-art neural-network (NN) based approaches to most of the widely used NLP tasks have achieved near human performance Turc et al. (2019); Vaswani et al. (2017). These tasks include among others Machine Translation, Web-based Question Answering, Information Extraction, Automatic Speech Recognition, Legal or Financial Document Analysis, Chatbot, Speech synthesis, Speech-to-speech translation and Text-mining.

This recent progress in NLP has positively impacted millions of lives and businesses around the world. However, although the research community has mostly conquered the issue of mathematically modelling the extremely complex NLP tasks, these neural-network models still require a large amount of manually annotated gold-standard dataset to train and optimize its model-parameters. The lack of such datasets limits their utility to only a few high-resource languages.

According to the Glottolog database[1] there are over 7751 languages in the world. Out of these languages only a handful (less than 1% in most cases) of languages are high-resource languages that possess sufficiently large manually annotated datasets to train the state-of-the-art model for the respective NLP task being performed Magueresse et al. (2020). Although the count of high-resource languages varies from task to task, yet in most of the cases, this set of high-resource languages include western European languages such as English, German, French, Italian etc. as well as Asian languages such as Arabic, Chinese, Japanese, Hindi Korean etc. Creating a sizeable

[1] https://glottolog.org/

training dataset for any NLP task in any low-resource language is challenging as it requires enormous financial resources as well as skilled labor. Furthermore, given the broad range of languages and tasks, complete coverage is almost impossible. Therefore, a large section of the world's population are still devoid of the benefits of the recent advancements achieved in the NLP field.

Researchers have attempted numerous types of approaches to address this issue of data-sparsity in low-resource languages. Earlier approaches include developing unsupervised models that do not require manually annotated datasets to be trained on. However, it is observed that these unsupervised models significantly underperform as compared to the trained supervised model. Another class of approaches called Cross-lingual Data Transfer (section 2.1.1.1) approaches have been proposed that aim to generate datasets in low-resource languages from the datasets available in high-resource languages using techniques such as Machine Translation (section 2.1.2.2) and Annotation Projection (section 2.1.2.1). These approaches show impressive performance for some significant tasks, but their application is extremely limited by the requirement of numerous cross-lingual resources such as parallel aligned corpora, bilingual lexicon etc. We will provide a brief literature review of all these approaches in the chapter 2.

Hence so far, the most promising class of approaches to effectively address the issue of data-sparsity in low-resource languages is Cross-lingual Model Transfer approaches. These approaches typically involve training a NN model using a high-resource language called *Source language* and adapting it to a low-resource language called *Target language*. The model parameters are adapted from source to target learning using cross-lingual word-representation learnings. In some cases, these approaches are applied in multilingual settings where the model is trained on a mixed polyglot corpus of high-resource languages and adapted to a single low-resource transfer language. Multi-lingual representations are used in this scenario. We will describe the cross-lingual model transferring in details in chapters 3 to 7.

The *Cross-lingual Model Transfer* based approaches show significant performance for most of the NLP tasks and only require unannotated plain text corpora in the low-resource language, thus increasing applicability to a large pool of low-resource languages. However, these approaches also suffer from one significant limitation. A *Cross-lingual Model Transfer* approach shows extremely high performance when the source and target languages are genealogically and typologically (and even geographically) closer to each other, but performance drops significantly when the languages are apart Ammar et al. (2016). For example, a cross-lingual model trained on the

source-language English would perform very well on the target-language German, and a cross-lingual model trained on the source-language Swedish would perform very well on the target-language Danish, but a cross-lingual model trained on the source English would perform poorly on the target-language Chinese. Hence these approaches have limited utility for many of the low-resource languages. In this thesis, we aim to address this issue by using linguistic typology knowledge.

Linguistic typology is the field of linguistics that aims to study and classify all the world's languages based on their syntactic, semantic and phonological properties. Typology research work involves identifying such features that can uniquely define most of the languages in the world. There are numerous publicly available typology databases (section 2.2.3) that provide a taxonomy of typological features and their possible values, the hierarchy among these features as well distinct feature-value for each language. These databases are created by linguistics over decades primarily to study the similarities and distinctions among the world's languages. However such typology databases can be utilised to improve the cross-lingual transferring ability of CLT based models from high-resource source language to low-resource target language, specifically in scenarios where source and target languages are genealogically and typologically apart.

Hence, the overall high-level research-objective of this thesis can be stated as follows:

Integrate the linguistic typology knowledge in the publicly available typology databases with the state-of-the-art neural network based cross-lingual/multilingual approaches to numerous NLP tasks

Although there are a handful of researchers that used linguistic typology with various cross-lingual/multilingual NLP models. We will provide an review of these approaches in subsequent chapters.

The application of typology with cross-lingual NLP in the past-work is very limited and is mainly confined to only morphological and word-order typology features from WALS database[2].

Our thesis is a wide-scope thesis. Hence, in this thesis we will experiment with numerous typology databases, typology feature-types and explore numerous different NLP tasks, instead of focusing deep on a single task and a single typology database.

[2]https://wals.info/

1.1 Research Objectives and Questions

In this thesis, we aimed to improve the performance of cross-lingual neural-network based approaches to numerous significant NLP with Linguistic typology knowledge available in various publicly available typology databases, specifically in the scenarios where source and target languages are genealogically, typologically, and geographically languages very distinct from each other. We experimented with four significant NLP tasks namely *Constituency Parsing*, *Dependency Parsing*, *Enhanced Dependency Parsing* and *Semantic Role Labelling*. These are intermediatory tasks that aid all the downstream end-user tasks such as Machine Translation, Information Retrieval, Chatbot, Question Answering etc.

Sections 1.1.1, 1.1.2, 1.1.3 and 1.1.4 provide a brief overview of each of the four NLP tasks. These sections will also describe the typology databases utilized as well as the knowledge injection mechanisms adopted for each of the four tasks. Furthermore, in these sections we will formally outline our research objectives and the research questions addressed, with respect to each of the four tasks.

1.1.1 Constituency Parsing

Constituency parsing (CP) is the task of autonomously extracting a phrase-based parse tree from a given sentence (section 3). Each node of such tree spans over a specific phrase within the input sentence, that describe either a single semantic unit (eg: time of some action) or a single syntactic unit (eg: Subject of main verb). A constituency parse-tree simply represents the hierarchy of all such phrasal nodes that exists in the given sentence. The root of the tree spans upon entire sentence and is generally labelled as S.

Recurrent Neural Network Grammar (RNNG) Dyer et al. (2016) is a state of the monolingual approach to constituency parsing task. In this segment of our research work, we evaluated the performance of cross-lingual variant of the RNNNG (CL-RNNG) model on numerous target-languages in both few-shot and zero-shot learning settings.

Subsequently we proposed Universal RNNG (UniRNNG) which is a modified version of CL-RNNG which utilises linguistic typology knowledge in WALS dataset[3] to improve cross-lingual transferring. We feed-in the typology features directly along with word representations. Overall, this segment aims to address following research questions.

[3]https://wals.info/

RQ1: Can the state-of-the-art Recurrent Neural Network Grammar (RNNG) approach to monolingual Constituency parsing be applied for cross-lingual Constituency parsing ?

RQ2: Within the cross-lingual transfer-learning settings, does mixed polyglot training lead to improvement in performance of the RNNG model, as compared to single source language training ?

RQ3: Does the performance of RNNG model within cross-lingual transfer-learning settings be improved by injecting the linguistic typology knowledge into it ?

1.1.2 Dependency Parsing

Dependency Parse-tree (DP) is another prominent framework to represent the syntax of a sentence, which is very distinct from the constituency parse-tree framework. Unlike the Constituency Parse-tree which represents the syntax of a given input sentence as a hierarchy of phrase-structure nodes, a dependency parse-tree on the other hand represents the syntax of a given input sentence as a set of word-pairs (section 5.1). Each such word-pair comprises a head-word and a dependent word, thereby depicting a single dependency-relationship. Both head-word and dependent word can be located anywhere within the input sentence. In case of the labelled dependency parse-tree each such dependency-relationship is also assigned a label indicating its type. Dependency Parsing is the task of autonomously generating a dependency parse-tree for a given input sentence. The dependency parse-tree is generated based on the Dependency grammar (DG) of the language being parsed, which simply comprises of all the possible word-level binary relationships that can exist in that language.
There are numerous approaches to dependency parsing been proposed and evaluated. These approaches can be classified into three categories namely *Transition-based approaches*, *Graph-based approaches* and *End-to-end approaches* (section 5.1.3). Both statistical and neural-network based monolingual approaches belonging to each of these categories have been proposed by researchers. In subsequent chapters we will provide a detailed literature review of the dependency-parsing task.
The *End-to-end approaches* are much simpler, easier to implement and resource-efficient while performing at par with the *Transition-based approaches* and the *Graph-based approaches*. However, in cross-lingual settings, most state-of-the-art approaches

to the dependency-parsing are *Graph-based approaches*. In our thesis, we proposed an *End-to-end BERT Based Dependency Parser* which can parse a sentence by directly predicting relative head-position tag for each word within input sentence. We evaluated this proposed model in both mono-lingual and cross-lingual/multilingual setups (using Multilingual BERT).
Subsequently, we aimed to improve the performance of our proposed *End-to-end BERT Based Dependency Parser* (section 5.4.1) in cross-lingual settings by utilising the linguistic typology knowledge available in URIEL database Littell et al. (2017). We injected this typology knowledge into the model using multitasking framework. Within the same segment we also re-implemented the state-of-the-art UDify model Kondratyuk and Straka (2019b). Subsequently, we injected the same linguistic typology knowledge available in URIEL database within the UDify model and re-evaluated the performance. Similar to the proposed *End-to-end BERT Based Dependency Parser*, we used the multitasking mechanism to inject this typology knowledge into UDify model.
Overall this segment of our research-work addresses following research questions.

RQ4: Does an End-to-end Dependency parser performs at par with the state-of-the-art Graph-based parser, within both monolingual and cross-lingual settings ? RQ5: Does injecting linguistic typology knowledge into an End-to-end cross-lingual dependency parser, through an auxiliary task of typology feature-value prediction, leads to improvement in performance of it ?

RQ6: Is the impact of adding the auxiliary task of typology feature-value prediction higher with mixed polyglot training scenerio, as compared to single source language training scenerio ?

RQ7: For the state-of-the-art UDify parser which is a multilingual multi-tasking model that performs four key tasks simultaneously namely UPOS-tagging, UFeat-tagging, Lammetization and Dependency-parsing, when an auxiliary task of typology feature-value prediction is added to it, does it impact the performances of other four NLP tasks ?

RQ8: Is there any correlation between the performance the end-to-end

parser on the main dependency-parsing task and the performance of it on the auxiliary task of linguistic typology feature-value prediction ?

1.1.3 Enhanced Dependency Parsing

The Enhanced Dependency Parsing (EDP) framework is an extension of the standard Dependency Parsing framework discussed in 1.1.2, which provides additional syntactic and semantic attributes that are missing in a standard dependency parse-tree. It is observed that such additional attribute knowledge does lead to improvement in performance on numerous downstream NLP tasks. For a given input sentence its standard Dependency Parse-tree is simply a subset of its Enhanced Dependency-tree Schuster and Manning (2016). Enhanced Dependency-parsing (EDP) is the task of autonomously generating the enhanced dependency parse-tree from a given input sentence. The Enhanced Dependency framework is proposed recently in 2015, hence there are only limited approaches to EDP been proposed in both the monolingual and cross-lingual settings.

In this segment of thesis, we proposed and evaluated a neural-network based approach to the end-to-end enhanced dependency-parsing. Our proposed model is an extension of the UDify model (section 1.1.2) for standard Dependency-parsing task with an addition auxiliary task of end-to-end EDP task. Subsequently, we injected the linguistic typology knowledge available in URIEL database into this proposed EDP model and observed the improvement in performance in multiple settings.

Thus in this segment we aim to address following research questions.

RQ9: Does the cross-lingual mBERT based End-to-end Enhanced Dependency Parser perform at par with various state-of-the-art cross-lingual approaches to the enhanced dependency parsing task ?

RQ10: Does linguistic typology knowledge injection into a cross-lingual mBERT based End-to-end Enhanced Dependency Parser improves its performance ? Is it better to feed-in the linguistic typology knowledge into the model directly along with word-representations, or to inject typology knowledge though an auxiliary task ?

1.1.4 Semantic Role Labelling

Semantic role labeling (SRL) is the task of identifying various semantic arguments (such as Agent, Patient, Instrument, etc.) for each of the target verb (predicate) within an input sentence (section 7.1). For a given input-sentence, a typical semantic role labeling approach aims to assign distinct labels to various words and phrases in the input sentence. Each assigned label indicates a unique semantic role. In case of a multi-predicate words, the approach performs labelling of all the words in the input-sentence for each predicate independently. SRL is useful as an intermediate step in numerous end-user semantic tasks such as Document categorization, Text-summarizing, Question-answering etc.
Numerous approaches to SRL in both monolingual and cross-lingual settings, has been proposed by the researchers (section 7.2). We re-implemented a state-of-the-art recurrent neural-network (RNN) based approach to cross-lingual SRL task proposed Cai and Lapata (2020). It is a comprehensive approach comprising two distinct neural-networks namely the *Semantic Role Labeller* and the *Semantic Role Compressor*. The approach represents the input-sentence as a sequence of word-embeddings generated by a popular publicly available pre-trained language model called BERT.
In this segment we injected the semantic typology knowledge available in the Valency Patterns Leipzig (ValPal) database into the respective cross-lingual SRL approach to improve its performance. We represented the entire typology knowledge about the target language within the Valpal database as a set of First-order-logic rules. Subsequently, we utilised the *Deep Probabilistic Logic* framework to fine-tune the *Semantic Role Labeller* component with the First-order-logic constraints.
Thus in this segment we aim to address following research questions.

RQ11: Does the performance of a simple BiLSTM model for the Semantic Role Labelling task improve, when the semantic typology knowledge of the target-language available in the ValPal database, is injected into it, within monolingual and polyglot training training scenerio ?

RQ12: Does the impact of injecting the ValPal database knowledge into the state-of-the-art cross-lingual BiLSTM based model for the Semantic Role Labelling task increases due to joint polyglot training as compared to the mono-lingual training ?

RQ13: Does extending the verb-inventory of ValPal database for a specific

target-language with other lexical databases (such as FrameNet and VerbNet) before injecting this ValPal knowledge into the cross-lingual BiLSTM based model for the Semantic Role Labelling, increases the impact of this knowledge injection ?

1.2 Chapter Outline

Section outlines different NLP tasks that are experimented with, as part of our broad-based thesis. Research work conducted with respect to each of these tasks are described as different chapters in this thesis. This section outlines a high-level overview all subsequent chapters in the thesis.

Chapter 2: This chapter will provide a comprehensive literature-review covering all the previously published work relevant to our thesis. This would include detailed overview of approaches to NLP for low-resource languages including unsupervised and cross-lingual approaches, the linguistic typology field as well as various open-source linguistic-typology databases, the previously published approaches to predict the missing typology knowledge and the previously published cross-lingual transfer-learning approaches to low-resource NLP that utilized the linguistic typology knowledge to improve their performance.

Chapter 3: This chapter will describe the Constituency Parsing task and the Constituency Grammar in details, including a detailed review of previous published approaches to both monolingual and cross-lingual Constituency Parsing. Subsequently, the chapter will describe our proposed UniRNNG model which utilises linguistic typology knowledge in WALS database. The chapter will provide details about experiments conducted to evaluate the proposed UniRNNG and the results achieved.

Chapter 4: This chapter will propose and describe a multitasking model to predict the WALS typology features for various languages. We proposed the model as a solution to the *SigTyp 2020 Shared Task* Bjerva et al. (2020). This proposed model forms the basis to our cross-lingual approaches to the Dependency-parsing task that we propose and describe in chapter 5. The chapter will provide details about experiments conducted to evaluate this proposed multitasking model and will discuss the results achieved.

Chapter 5: This chapter will describe the Dependency Parsing task and the Dependency Grammar in details, including a detailed review of previous published approaches to both monolingual and cross-lingual Constituency Parsing. The will describe two distinct cross-lingual models for dependency parsing namely *End-to-end BERT Based Dependency Parser* and *UDify*. Subsequently, the chapter describes the multitasking based mechanism to inject linguistic typology knowledge available in URIEL database into both of these models. The chapter will provide details about numerous experiments conducted to evaluate both dependency parsers and will discuss the results obtained, under different settings.

Chapter 6: This chapter describes the Enhanced Dependency Parsing framework, outline the distinction between standard and enhanced Dependency parsing and will provide a litrature review of numerous approaches to Enhanced Dependency Parsing task. Subsequently, the chapter will propose an end-to-end multitasking Enhanced Dependency Parser which is inspired by the *UDify* parser discussed in chapter 5. The parser uses linguistic typology knowledge provided in URIEL database. The chapter will provide details about experiments conducted to evaluate the proposed Enhanced Dependency parser and will discuss the results obtained.

Chapter 7: This chapter describes the Semantic Role Labelling task as well as provide a literature review of various previously published approaches to both monolingual and cross-lingual semantic role labelling. Subsequently, the chapter describes the Valency Patterns Leipzig (ValPal) database. The chapter will also describe describe mechanism to convert entire valpal database knowledge as first-or-logic rules and inject it into the state-of-the-art cross-lingual semantic role labeller. Finally, the chapter will provide details about experiments conducted and the results achieved.

Chapter 8: This chapter will conclude the thesis. The chapter will review the research objectives and questions outlined in this chapter, outcomes achieved and will provide directions for future research.

Chapter 2

Related Work

In this chapter, we review the previously published work related to our dissertation. We divide the entire literature review into four segments. Firstly, in section 2.1 we categorize and provide a high-level overview of various approaches to NLP for low-resource languages including cross-lingual transfer learning approaches. In section 2.2, we discuss the research work related to linguistic typology and universals as well as list various open-source linguistic-typology databases. In this section we also discuss the issue of missing typology feature-values in most open-source typology databases which limits their utility. Subsequently, in section 2.3 we discuss the proposed ML based approaches to predict the missing typology knowledge in these public databases. Finally in section 2.4 we discuss the previously published cross-lingual transfer-learning approaches to low-resource NLP that utilized the linguistic typology knowledge to improve their performance.
This review is strongly inspired by and extends the review-work published by Ponti et al. (2019).

2.1 NLP for Low-resource languages

Most state-of-the-art neural network approaches to standard NLP tasks such as parsing, machine translation, question answering, information retrieval etc. show near human performance. However, these approaches are supervised approaches that require large manually annotated datasets to be trained on. Such datasets are available in only a handful of high-resource languages such as English, Arabic, Chinese etc. Hammarström et al. (2017). Hence, despite great advancements made in the field of NLP over the last decade, most of the world's population is still excluded from the benefits of NLP. There is no standard definition of a high or low resource language. Hence such categorisation of world's languages is very subjective and varies

from task-to-task. The creation of gold-standard linguistic resources and datasets for these low-resource languages is an expensive and time-consuming process which requires skilled labor. Furthermore, the wide range of languages and the possible NLP tasks makes the complete coverage unrealistic.

Earliest works by researchers to address this issue of data-sparsity in low-resource languages include various **unsupervised learning approaches** (Snyder and Barzilay (2008); Vulić et al. (2011); Titov and Klementiev (2012); inter alia). These approaches entirely abandon the use of annotated datasets and instead aim to build probabilistic models for various NLP tasks based on known linguistic knowledge as well as the observed patterns within the unlabeled text. However, these approaches significantly underperform the state-of-the-art supervised learning approaches Täckström et al. (2013) and are rarely combined with typological knowledge. Thus, we do not discuss these approaches in detail in this literature review.

A more promising set of approaches to NLP for low-resource languages include **cross-lingual/multilingual approaches**. These approaches utilize the annotated datasets available in a handful of high-resource languages (such as English, Chinese, Arabic etc.) to build NLP models for the low-resource languages for which such training datasets are not adequately available. These approaches are essentially transfer-learning approaches as they aim to transfer linguistic knowledge from the high-resource languages called *Source Languages* to the low resource languages called *Target Languages.* Transferring such knowledge is challenging as the *Source* and *Target* languages can differ significantly in the lexical, word orders, syntactic and semantic properties Ponti et al. (2018a). All these cross-lingual approaches can be classified into two broad categories namely ***Data-transfer Approaches*** and ***Model-transfer Approaches*** described in sections 2.1.2 and 2.1.1. Figure 2.1 depicts the illustrations of these approaches.

2.1.1 Model-transfer approaches

Although the *Data-transfer approaches* described in section 2.1.2 show strong performance on a selected set of low-resource languages, the utility of these approaches are limited by the requirements of resources Agić et al. (2015) such as parallel raw text corpora (for Alignment Projection approaches), translation system or bilingual lexicon (for Machine Translation approaches). It is impractical to assume the availability of such resources for very low-resource languages.

Such limitations are effectively addressed by the *Model-transfer approaches.* All

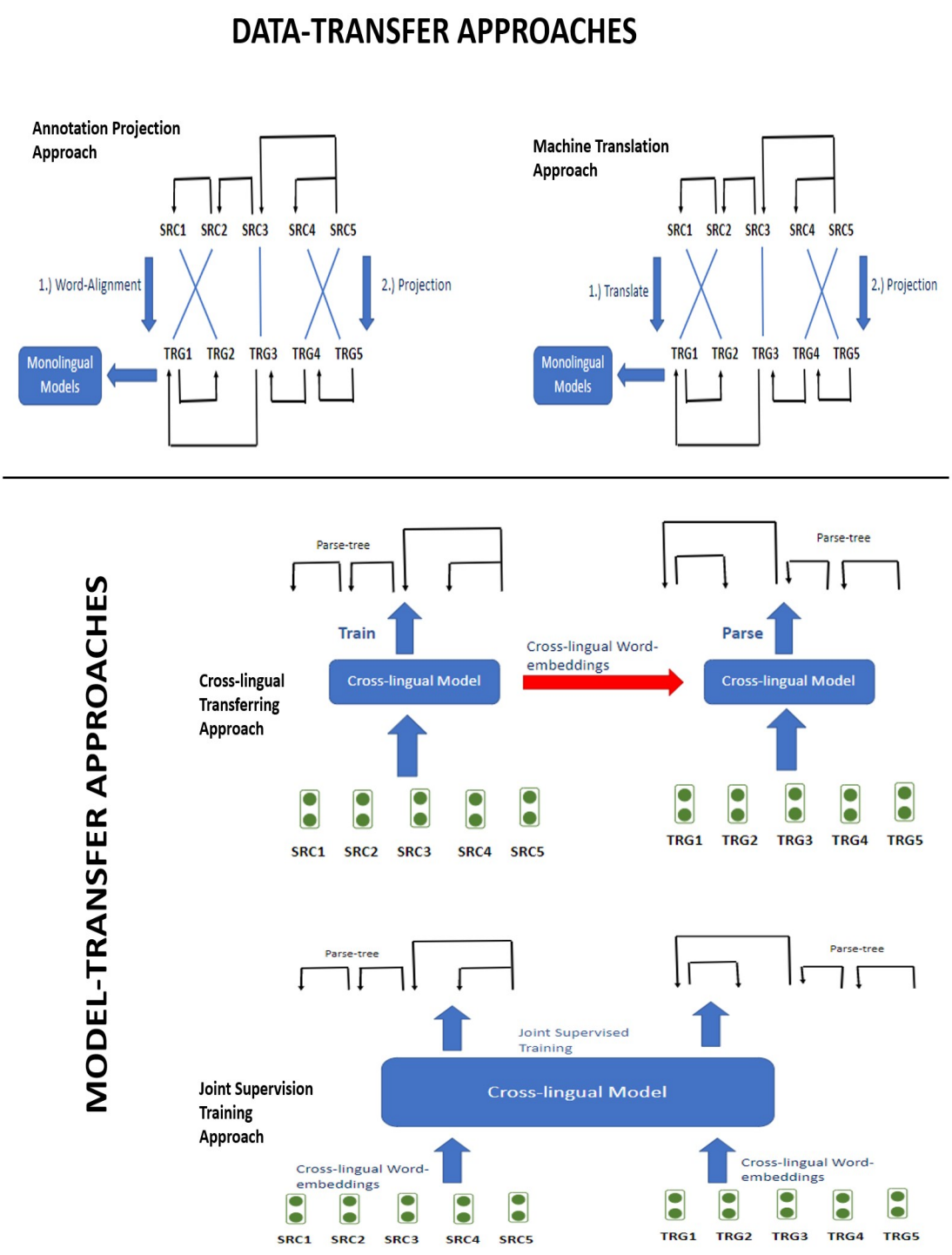


Figure 2.1: Illustrations of various approaches to low-resource NLP

the Model-transfer approaches can be classified into two categories namely ***Cross-lingual Transfer Learning*** approaches and ***Multilingual Joint Supervised Learning*** approaches described as sections 2.1.1.1 and 2.1.1.2 respectively. Due to the incompatibility in the source and target language vocabularies, the state-of-the-art Model-transfer approaches require cross-lingual/multilingual text-representations to make the cross-lingual transferring possible. Two broad categories of such multilingual text-representation include ***Cross-lingual Word-Embeddings*** described in section 2.2 and ***Transformer-based language models*** described in section 2.1.1.4.

2.1.1.1 Cross-lingual Transfer Learning approaches

Cross-lingual Transfer Learning (CLT) based approaches typically involve training a model on a high-resource source-language and applying it on a low-resource target-language Zeman and Resnik (2008). CLT approaches can be applied in two scenarios namely *Zero-shot Learning* Xian et al. (2017) where no annotated dataset is available in the target-language, and *Few-shot Learning* Wang and Yao (2019) scenarios where sparse annotated datasets are available in the target-language. In the Few-shot scenario the model is either pre-trained on the source-language and fine-tuned to the target-language (Lin et al. (2021), Zhao et al. (2021)) or jointly trained on the source and target language (sec 2.1.1.2).

Early CLT based approaches such as (Nivre et al. (2016), Zhang et al. (2012)) used delexicalized or harmonized features (such as POS-tag sequence) to represent the source and target language sentences. Subsequently, Täckström et al. (2012) augmented these delexicalized word-representations with the multilingual Brown word clusters Ciosici et al. (2019). However, such delexicalized features are also unavailable for most low-resource languages and ignoring lexical information impacts performance significantly. Hence modern approaches instead use various cross-lingual/multilingual word-embeddings which can be learnt from simple raw-text corpora in source and target languages (sec 2.1.1.3).

On the other hand, most state-of-the-art approaches to various NLP are transformer-based approaches Vaswani et al. (2017) that utilize multilingual transformer-based language-models (such as mBERT Devlin et al. (2019)) for text-representation (sec 2.1.1.4). To train these transformer-based models for a specific NLP task, a task-specific layer is added on top of the already available pre-trained transformer language-model (such as mBERT Devlin et al. (2019)) and the weights are fine-tuned on the

source-language training dataset and then can subsequently be applied to the target-language. We refer to Rothman (2021) for more detailed description of this fine-tuning process.

2.1.1.2 Multilingual Joint Supervised Learning

It simply involves training NLP models on a joint multilingual mixed training corpus. The multilingually trained models usually outperform monolingual models as these can leverage more (but noisier) data Ammar et al. (2016). Furthermore, in a similar way, as the proficiency of a speaker's previous languages can enhance his/her ability to learn a new language Abu-Rabia and Sanitsky (2010), a model which is trained on multilingual dataset can learn to generalize (and thereby perform well) over unknown or lesser-known languages. Hence even in zero-shot cross-lingual scenarios Xian et al. (2017), it is observed that the models trained on a joint multilingual corpus of source-languages outperform models trained on a single source language Fang and Cohn (2017). Multilingual models are also observed to be more cost-effective in terms of model-parameters Pappas and Popescu-Belis (2017). Multilingual Joint Training is particularly useful in scenarios where all languages are low-resource Khapra et al. (2011) or in code-switching scenarios Adel et al. (2013).

Multilingual joint learning strategically involves parameter sharing Johnson et al. (2017) across languages. Typically, the architecture of a multilingual neural-network model comprises of language-specific or shared parameters across languages. Shared parameters can include input parameters such as word-embeddings Guo et al. (2016a) and character-embeddings Yang et al. (2016), as well as model-parameters such as shared hidden-layers Duong et al. (2015b) or the shared attention-layers Pappas and Popescu-Belis (2017).

Various approaches also achieved parameter sharing from separate language-specific models by minimizing the distance between the hidden parameters Duong et al. (2015a) or latent representations of parallel sentences (Niehues et al. (2011), Zhou et al. (2015)).

Numerous researchers induct the language-id vector Guo et al. (2016a) along with the word-embedding sequence during the multilingual training. The intuition is that the model would be tailored to the specific target-language. These language-id vectors can be input directly such as one-hot embedding or typology embedding vector Ammar et al. (2016) or could be learnt in an end-to-end nlp task (Tsvetkov et al. (2016), Östling and Tiedemann (2016)) or neural machine translation task (Johnson et al. (2017), Ha et al. (2016)).

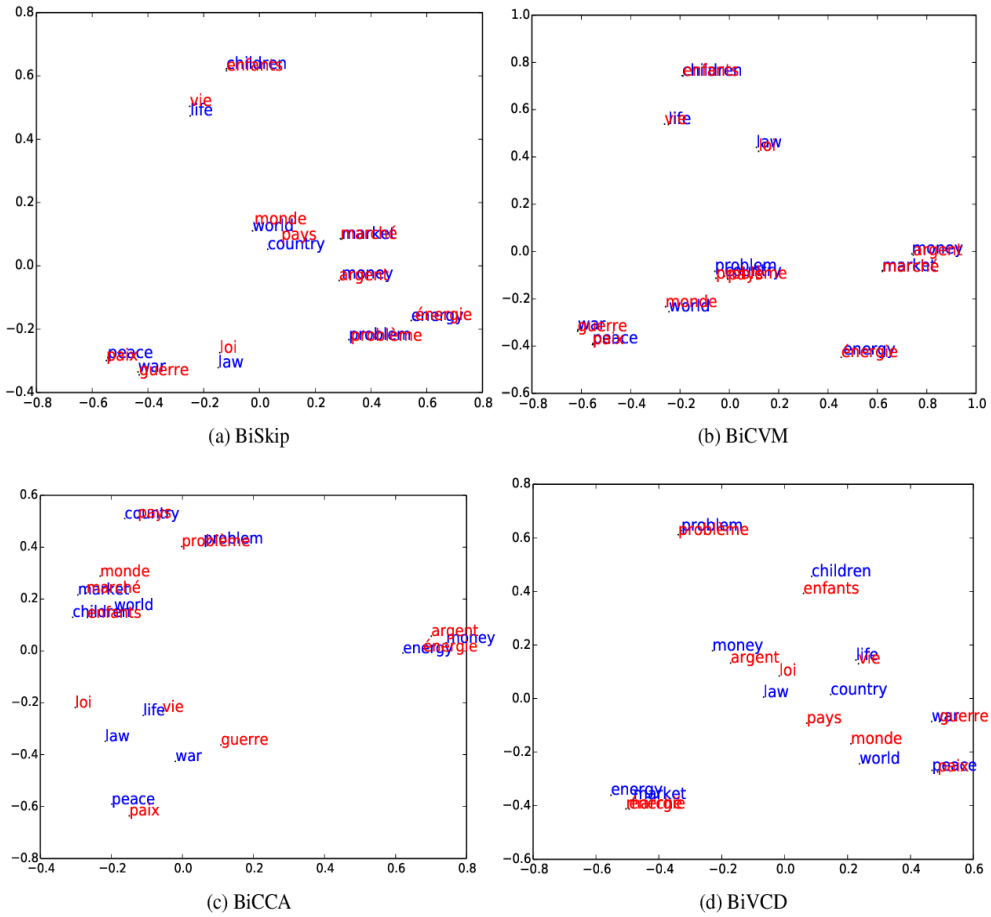


Figure 2.2: PCA projections of popular word-embeddings of some frequent English words and their French translations

2.1.1.3 Multilingual Word-Embeddings

Due to the distinctions between vocabulary, syntactic and semantic properties of the source and target languages, the CLT based models require multilingual text-representations to make cross-lingual model-transferring feasible. As explained in sec 2.1.1.1, the earliest cross-lingual model-transfer utilized only the delexicalized features such as POS-tags to perform cross-lingual transferring. However, the loss of lexical knowledge leads to significant loss in performance Duong et al. (2015a).
Hence, the advanced neural-network based approaches instead use contextualized *Multilingual Word-embeddings*, which can be learnt by training a language-model on mixed polyglot corpora. It is observed that these multilingual word-embeddings multilingual word-embeddings encode both lexical and semantic properties of the words. This can be observed in figure 2.2 which depicts the PCA Abdi and Williams (2010) projections of popular word-embeddings (trained on a common English-French cor-

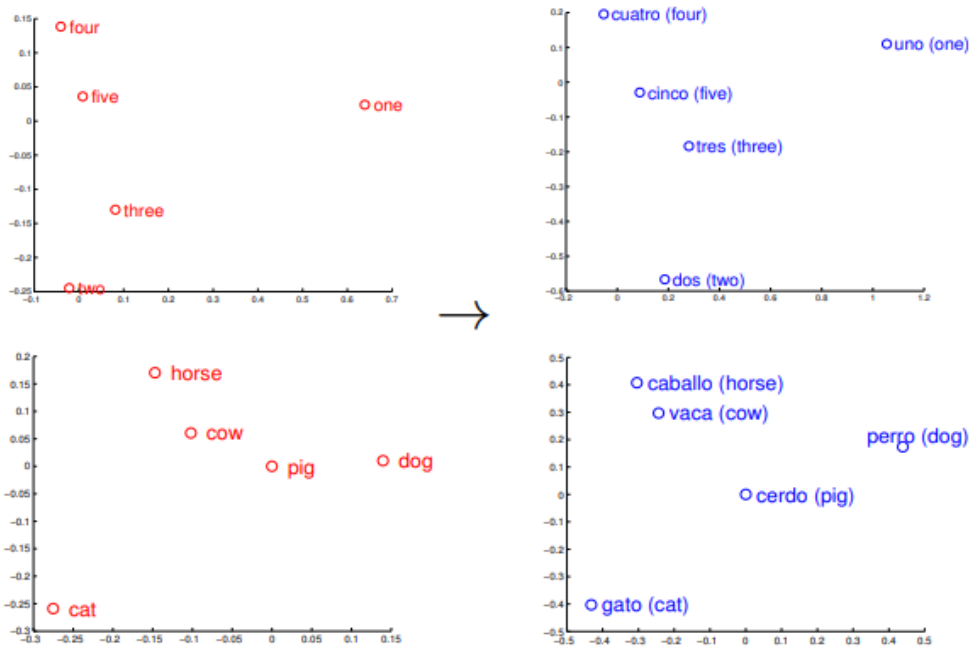


Figure 2.3: PCA projections of monolingual skip-gram word2vec embeddings of English and Spanish words, as computed by Mikolov et al. (2013a)

pus) of some frequent English words and their French translations, projected in the similar vector-space. It is evident in the figure that semantically similar words acquire similar embedding (regardless of the language).

In this section, we describe various multilingual word-embeddings that are commonly utilised for *Cross-lingual Model Transfer-learning*. We used the same classification as proposed by Ruder et al. (2019b), which is based on methods to generate these multilingual embeddings.

Monolingual mapping is the earliest and still most popular technique to learn the multilingual word-embeddings for the CLT based approaches. The technique simply involves learning independent monolingual embeddings in the source and target languages, and subsequently utilizing a feed-forward based *Linear Autoencoder* Kornblith et al. (2019) model to project the target-language embeddings into the source-language space. The *Linear Autoencoder* is trained either on a bilingual lexicon Mikolov et al. (2013a) or in an unsupervised adversarial manner Conneau et al. (2018a). Alternatively, instead of projecting Target language embeddings into the Source language embedding-space, both can be projected on a new, lower-dimensional space through canonical correlation analysis (CCA) (Ammar et al. (2016), Guo et al. (2015)). Figure 2.3 depicts the PCA projections of monolingual skip-gram word2vec

embeddings of English and Spanish words, as computed by Mikolov et al. (2013a). As evident in the figure that the words follow similar distribution pattern in both English and Spanish. Subsequently authors of Mikolov et al. (2013a) performed Linear Transformation to project both embedding-spaces into a common vector-space. These transformations can then act as cross-lingual embeddings to be used to perform cross-lingual transfer from English to Spanish.

Pseudo cross-lingual learning is a unique technique which involves replacing selected words in a raw-text source language corpus with their respective target language translations and vice-a-versa, thereby building a large mixed corpus. Subsequently both source and target language word-embeddings are trained on this pseudo code-mixed corpus. Word-substitutions are performed using a bilingual lexicon Xiao and Guo (2014), through machine-translation (Gouws and Søgaard (2015); Duong et al. (2016)), or simply by randomly shuffling words between aligned corpora (if available) in the two languages Vulic and Moens (2015).

Cross-lingual Fine-tuning approaches are very similar to the Monolingual mapping approaches. These approaches train independent monolingual embeddings for the source and target languages and subsequently fine-tune these monolingual embeddings (to bring them into similar space) by optimizing on various sentence-level cross-lingual constraints. These include tasks such as minimizing the distance between hidden representations of similar sentence Hermann and Blunsom (2013), similar sentence decoding Lauly et al. (2014), minimizing correlation loss between similar texts Chandar AP et al. (2014) etc.

Finally, **Joint optimization** approaches are similar to Cross-lingual Fine-tuning with the only difference being that in Joint optimization approaches the monolingual word-embedding training and cross-lingual constrains are optimized simultaneously for both languages by minimizing a combined loss-function. The constrain tasks include alignment-based translations Klementiev et al. (2012), cross-lingual word contexts Luong et al. (2015), minimizing the distance between similar sentence representations Gouws et al. (2015), image description Rotman et al. (2018) etc.

2.1.1.4 Transformer Based Language Modeling

Vaswani et al. (2017) proposed ***Transformer architecture*** which is a unique neural-network architecture to process the sequential data without utilizing any recurrence. Before the introduction of Transformers, various sequential deep-learning architectures such as RNN Medsker and Jain (2001), LSTM Hochreiter and Schmidhuber (1997)), GRU Chung et al. (2014) etc. were widely used to process temporal

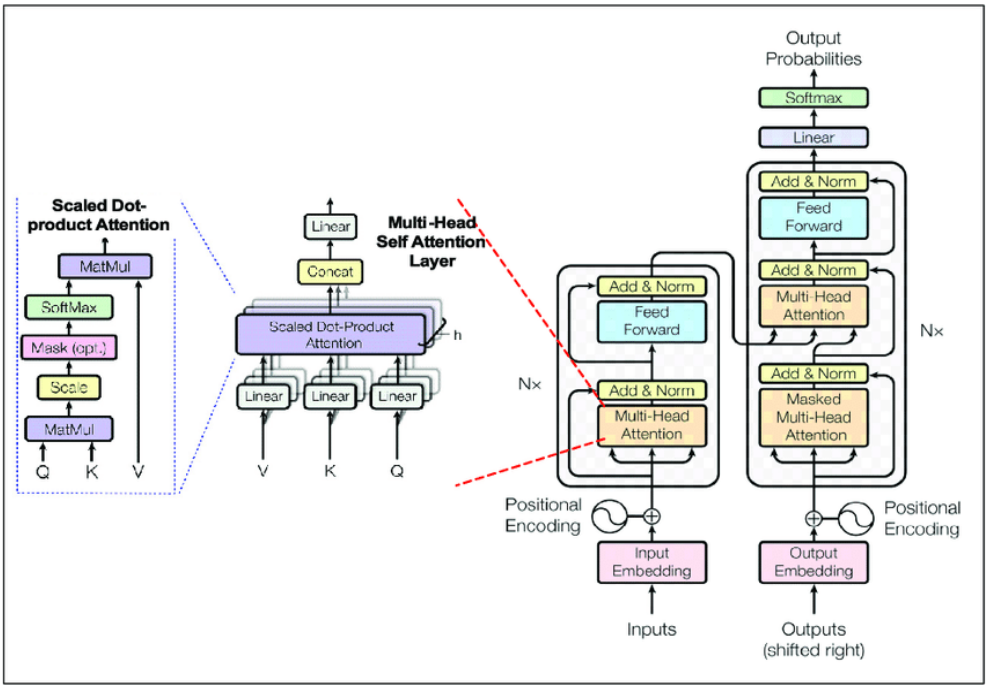


Figure 2.4: Transformer architecture. Figures from Vaswani et al. (2017)

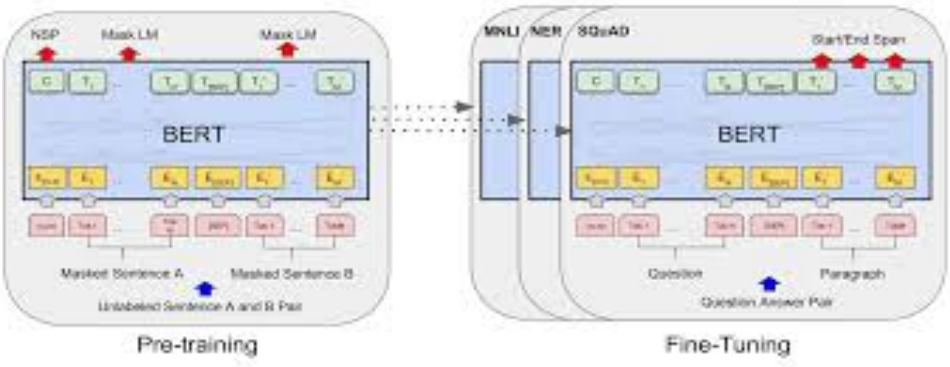


Figure 2.5: BERT Architecture. Figure from Devlin et al. (2019)

data (such as sequence of words in a sentence).
Given a sequence of words $w_1, w_2, w_3....w_T$, for any word w_t at time-step t within this sequence, to compute its hidden-representation r_t, the RNN based architectures (such simple-RNN, LSTM, GRU etc.) require the value of the hidden-state r_{t-1} of the previous word w_{t-1} within the sequence. Hence these architectures by design, allow the computation at one word in the sequence at a time. This sequential nature of these architectures does not leave any scope of parallelization (unlike in CNNs Shin et al. (2016) where various filters can be applied in parallel). The Transformer architecture addresses this issue effectively.
Transformer architectures instead aim to capture the contextual information about the word in a sentence through a self-attention layer. The architecture appends the position-encoding representing the position of each word within the sentence, to their word-embeddings. Subsequently this word-embedding (with positional-encoding) sequence is fed into the self-attention layers to encode context of each word.Transformers compute the contextual hidden representation of each word in the input sentence simultaneously in parallel. Figure 2.4 depicts the architecture of Transformer model.
Once the Transformer architecture was proposed, researchers subsequently attempted to train unsupervised language models based on it. BERT Devlin et al. (2019) is the first such transformer based language model that is trained on raw text-corpus. The parameters of BERT model were trained by optimising on the cloze task Taylor (1953) and next sentence identification task. In the Cloze task, selected words in the raw-text training corpus are masked and the model is trained to simply predict these masked words from surrounding words. Whereas in next sentence identification task, a pair of sentence is classified as adjacent and non-adjacent sentences.
The pre-trained BERT language model is subsequently made available open-source. Users can utilise this language model for numerous downstream tasks by adding subsequent layers to it. The model parameters can also be fine tuned for specific tasks based on available training data. Figure 2.5 depicts the architecture and usage of BERT. Authors have also published a Multilingual variant of BERT (called mBERT) which is trained on a mixed polyglot corpus including over 80 languages.
Inspired by BERT, numerous other Transformer based Language models were trained and are made available online. Notable examples include BART Lewis et al. (2019), XLM Conneau and Lample (2019), XLM-R Conneau et al. (2019), GPT-2 Radford et al. (2019), ALBERT Lan et al. (2019) etc. These models significantly vary in training mechanisms but are mostly similar in architecture design and usage. Furthermore, Transformer based models are developed for tasks other than NLP such

as Parmar et al. (2018) for Image data, Wave2Vec Schneider et al. (2019) for Speech processing etc.

2.1.2 Data-transfer approaches

These approaches aim to create training datasets in the low-resource *Target languages* from the available datasets in the high-resource *Source languages*. Subsequently, the NLP models for the target-languages are trained on these autonomously created datasets. There are two key categories of data-transfer approaches namely **Annotation projection approaches** and **Machine Translation approaches** described as sections 2.1.2.1 and 2.1.2.2.

2.1.2.1 Annotation-projection approaches

The *Annotation-projection* approaches were first introduced by Yarowsky and Ngai (2001) and Hwa et al. (2005) (almost simultaneously). These proposed approaches simply involved performing word-alignments on a pair of parallel raw-text corpora in the source and target languages using a translation lexicon. After such word-alignments, the authors performed the syntactic parsing of the source-language raw-text corpus in the pair (using pre-trained source-language parser). Subsequently, the predicted annotations (e.g. PoS-tags, syntactic trees) are directly projected to the paired target-language corpus and used to train a supervised model in the target language. Later refinements to these approaches are referred to as *Soft annotation projections* Das and Petrov (2011), Padó and Lapata (2009). These approaches use numerous constraints derived from known linguistic properties of the target language to complement the word-alignment. Furthermore, some approaches project label properties (Wang and Manning (2014), Agić et al. (2014)) or the sets of most likely labels (Khapra et al. (2011); Wisniewski et al. (2014)) instead of a single label for each word.

2.1.2.2 Machine translation approach

The Machine translation approaches Durrett et al. (2012) can be applied when the parallel raw-text corpora are not available in source and target languages. In these approaches, each sentence within the source-language training corpora is machine-translated into the target language using a translation-model (if available) or a bilingual lexicon Banea et al. (2008). Subsequently, the annotations are projected from source to target language.

There are numerous approaches that utilised Machine Translation to build and evaluate Cross-lingual Transfer Learning approaches to various NLP tasks and for various target languages. There are three main types of approaches to train and test Cross-Lingual transfer learning approaches. All these approaches utilise cross-lingual/multilingual text-representations described in sections 2.1.1.3 and 2.1.1.4 for cross-lingual transferring.

The first set of approaches aim to translate a large English language corpus into the respective target language. Subsequently, this translated corpus is used to train a monolingual model, which is then evaluated on an already available test dataset in the target language. These approaches are referred to as TRANSLATE-TRAIN approaches Conneau et al. (2018b).

On the other hand, the Second set of approaches aim translate the available target language test dataset. Subsequently the approaches train a Monolingual model on available English training corpus and evaluate it on this translated test corpus. These approaches are referred to as TRANSLATE-TEST approaches Singh et al. (2019).

However Artetxe et al. (2020) proved that training the model on raw English training corpus and evaluating it on translated test corpus. Instead Artetxe et al. (2020) suggest translating the train corpus to the target and then back translating into English for significant improvement in performance.

Finally, the third set of approaches simply train a model on raw English train corpus and evaluates on raw target language test corpus. These include standard Zero-shot (or Few-shot if few target language examples are included in the training set) described in section 2.1.1.1.

2.2 Linguistic Typology and Databases

2.2.1 Linguistic Typology

Linguistic Typology is the branch of linguistics which involves classification of human languages according to their phonological, syntactic and semantic properties (Comrie (1989); Croft (2002)). Linguistic typology can be both synchronic and diachronic. *Synchronic typology* involves the study of structural similarities and differences between languages which are contemporary to each other (in any time-period), while ***Diachronic typology*** involves the study of historical evolution of a language. In our work, we utilized the ***Synchronic typology*** knowledge with cross-lingual transfer learning.

Typologists identify structural and semantic features to represent the properties of

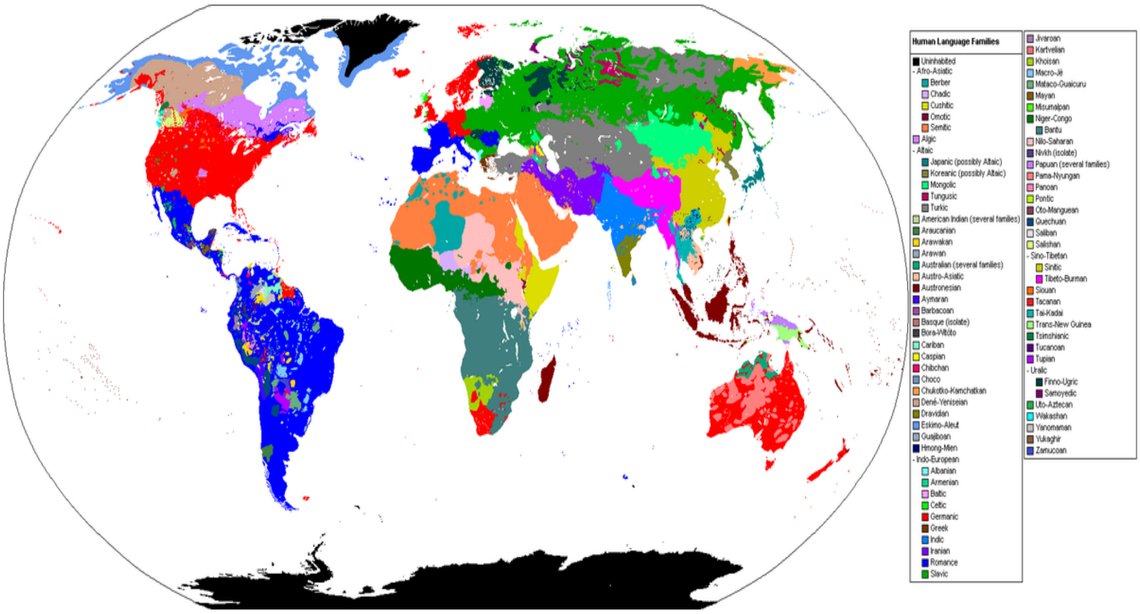


Figure 2.6: The geographical distribution of various linguistic-families in the world. Figure from Pereltsvaig (2020)

any language. These typology features can include syntactic Dixon and Dixon (1994), word-order Greenberg (1969), lexical Walker (2014), phonological, morpho and semantic features (d'Andrade (1995), Dixon (1977), Berlin and Kay (1991), Bowerman et al. (2001), Talmy (1991)). Each such feature has a set of values. Each of the world's languages possesses one of the possible values as the dominant value for the respective feature. For very-low resource language, the value of each feature-value is extracted from the empirical typology documentation Bickel (2007). Such typological documentation involves collecting texts or speech excerpts and assessing the features of a language based on their analysis.

Languages can be classified on numerous criteria such as number of speakers, time-period, age-group of speakers etc. However, the popular classification criteria among typology researchers are the ***Genealogical classification*** and ***Geographical classification*** of languages. *Genealogical classification* involves grouping the world's languages into families according to their degree of diachronic (historical) relatedness Wichmann (2017). In other words, the languages which originate from the same historical language are grouped together as one linguistic family/sub-family. Figure 2.6 depicts the geographical distribution of various Genealogical linguistic-families in the world. On the hand, *Geographical classification* involves grouping based on the place of origin in the world Lyovin et al. (1997).

Although typology classification and Genealogical/Geographical classification are dis-

tinct, the cross-lingual analysis reveals that the distribution of the typology values across world's languages is far from random. Instead, a strong correlation is observed between the Typology classification and Genealogical/Geographical classification of world's languages. In other words, Languages belonging to the same class/geographical area possess similar properties Pereltsvaig (2020).

2.2.2 Linguistic Universals

Linguistic Typology research also includes identification of *Language Universals*. A language universal is typology pattern observed within a distinct group of languages. As there is a strong correlation between Genealogical/Geographical classification and Typology, such universals are indirectly observed around various linguistic classes.
Linguists identify two distinct kinds of Language Universals namely ***Absolute*** and ***Implicational*** universals. The *Absolute Universals* are typology rules that apply to most of the natural languages (with few exceptions though). For example, ***A language always has Nouns and Verbs***, ***Any spoken language has vowels*** etc. On the other hand, Implicational universals are inter-dependency rules between various typology features. For example, ***Languages with Subject-Object-Verb order have post-position spatial or temporal qualifier***.
The Implicational Universals can be both *Unidirectional* and *Bidirectional*. In a Bidirectional universal the two typology features imply the existence of each other. For example, the languages with the value of Typology-feature 'post-positions' is True, have the Subject-Object-Verb order feature-value as SOV, and likewise the languages with the Subject-Object-Verb feature-value as SOV also have the value of Typology-feature 'post-positions' is True. Hence the implication works both ways. On the other hand, in a Unidirectional universal, the implication works only one-ways. For example, the Languages with value of typology-feature relative-clause before noun is True, also have Subject-Object-Verb feature-value as SOV. However, the reverse is not true. Hence the universal is Unidirectional.

2.2.2.1 Principle and Parameter Framework

The research-work related to identification of universals is inspired by Noam Chomsky's work of Principle and Parameter Framework Joseph Aoun Yen-Hui and Keyser (1991) of linguistic knowledge acquisition. The Principle and Parameter Framework (P and P) states that all human languages, while being superficially as diverse as they are, share some fundamental similarities. Thus, he argues that deep down the specific

Name	**Citation**	**Types of typology-features included**	**Number of Languages**	**Number of Attributes**
World Atlas of Language Structures (WALS)	Dryer and Haspelmath (2013)	Phonology, Morphology, Word-order, Syntax, Semantics, Lexical	2871	192
Atlas of Pidgin and Creole Language Structures (APiCS)	Michaelis et al. (2013)	Phonology, Morphology	76	335
URIEL Compendium	Littel et al. (2016)	Phonology, Morphology, Word-order, Syntax, Semantics, Lexical, Georpahical, Language-id	8070	284
Syntactic Structures of the World's Languages (SSWL)	Collins and Kayne (2009)	Morphosyntax	262	148
AUTOTYP	Bickel et al. (2017)	Morphosyntax	825	1000
Valency Patterns Leipzig (ValPaL)	Hartmann et al. (2013)	Predicate–argument structures	36	80 (1156 values)
Lyon–Albuquerque Phonological Systems Database (LAPSyD)	Maddieson et al. (2013)	Phonology	422	70
PHOIBLE Online	Moran et al. (2014)	Phonology	2155	2,160
StressTyp2	Goedemans et al. (2014)	Phonology	699	927
World Loanword Database (WOLD)	Haspelmath and Tadmor (2009)	Lexical Semantics	41	24 (2000 values)
Intercontinental Dictionary Series (IDS)	Key and Comrie (2015)	Lexical Semantics	329	1310
Automated Similarity Judgment Program (ASJP)	Wichmann et al. (2013)	Lexical Semantics	7221	40

Table 2.1: Major publicly available typological databases (listed by Ponti et al. (2019))

grammars of various natural languages, there exists a Universal Grammar Chomsky (1960). Universal Grammar has two key components namely the Principles which are shared by all natural languages and the Parameters which have unique values for each natural language.
The principles are already encoded within the genetics of a new-born child, while during the language-acquisition the child just tunes the parameters of languages being acquired. The Typology-features can intuitively be considered as the Parameters and the universals can intuitively be considered as the Principles.

2.2.3 Typology databases

In this section, we list and describe the popular (mostly open source) typology databases available online and discuss their drawbacks which limit their utility into state-of-the-art cross lingual models (section 2.2.3.1).
These typology databases are created manually by the linguistic community over the years. These databases provide taxonomy of the typological features, their possible-values, as well as the value of these features for each of the languages including very low-resource languages. Table 2.1 lists the major typological databases available online as listed by Ponti et al. (2019).
Some databases listed in table 2.1 such as World Atlas of Language Structures (WALS) Dryer and Haspelmath (2013) and the Atlas of Pidgin and Creole Language Structures (APiCS) Michaelis et al. (2013) are very comprehensive databases that comprise of typology knowledge about a large pool of languages at multiple levels of language descriptions including word-level, morphological, syntactic, and semantic features.
Among all the databases listed in table 2.1, WALS has been the most popular and most widely used by the NLP community (sec 2.4). The database comprises of 142 typological features in total, 1–19 deal with phonology, 20–29 with morphology, 30–57 with nominal categories, 58–64 with nominal syntax, 65–80 with verbal categories, 81–97 and 143–144 with word order, 98–121 with simple clauses, 122–128 with complex sentences, 129–138 with the lexicon, and 139–142 with other properties. The WALS database comprises of both universal features which are shared by all languages as well as language-specific typology features. Figure 2.7 depicts the distribution of an example word-order WALS feature '81A Order of Subject, Object and Verb' across the languages around the world.
Other databases cover typology-features only at a specific level of linguage description. For example, the databases Syntactic Structures of the World's Languages (SSWL)

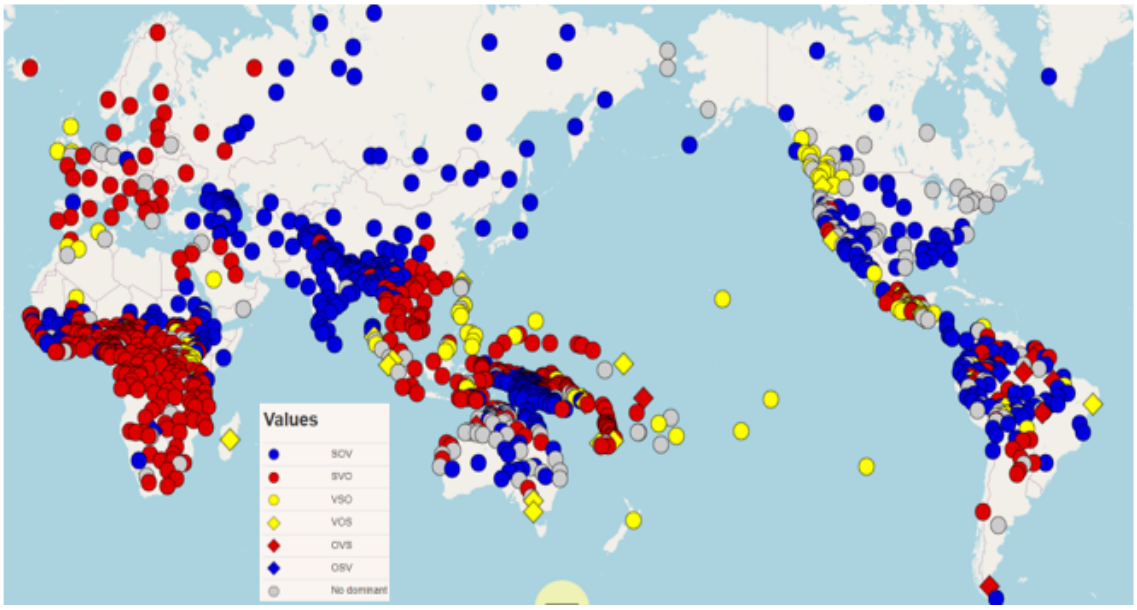


Figure 2.7: Distribution of WALS feature '81A: Order of Subject, Object and Verb' across languages

Collins and Kayne (2009) and AUTOTYP Bickel et al. (2017) provide syntactic typology features. SSWL features are crafted manually crafted, whereas AUTOTYP features are derived autonomously from the data scripts. On the other hand the Valency Patterns Leipzig (ValPaL) Hartmann et al. (2013) is a semantic database that stores knowledge about various verb-forms and their subject-predicate relationship. On the other hand, the Phonetics Information Base and Lexicon (PHOIBLE) Moran et al. (2014) stores information about phonetic and phone-inventory features. The Lyon–Albuquerque Phonological Systems Database (LAPSyD) Maddieson et al. (2013), further provides other articulatory features such as syllabic structures, tonal systems etc. Finally, the StressTyp2 Goedemans et al. (2014) provides stress-related articulation features about various languages.
Table 2.1 also comprises of various lexical databases. For example, the World Loanword Database (WOLD) Haspelmath and Tadmor (2009), Automated Similarity Judgment Program (ASJP) Wichmann et al. (2013) and the Intercontinental Dictionary Series (IDS) Key and Comrie (2015). These databases provide loanword vocabulary and word-pair translations in multiple language pairs.

2.2.3.1 Issues with databases

All these databases suffer from the following major shortcomings (to varying degrees) that limit their utility into modern Cross-lingual NLP models.

1. **Missing Typology:** In most databases listed in table 2.1, the values of numerous typology-features for most of the languages are missing. The issue is more prominent for very low-resource languages which are not well-documented. Sometimes even for a well-documented language, a feature-value can be missing if no dominant value of the specific typology-feature is observed for that language.

2. **Granularity:** Most typology databases listed in Table 2.1 assign only a single value to each typology-feature for a specific-language. This is the most common value observed for the language. However, many exceptions can be observed for each typology-feature in each language. The issue of granularity is more prominently experienced for the semantic and phonological features rather than for the syntactic features, in most languages. Injecting typology-knowledge into a state-of-the-art cross-lingual model without accounting for granularity may indeed lead to a drop in performance rather than a rise.

3. **Redundancy:** Most typology databases comprises of redundant features. For example, WALS database contains a syntactic feature called '81A Order of Subject, Object and Verb' with possible values as SVO, OVS, VOS etc. The database also contains features namely '*82A Order of Subject and Verb*' and '*83A Order of Subject and Verb*'. The issue of redundancy is usually dealt with by manually removing the logically redundant features before injecting the typology knowledge into a neural-network model.

4. **Non-applicability of Features:** Some of the databases listed in Table 2.1 consist of some features that, by definition, apply only to a subset of languages that share some another typology feature-value. For instance, WALS consists of feature '*113A Symmetric or Asymmetric Standard Negation*' with values as Symmetric/Asymmetric. WALS also comprises feature '*114A Subtypes of Asymmetric Standard Negation*' which apply only to languages with feature-value of 113A feature as Asymmetric. Mostly NA value is assigned to languages for which a feature does not apply.

2.3 Prediction of Missing Typology

As discussed in section 2.2.3.1 most of the typology databases listed in Table 2.1 suffer from a major shortcoming of missing feature-values for low-resource languages. This

sparked a new line of research-work which involved utilizing machine-learning/deep-learning techniques to automatically predict such missing feature-values for the low-resource languages. Section 2.3.1 describes various approaches to the autonomous acquisition of missing typology feature-values in details.
Apart from saving time and resources, the autonomous acquisition of missing typology feature-values through machine-learning has several technical advantages over manually crafted those rules. For example, most ML/Dl based approaches learn language representation matrices to represent entire typology knowledge about specific languages. These representations encode additional information which is not included in the manually crafted databases. Furthermore, these approaches can provide/encode the distribution of various feature-values within a single language, rather than just storing a single majority value, thus addressing the issue of granularity. Finally, the ML/DL based approaches allow the continuous representation of languages rather than representing them as discrete cross-lingual typology features.

2.3.1 Approaches to typology prediction

The approaches to autonomous acquisition of typology feature-values can be classified into four categories namely Annotation-based approaches, Unsupervised Clustering approaches, Supervised approaches and Heuristic Distribution approaches described in sections 2.3.1.1, 2.3.1.2, 2.3.1.3 and 2.3.1.4 respectively.

2.3.1.1 Annotation-based approaches

These approaches tend to extract the missing typology feature-values for a language, directly from its available annotated raw-text corpus (either created or available). For example, if a constituency treebank (chapter 3) or a dependency treebank (chapter 4) of a specific language is available, the distribution (as well as dominant values) on most of the syntactic and word-order features for the respective language can then be directly observed within these treebanks (Liu (2010), Bender et al. (2013)). Figure 2.8 depicts the process adopted by Bender et al. (2013) to create a constituency tree for a Welsh sentence (through Annotation projection), and subsequently deriving values of word-order typology-features 'Order of Verb and Subject', 'Order of Determinant and Noun' for Welish.
However, such an annotated corpus is not available in most less-documented languages. Hence researcher have utilized techniques that are similar to the techniques described in section 2.1.2.1 for low-resource NLP to artificially generate such annotated corpus.

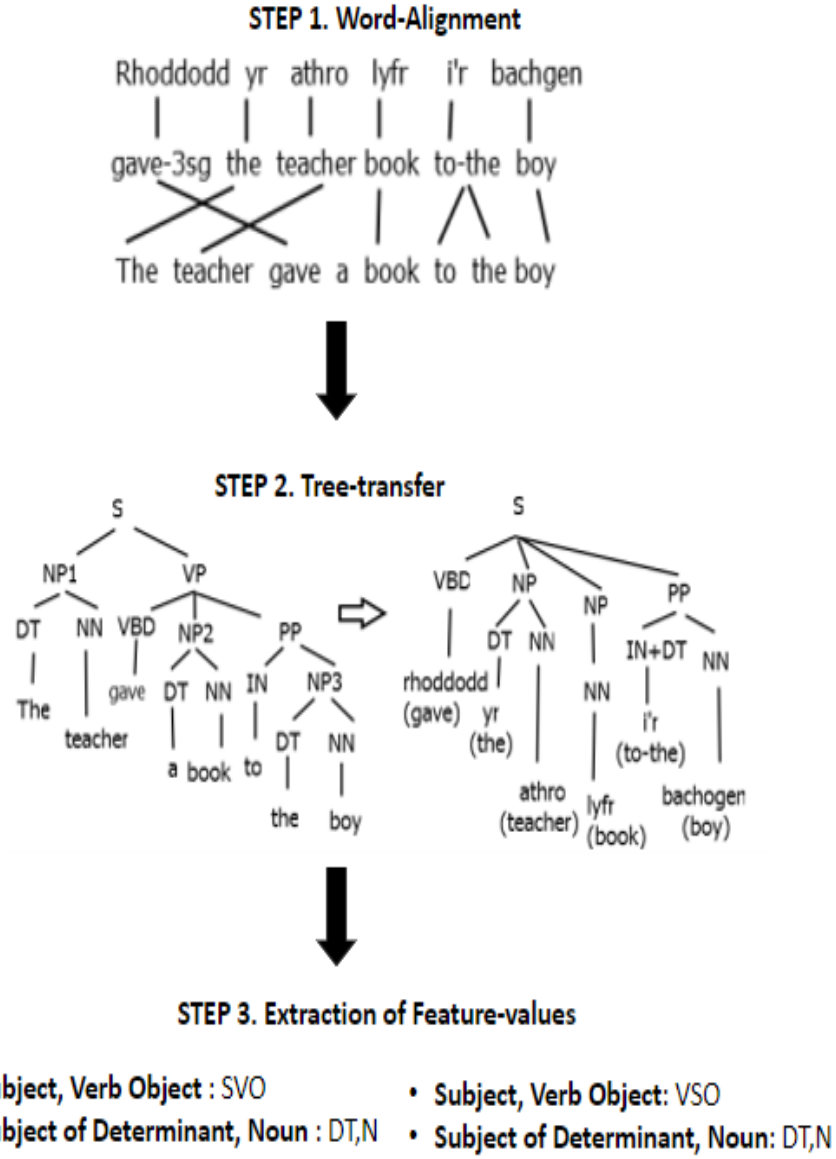


Figure 2.8: The process adopted by Bender et al. (2013) to create a constituency tree for a Welsh sentence (through Annotation projection), and subsequently deriving values of word-order typology-features 'Order of Verb and Subject', 'Order of Determinant and Noun' for Welish.

For instance, Östling (2015) used annotation-projection with multilingual lexicon to synthesize corpora in multiple target languages with the projected morphological and syntactic annotations, from an available high-resource source-language corpus. Subsequently they learnt word-order and lexical typology feature-values for these target-languages from such synthesized corpus. Whereas Gaddy et al. (2016) used cross-lingual model-transfer approach to learn a POS-tagger for various target-languages and subsequently generated POS-tag annotated corpora in those target-languages from the available raw-text corpora.

Once the synthetic annotated corpus is generated, there are multiple ways adopted by researchers to extract the actual values and distribution of various typology features for the specific language. The most common way is to assign a value to a typology-feature is to simply assign the average or the prominent value as observed in the corpus. However Gaddy et al. (2016) used SVM based classifier to classify the value of each typology-feature for each target language.

Finally, some researchers Lewis and Xia (2008), Bender et al. (2013) extracted typological knowledge from the Interlinear Glossed Texts (IGT). These are collections of example sentences and speech-samples that are collated by the linguists for the record. The IGTs mark grammatical and morphological attributes which can be used to derive typology feature values for the specific low-resource language.

2.3.1.2 Unsupervised Clustering approaches

These approaches aim to acquire missing typology feature values for a low-resource language from other well-documented languages (for which these feature-values are known). This is done by clustering the languages according to some shared property, and thereafter every unknown typology feature for any language is simply assigned the majority value within its respective cluster.

Language clustering can be done based on known typology properties (e.g. Teh et al. (2007)) or or based on language genus Coke et al. (2016). Georgi et al. (2010) demonstrated that typology-based clustering outperforms genealogical based clustering on the missing typology prediction task. Various unsupervised algorithms can be adopted for the language clustering such as k-means, k-medoids, the Unweighted Pair Group Method with Arithmetic mean (UPGMA), hierarchical clustering etc.

Some approaches instead performed clustering based on language representation vectors. These language vectors are learnt end-to-end as part of training a neural model for a multilingual downstream NLP task, such as many-to-one Neural Machine Translation Johnson et al. (2017). To learn such language-vectors, a language-id token is

appended at the beginning of each sentence in a polyglot corpus. Subsequently the multilingual neural network is trained on this mixed polyglot corpus end-to-end. The average of the hidden-states corresponding to each appended language-id token is considered the representation vector of the respective language.
These language-representation vectors are used as features while performing the language-clustering TO propagate typological feature-values Bjerva and Augenstein (2018). On the other hand, Malaviya et al. (2017) used these language-representations as features to train A logistic regression model FOR missing typology feature-value prediction. These language-representation vectors can also be used directly within the cross-lingual NLP models to inject linguistic typology knowledge into them, as these representation vectors are expected to encode all the typology knowledge about the respective language.

2.3.1.3 Supervised approaches

Like Unsupervised approaches described in section 2.3.1.2, these approaches also aim to predict the missing typology feature value for a low-resource language from other languages for which the respective feature-value is known. The only difference is that these approaches use supervised machine learning techniques to predict these feature-values. Takamura et al. (2016) and Malaviya et al. (2017) used logistic regression classifier to predict the missing values of WALS features, whereas Wang and Eisner (2017) used deep neural network classifier. Both approaches used other WALS typology features as model predictors.
The supervised approaches can also be guided by non typological predictors. For example, Murawaki (2017) used the genealogical and areal features (along with typology features) to represent each language as a binary latent vector. The approach adopted a Baysian classifier to predict missing feature-value. On the other hand, (Cotterell and Eisner (2017), Cotterell and Eisner (2018)) used various universal cognitive principles such as dispersion and focalization in a model to build phone inventories.
A class of supervised approach simply use the implicational universals Greenberg et al. (1963) with probabilistic models to predict the missing typology feature-value. Using such universals, missing feature-values can be deduced by First-order-logic operations. For instance, based on implicational rule that ***High consonant/vowel ratio + No front-rounded vowels*** $\rightarrow$ ***No tones***, if the former features are known, the latter feature value can be deduced directly. Daumé III and Campbell (2009) proposed a Bayesian model to learn probabilistic implicational universal and thereby predict

missing feature-values. Whereas Lu (2013) proposed a Directed Acyclic Graph based approach to missing typology feature-value prediction from implicational universals.

2.3.1.4 Heuristic Distribution approaches

In these approaches, the typology properties of a low resource language are extracted by analyzing various word-level distributions observed within multilingual parallel text corpora. For example, Wälchli and Cysouw (2012) represented the motion verb distribution within a multilingual corpus as a matrix. Subsequently, the authors performed dimensionality reduction approaches to transform this distribution matrix into a Hamming distance matrix Norouzi et al. (2012). This provides a continuous mapping of lexical verb-semantic properties between various languages.
On the other hand Asgari and Schütze (2017) outlined a distribution based framework to obtain markers of various grammatical features across languages. Finally, missing typology feature-values can be computed by simply observing the word-distributions within monolingual text-documents, through using known linguistic universal facts. For example, Roy et al. (2014) calculated the value of the order of Noun and Adpositions directly from the monolingual text-documents. For any language, Adpositions are the most frequent words, hence can be observed directly. Subsequently, the positions of nouns were established by the authors through various universal linguistic constraints.

2.4 NLP with Typology

As described in chapter 1, the aim of this project is to improve the performance of cross-lingual transfer-learning based NLP models, by inducting the linguistic typology knowledge into them. In this section we review the previously published similar work. In section 2.2.3 we described various publicly available typology databases. In section 2.4.1 we give a high-level overview of the various typology features used previously with the Cross-lingual NLP approaches. In section 2.3.1 we classify and describe various approaches to utilize Typology knowledge with Cross-lingual/Multilingual NLP published previously.

2.4.1 Typology features for NLP

The previous work in Cross-lingual NLP with typology, is primarily limited to the utilization of word order features from WALS Dryer and Haspelmath (2013) aimed at the task of dependency parsing, as word order typology knowledge of a low-resource

Features	Ammar et al. (2016)	Daiber et al. (2016)	Naseem et al. (2012)	Täckström et al. (2013)	Zhang et al. (2012)	Barzilay and Zhang (2015)
89A:Numeral and Noun		✓	✓			
88A:Demonstrative and Noun		✓	✓		✓	
87A:Adjective and Noun	✓	✓	✓	✓	✓	✓
86A:Genitive and Noun	✓	✓	✓	✓	✓	✓
85A:Adposition and Noun Phrase	✓	✓	✓	✓	✓	✓
84A:Object, Oblique and Verb		✓				
83A:Object and Verb	✓	✓				✓
82A:Subject and Verb	✓	✓				✓
81A:Subject, Object and Verb		✓	✓	✓	✓	

Table 2.2: The WALS word-order features utilized by various popular modern approaches cross-lingual dependency parsers with linguistic typology knowledge

target-language can provide crucial guidance to a cross-lingual parser (trained on a different high-resource source language) in predicting the dependency relationships Naseem et al. (2012). Table 2.2 outlines the WALS word-order features utilized by various previously published cross-lingual dependency parsers. As evident in table 2.2, all these CLT based dependency parsing approaches used quite similar word order features, inspired by Naseem et al. (2012) indeed.

On the other hand, Daiber et al. (2016) utilized a more comprehensive subset of WALS typology feature, which included nominal category featues (e.g. 'Conjunctions and Universal Quantifiers') and nominal syntactic features (e.g. 'Possessive Classification') features along the WALS word-order features. Berzak et al. (2016) utilised all the features from WALS database except the lexical and the redundant ones. Whereas Søgaard and Wulff (2012) included all the WALS features except phonological features. Tsvetkov et al. (2016) used binarized phonological features from URIEL Littel et al. (2016).

Finally, some previous approaches Agić (2017) and Ammar et al. (2016) considered

the full set of available typology features without adopting any pre-selection. L. On the other hand, Schone and Jurafsky (2001) did not basic typological features, but instead numerous derived implicational universals Plank and Filimonova (2000).

2.4.2 Approaches to Cross-lingual NLP with Typology knowledge

This section describes the previously published cross-lingual approaches to various NLP tasks which utilize typology knowledge. These approaches utilized typology knowledge either to perform feature engineering, source-target language mapping or facilitating cross-lingual transfer of model parameters. All previous approaches to cross-lingual NLP can be classified into four categories namely Selective source sharing, Target language biasing, Data selection with Typology and Rule-based approaches described as sections 2.4.2.1, 2.4.2.2, 2.4.2.3 and 2.4.2.4.

2.4.2.1 Selective source sharing

This approach was first introduced by Naseem et al. (2012) in a generative model for cross-lingual dependency parsing. The generative parser proposed by Naseem et al. (2012) is trained on a joint polyglot corpus of high resource source languages. The authors assume that the head-modifier relationships in any language are always derived from a set of universal rules which are shared by all languages, whereas the order of head and modifier in a sentence are based on language-specific properties. For example, in all the languages, a noun is always modified by an adjective. However, in some languages (such as English) the adjective precedes the noun while in other languages (such as Nihali) the noun precedes the adjective.
Based on this intuition, the proposed approach aimed to learn the dependency relations from all source languages, while the ordering in these relations (direction of head-dependency) only from the typologically similar source languages, within mixed polyglot training corpus. Hence the approach builds two distinct models for the Head-Modifier relationship prediction and Direction-prediction. The probability of the direction (left or right) of a head-modifier relationship is computed by applying following equation 1

$$P(d|m,h,l) = \sigma(wg(m,h,l,f_l)) \tag{2.1}$$

Here Σ denotes SoftMax function and w indicates the trainable weights. Function g() takes four inputs namely head POS-tag as h, Modifier POS-tag as m, Language-id as l and the typology properties of the language l being parsed represented as feature

f_l. Hence the probability of the direction is dependent on the typology of language being parsed.
A discriminative version of the above-described model was proposed by Täckström et al. (2013). Unlike by Naseem et al. (2012) model which considers all typology features during the prediction of any relationship direction, the discriminative model only utilized relevant features for each head-modifier relation while predicting its direction. For example, if the WALS typology feature 'Order of subject, verb, and object' is relevant only if the head is verb and modifier is the Noun. Hence the proposed model is a delexicalized first-order graph-based parser based on a carefully selected feature set. From the set proposed model considered only the universal typology features related to selection preferences and dependency length suggested by McDonald et al. (2005) as well as manually crafted language-specific features, while representing the typological properties of the target language.
This approach was further extended by Barzilay and Zhang (2015) which is a tensor-based models that avoids the need of manual crafting and manual feature-selection. It represents the entire typology knowledge as compact tensor representation and aims to train the model to automatically select relevant features, rather than manually selecting them.

2.4.2.2 Target language Biasing

These are model-transfer approaches that utilize linguistic typology knowledge to tune the parameters of the shared model towards the target-language on which NLP is being performed. These approaches involve training a model usually on a mixed polyglot corpus, with the typology knowledge about the language of each training-batch being inputted along with text-representation vectors. This improves cross-lingual transferring ability of the model, specifically in case when the target languages are very distinct from all source languages on which the cross-lingual model is trained.
Daiber et al. (2016) built probabilistic word-alignment pairs in a multilingual for parallel corpus. Subsequently the authors trained a machine translation model on this aligned multilingual corpus and injected these alignment probabilities as input.
On the other hand, Ammar et al. (2016) utilized WALS typology knowledge to improve the performance of a cross-lingual Transition based dependency parser. This Transition-based parser comprises of a Stack s, Buffer b and a set of all possible actions A, this transition-based parser selects the best action $a \in A$ to be taken at time-step t (eg: SHIFT ot REDUCE) given the current state of stack s_t, current state of buffer b_t and previous action-sequence a_{t-1} by applying equation 2.2. Here

$P_t \in R^{|A|}$ is the probability vectors comprising of probabilities of all actions $a \in A$ to be taken at time-step t. The action at time-step t is computed as $a_t = argmax(P_t)$. The process is continued until the buffer is empty and the stack comprises of full dependency tree.

$$P_t = max\{0, W * [s_t : b_t : a_{t-1} : l] + W_{bias}\} \tag{2.2}$$

The authors encode the current states of stack and buffer using stack- LSTM models with cross-lingual word-embedding. The authors also appended Language-id vector along with word-embeddings for target-language biasing. They experimented with numerous language-id such as One-hot vector as well vector comprising of all WALS typology features.
Tsvetkov et al. (2016) injected phonological typology knowledge about various languages while training a phone-level language model. On the other hand, some approaches such as Schone and Jurafsky (2001) used typology knowledge to define the prior model in a Bayesian Network.

2.4.2.3 Data selection with Typology

These approaches aim to utilize linguistic typology knowledge to perform the source language selection or source training example selection (in case of polyglot source training corpus) for the cross-lingual training, based on similarity between source and target languages. The typology-based data-selection is commonly adopted with either the cross-lingual model-transfer approaches (section 2.1.1) to select the most suitable source languages which are comparatively typologically closer to the target language (Deri and Knight et al 2016), or with the joint supervision approaches (section 2.1.1.2) to weigh the contribution of each example within the joint polyglot training corpus. For example, Søgaard and Wulff (2012) and Agić (2017) weighted examples in joint polyglot corpus based on the Hamming distance Hamming (1950) between the target and source language (example's language) typological vector.
Selection of source-languages can also be data-driven fashion, instead of measuring similarity between typological vectors derived from various typology databases. For example, Rosa and Žabokrtskỳ (2015) performed the source-language selection for a cross-lingual delexicalized parser, based on the KL divergence Csiszár (1975) distances between part-of-speech trigram distributions of various source languages and the trigram distributions of target language being parsed. On the other hand, Ponti

et al. (2018a) made the source-language selection based on the Jaccard distance Murphy (1996) between the morphological features and the tree-edit distance Bille (2005) of dependency parses of similar sentences, between source and target-languages.

2.4.2.4 Rule-based approach with Typology

A unique approach to utilize typology knowledge for low-resource parsing was proposed by Bender (2016). The proposed approach built a rule-based grammar from the known typology features. The built grammar is within the Minimal Recursion Semantics framework Copestake et al. (2005) and can be used to directly perform semantic parsing of any natural language.

2.5 Conclusion

As explained in Chapter 1, our research work involves utilising linguistic typology knowledge available in numerous external databases to improve the performances of numerous state-of-the-art cross-lingual neural-network based models for various key NLP tasks. In this chapter, we reviewed the previously published work relevant to the research work described in subsequent chapters of the dissertation.

We divide the entire literature review into four segments. In the first segment we provided a high-level overview of various approaches to NLP for low-resource languages. These include *Data-Transfer Approaches* and *Model-transfer Approaches*. In this section we also provide an overview of various cross-lingual word-representations as well as Transformer based language-models as these play a key role in the cross-lingual transferring. Subsequently, in second segment we list and describe various external linguistic typology databases. We utilised the knowledge available in some of these databases to improve the state-of-the-art cross-lingual Model-transfer approaches to numerous NLP tasks.

In the third segment we describe the issue of missing typology that exists in all the popular typology databases. The issue makes the utilisation of the database with any cross-lingual NLP model difficult. Subsequently, we provided overview of ML based approaches to predict such missing feature-values. In our work, we indeed used the mechanisms in these ML approaches indeed to overcome the missing typology issue while we utilise the typology-knowledge with different cross-lingual models. Finally, in the last segment we discussed previously published approaches to cross-lingual that utilised linguistic typology knowledge.

Hence, in this chapter we provided all the background knowledge necessary for a

reader of this thesis to be aware of, to fully understand our original work related to cross-lingual transfer learning based NLP with linguistic typology knowledge, to be described in subsequent chapters, as our work is indeed built on the work described in this chapter.

Chapter 3

Cross-lingual Constituency Paring with Linguistic Typology Knowledge

This chapter is based on our research work published as following paper:

- **Universal Recurrent Neural Network Grammar.** In Proceedings Of 33RD ANNUAL CONFERENCE ON COMPUTATIONAL LiNGUISTICS AND SPEECH PROCESSING (ROCLING) 2021

There are two key frameworks to represent the syntax of an input sentence namely the *Constituency parse-tree* framework and the *Dependency parse-tree* framework respectively. In our work, we improved the performances of state-of-the-art cross-lingual approaches to both *Constituency Parsing (CP)* and *Dependency Parsing (DP)* using linguistic typology knowledge.

In this chapter we will describe the CP task in detail as well as our proposed cross-lingual approach to CP with linguistic typology knowledge in WALS database. In chapter 5 we will review the DP task and the proposed cross-lingual DP model in details.

Section 3.1 provides the high level over-view of phrase-based/constituency grammar as well as the CP task. Section 3.2 described treebank structure whereas Section 3.3 provides a review of monolingual approaches to CP, including both statistical and neural approaches. Finally section 3.4 provides our proposed model and the subsequent sections will describe the experiments to evaluate our proposed model and the results obtained.

Annotation	**Phrase**
S	The school children will visit the Dublin museum during the first week of September
NP	The school children
VP	will visit the Dublin museum during the first week of September
VP	visit the Dublin museum during the first week of September
NP	the Dublin museum
PP	during the first week of September
NP	the first week of September
PP	of September
NP	the first week

Table 3.1: Constituent phrases in the example sentence *The school children will visit the Dublin museum during the first week of September.*

Non-terminal CFG rules	**Terminal CFG rules**
$S \rightarrow NP\ VP$	$Det \rightarrow that\|this\|the\|a$
$S \rightarrow Aux\ NP\ VP$	$NN \rightarrow university\|weekend$
$S \rightarrow VP$	$VBZ \rightarrow goes$
$NP \rightarrow PRP$	$PRP \rightarrow she$
$NP \rightarrow NN$	$Aux \rightarrow does$
$NP \rightarrow Det\ Nom$	$TO \rightarrow to$
$Nom \rightarrow NN$	
$Nom \rightarrow Nom\ NN$	
$Nom \rightarrow Nom\ PP$	
$VP \rightarrow VBZ$	
$VP \rightarrow VBZ\ NP$	
$VP \rightarrow VBZ\ PP\ PP$	
$VP \rightarrow VBZ\ PP$	
$VP \rightarrow VP\ PP$	
$PP \rightarrow TO\ NP$	
$PP \rightarrow IN\ NP$	
$PP \rightarrow PRP\ NP$	

Table 3.2: Examples of Context Free Grammar (CFG) rules

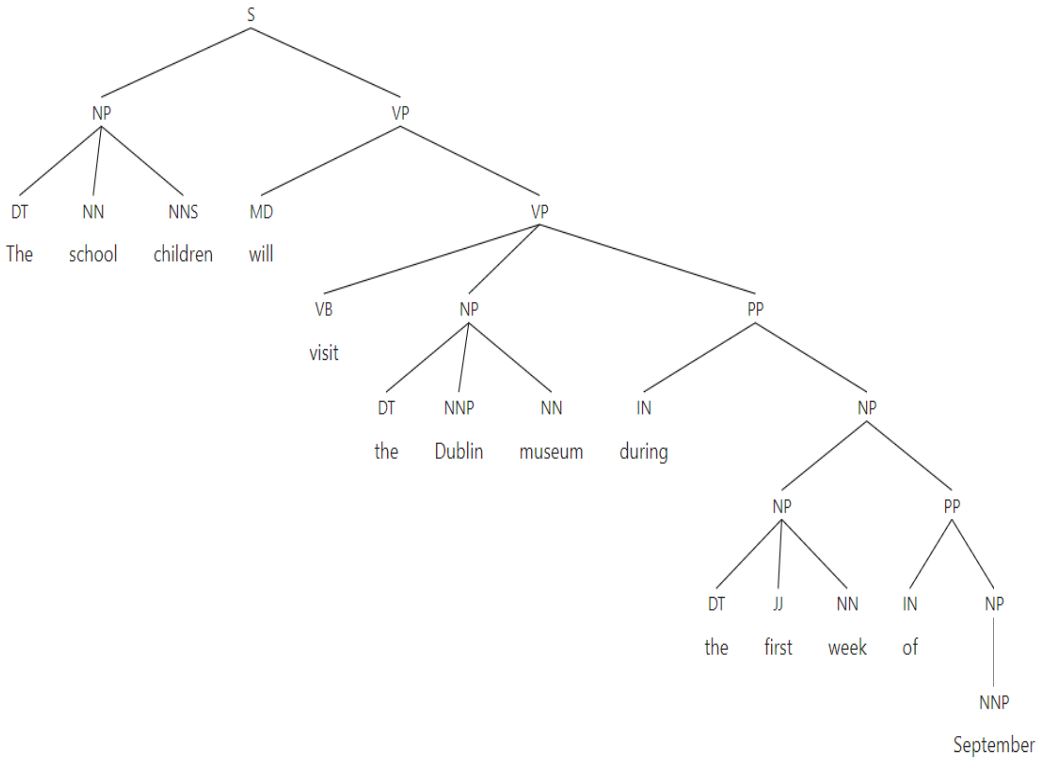


Figure 3.1: Example of a constituency parse-tree

3.1 Phrase-based Grammar

3.1.1 Phrase Constituency

Syntactic constituency grouping aims to group words in a sentence into *phrase constituents*. A word-sequence is identified as a phrase-constituent if all the words in that sequence share some semantic or syntactic property. Each such constituent acts as a single syntactic unit, in the constituency parse tree of the input sentence. For example, Table 3.1 lists all the constituents that can be extracted from the following sentence:

- ***The school children will visit the Dublin museum during the first week of September.***

As evident in Table 3.1, the words *the first week of September* can be grouped together into a single ***Noun-Phrase (NP)*** constituent as they convey a single semantic information i.e. the time of visit. Similarly, the words *the Dublin museum* form another independent ***Noun-Phrase (NP)*** constituent as they collectively describe the place to be visited.

Words can also be grouped into constituents based on the shared syntactic characteristic. For example, the words *The school children* collectively describe the subject of the sentence, hence forming a ***Noun-Phrase (NP)*** constituent.
Furthermore, various constituents in a sentence have a hierarchical structure. For example, as evident in Table 3.1, the words *visit the Dublin museum during the first week of September* forms a single **Verb-Phrase (VP)** constituent of the sentence and is comprised of three smaller constituents namely **Base-verb (VB)**, **Preposition-phrase (PP)** and **Noun-phrase (NP)**.

3.1.2 Context Free Grammar

Context Free Grammar (CFG) or **Phrase-Structure Grammar** is the most widely used framework to extract the entire hierarchy of phrase-constituents from an input sentence. Such a hierarchy of all phrase-constituents in a sentence is represented as its **Constituency parse tree**. Figure 3.2 depicts the constituency parse tree structure of an example sentence *I live in Galway*. The concept of phrase-based grammar date backs to 1900 Wundt (1900) but was formalised by linguist Noam Chomsky in 1956 Chomsky (1956) and Backus in 1959 JW (1959) independently.
A **context-free grammar** typically comprises of a very large set of rules or productions. Each such rule express a possible (allowed) way to group one or more words or phrases together. For example, consider following two rules:

$$NP \rightarrow Det\ Nom$$

$$NP \rightarrow NNP$$

These rules express that a **Noun Phrase (NP)** constituent can be comprised of either a single **Proper Noun (NNP)** constituent or a **Determiner (Det)** constituent followed by a **Nominal (Nom)** constituent. On the other hand, a Nominal (Nom) can be furthur defined by following rules

$$Nom \rightarrow NN$$

$$Nom \rightarrow Nom\ NN$$

These rules express that a **Nominal (Nom)** type constituent can comprise either a **Noun (NN)** or another **Nominal (Nom)** followed by a **Noun (NN)**.
Apart from Constituency-types, the CFG rules can also comprise of lexical units (on

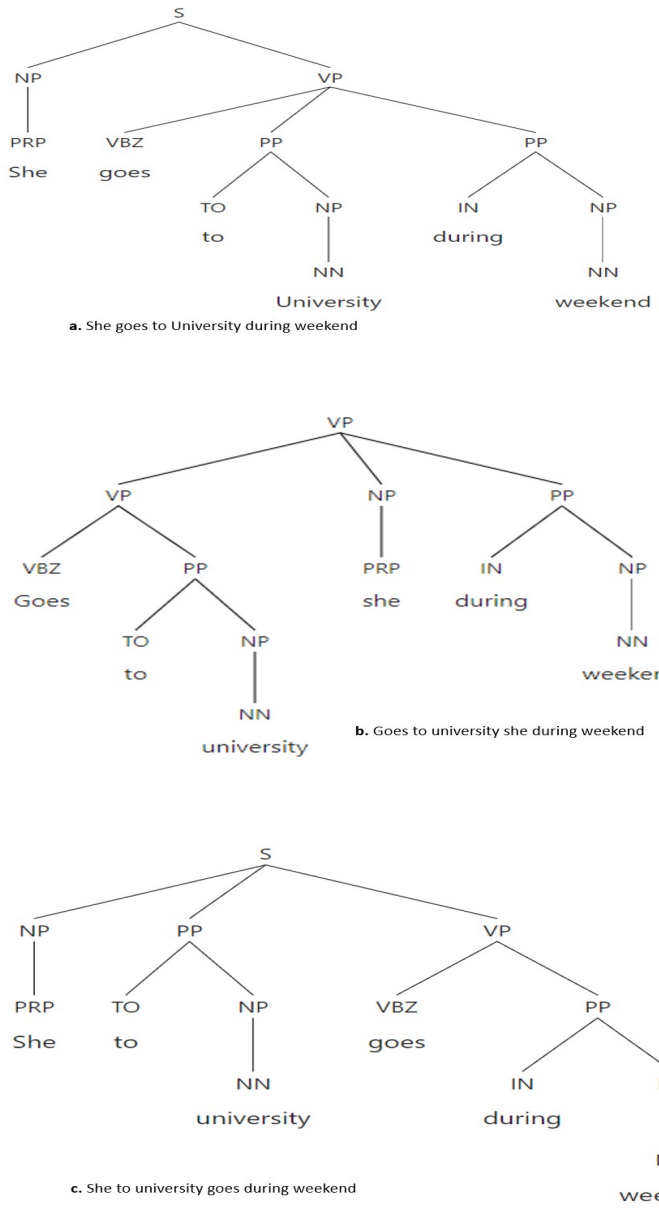


Figure 3.2: Examples of constituency parse trees based on CFG rules.

```
(S (NP The school children)
   (VP will
       (VP visit
           (NP the Dublin museum)
           (PP during
               (NP (NP the first week)
                   (PP of
                       (NP September)))))))
```

Figure 3.3: Representation of constituency parse-tree in Bracket format

the right hand side of the rule). For example, the following two rules comprise of lexical right-hand side.

$$NN \rightarrow School$$

$$Det \rightarrow the$$

$$Det \rightarrow a$$

$$V \rightarrow eat$$

. Each such CFG rule can form branches of a constituency parse tree with left-hand side being the parent node and right-hand side being the children node. The nodes indicating constituency types (such as NNP, Nom etc.) are called **Non-terminal** nodes whereas nodes comprising of lexical units are called **Terminal** nodes.
Hence formally, a context-free grammar G of any language L is defined by three parameters listed as follws:

1. N: Set of non-terminal symbols
2. Σ: Set of terminal nodes (vocabulary of the language)
3. R: Set of rules of the format $A \rightarrow \beta$ where A is always a single non-terminal and β is a sequence of terminal and non-terminal symbols

Furthermore, each CFG consists of a designated **Start (S)** node as a member of its **Non-Terminal** node-set N indicating the constituent-type *Start* This node forms the root node of any complete parse-tree, and usually comprises of all the words in the input-sentence been parsed.
Hence, any input sentence $\hat{S}$ in a language L is considered to be grammatically correct, if at-least one complete *constituency parse-tree* can be constructed for that sentence based on CFG of L. A complete *constituency parse-tree* has following properties:

Terminal Rules	**Non-terminal rules**
$DT \rightarrow The$	$NP \rightarrow DT\ NN\ NNS$
$NN \rightarrow school$	$VP \rightarrow MD\ VP$
$NNS \rightarrow children$	$VP \rightarrow VB\ NP\ PP$
$MD \rightarrow will$	$NP \rightarrow DT\ NNP\ NN$
$VB \rightarrow visit$	$PP \rightarrow IN\ NP$
$NNP \rightarrow Dublin$	$NP \rightarrow NP\ PP$
$NN \rightarrow museum$	$NP \rightarrow DT\ JI\ NN$
$IN \rightarrow during$	$NP \rightarrow NNP$
$JI \rightarrow first$	
$NN \rightarrow week$	
$IN \rightarrow of$	
$NNP \rightarrow September$	

Table 3.3: CFG rules generated from the parse-tree depicted in Figure 3.1

1. Root node is of type ***Start (S)***.

2. All the words in the input sentence $\hat{S}$ form leaf nodes of the tree.

3. All the branches of the tree should be legal (based on valid CFG rules in R_L).

Figure 3.2 depicts an example of a valid and an invalid parse-tree based on the limited set of CFG rules outlined in Table 3.2. In the figure, tree b is incomplete as it does not have single root node S thus making it illegal. On the other hand, tree c is invalid because it consists of an illegal branch formed by the following invalid rule, which is not in the CFG (Table 3.2) of language of the sentence being parsed.

$$S \rightarrow NP\ PP\ VP$$

3.2 Treebanks

In linguistics, a treebank (term coined by Geoffrey Leech Wilson et al. (2003)) refers to a text-corpora with each sentence been paired to its corresponding syntactic or semantic sentence-structure representations (syntactic or semantic parse-tree). These sentences are often annotated manually by the trained linguistics thus making them gold standard. There are numerous open-source treebanks available with various syntactic and semantic annotations. This section describes various constituency parsing treebanks while section 5.2.1 will discuss the available dependency parsing treebanks. Since the treebank is created manually, generally the CFG of the language (if unknown) can be derived from the treebank itself. In such a case, CFG is defined by

Annotation	**Description**
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
FW	Foreign word
IN	Preposition conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
PRP	Personal pronoun
PP	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
TO	to
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund/present participle
VBN	Verb, past participle
VBP	Verb, non-3rd position singular present
WDT	wh-determiner
WP	Possessive wh-pronoun
WRB	wh-adverb
.	Punctuation
’	Comma

Table 3.4: Selected Annotations in the Penn treebank Taylor et al. (2003)

CFG Format	**CNF Format**
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$VP \rightarrow read\|study\|walk\|fly\|book\|...$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow PRP$	$NP \rightarrow I\|she\|me\|you\|we\|...$
$NP \rightarrow NNP$	$NP \rightarrow Galway\|University\|Ireland\|Dublin\|...$
$NP \rightarrow Det\ Nom$	$NP \rightarrow Det\ Nom$
$Nom \rightarrow Noun$	$Nom \rightarrow book\|flight\|meal\|money\|course\|....$
$Nom \rightarrow NomNN$	$Nom \rightarrow Nom\ Noun$
$Nom \rightarrow NomPP$	$Nom \rightarrow Nom\ PP$
$VP \rightarrow VB$	$VP \rightarrow book\|include\|prefer\|...$
$VP \rightarrow VB\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow VB\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow VB\ NP$
$VP \rightarrow VB\ PP$	$VP \rightarrow VB\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow PRP\ NP$	$PP \rightarrow PRP\ NP$

Table 3.5: Example of CFG grammar translated to CNF format. Example from Martin (2021a)

the set of all CFG rules that are observed within these manually annotated treebank. Table 3.3 depicts the CFG rules that are generated from an example parse-tree.
The most popular constituency parsing treebank is the ***Penn Treebank*** Taylor et al. (2003) which was developed by *Linguistic Data Consortium* and *University of Pennsylvania* in the 1990s. The treebank was created by manually annotating sentences from the Brown Francis and Kucera (1979), Switchboard Godfrey et al. (1992), ATIS, and Wall Street Journal Paul and Baker (1992) corpora of English.
The most widely used English CFG is in fact derived from the Penn-treebank corpus. The Penn-treebanks corpus also defined the nested bracket format to representation a constituency parse tree indicated in Figure 3.3. The Penn treebank was further extended for the Arabic Maamouri et al. (2004) and Chinese Xue et al. (2005) languages.
Penn treebank also provided the most widely used constituency types and their annotations. Table 3.4 lists some significant annotations and constiituency-types within the Penn treebank. The annotation set provided by the Penn treebank indeed formed the basis of other treebanks and CFGs in other languages Seki et al. (1991).
Apart from Penn, an other notable treebank is the ***BulTreeBank*** Simov and Osenova (2004). This treebank follows a specific language-theory unlike the **Penn** treebank. This treebank provides the *Head Driven Phrase structure grammar (HPSG)* Pollard and Sag (1994) which is distinct from CFG format.
After the Penn English treebank, numerous linguists developed and publicly released treebanks in other languages as well. Tables 3.8 and 3.9 lists some of these treebanks in various languages. Some of these treebanks were be used by us to train and evaluate our proposed multilingual constituency parser as explained in section 3.8.

3.3 Approaches to Monolingual Constituency Parsing

This section reviews the previously proposed approaches to the Constituency Parsing task. In section 3.3.1 we will review various dynamic programming based approaches to CP task including the ***CKY algorithm*** Manacher (1978), which is the most widely used algorithm to generate constituency parse-tree of an input sentence from the available CFG grammar. In subsequent sections we review modern neural-network based approaches to constituency parsing including *discriminative*, *generative* and *unsupervised* approaches.

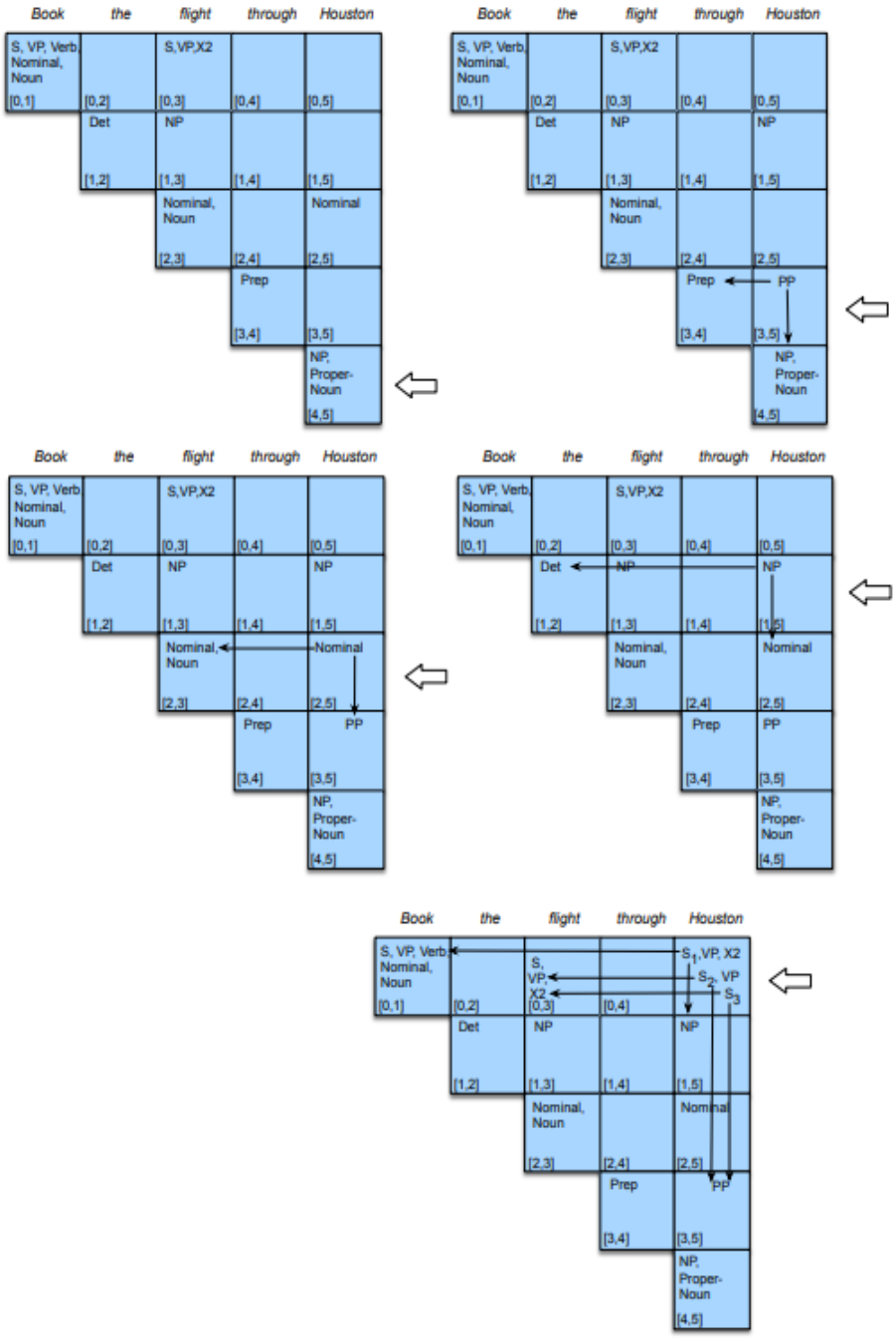


Figure 3.4: Demonastration of slot filling in the CKY algorithm during parsing of an example sentence *Book the flight through Houston*. Figure from Martin (2021a).

Algorithm 1 CKY Algorithm

Require: Input-sentence S as s word-sequence $w = w_1, w_2, ...w_N$; CFG rule-set in CNF format;

```
for i from 1 to N do
    for each r ∈ CFG ∋ r → w_i do
        table[i1, i] ← table[i1, i] ∪ r
    end for
    for j from i − 1 to 0 do
        for k from j + 1 to i − 1 do
            for each r ∈ CFG ∋ r ∼ A → B C ∋ B ∈ table[j, k] and C ∈ table[k, j] do
                table[j, i] ← table[j, i] ∪ r
            end for
        end for
    end for
end for
```

3.3.1 Dynamic Programming approaches

Dynamic programming Bellman (1966) refers to a class of algorithms that can be used to explore a very large search-space efficiently (both time and space efficiency) to find the desired goal-state or the maximum scoring state. A typical dynamic-programming approach resolves a comparatively larger problem by recursively breaking it into smaller problems and resolving them. For the task of parse-tree generation, these sub-problems include the generation of sub-trees.

Cocke-Kasami Younger (CKY) Manacher (1978) algorithm is the most popular *dynamic-programming* based approach to CP task. CKY is a recursive algorithm which starts with the entire word-sequence as a sequence of single-node sub-trees. Subsequently, at each iteration (at each level) the algorithm groups the smaller sub-trees together to generate the sequence of larger sub-trees based on the CFG rules of the language being parsed. The process is continued until only a single parse tree comprising of all initial nodes exists. This tree is outputted as the desired consituency parse-tree of the language being parsed. Section 3.3.1.2 will outline the entire parsing process in details.

CKY algorithm requires the CFG rules to be written in **Chomasky Normal Form (CNF)**. Section 3.3.1.1 will describe the CNF and various approaches to convert standard CFG rules into CNF format. Chart parsing Kaplan (1973); Allén (1982) is a another popular rule-based approach to CP task.

3.3.1.1 Chomsky Normal Form

The *CKY algorithm* requires the CFG grammar to be in Chomsky Normal Form (CNF). As already explained in section 3.1.2, the CFG rule is in format as $A \rightarrow \beta$ where A is a non-terminal and β is a sequence of terminal and non-terminal symbols. On the another hand, in CNF format a grammar rule can be either in the format $A \rightarrow B\ C$ or in the format $A \rightarrow b$ where b is a non-terminal node (word). Hence standard CFG rules are needed to be converted into the CNF format.
There are three kinds of CFG rules which needs to be converted to CNF format are *rules with mixed terminals with non-terminal nodes on right-hand side*, **rules with a single non-terminal on the right-hand side** and **rules in which the length of the right-hand side is greater than two**. These rules are translated into the CNF format by introducing additional Non-terminal nodes and rules.
For example, a CFG rule $NP \rightarrow PP\ ADJ\ NN$ in which the length of the right-hand side is greater than two can be re-written as two new rules namely $NP \rightarrow PP\ X$ and $X \rightarrow ADJ\ NN$. Here, X is a new Non-terminal introduced for translation. Similarly the rule $INF-VP \rightarrow to\ VP$ with mixed terminal and non-terminal nodes on right-hand side can be translated as two rules $INF-VP \rightarrow TO\ VP$ and $TO \rightarrow to$. Here TO is a new non-terminal added to the inventory. Table 3.5 depicts an example CFG grammar been converted to CNF format.

3.3.1.2 CKY Parsing

As our CFG is now in CNF format, each node in our constituency parse-tree will have two daughter nodes (except at final level comprising of POS-tags). Let there be an input sentence S to be parsed with a CFG. The CKY algorithm would parse the sentence S by making a matrix of dimension $R^{|S|*|S|}$, and builds the parse-tree by filling all cells in the upper right triangular portion of the matrix. Here $|S|$ is the length of sentence S. Each cell (i, j) in the matrix is filled with possible non-terminals that can represent the constituents comprising of all the words from index i to j in the input sentence $(i, j \leq n)$.
Algorithm 1 outlines the CKY algorithm. The parsing process starts with filling in the diagonal columns. Any j^{th} column on the diagonal is filled by all non-terminal symbols that satisfy the CFG rule-type $A \rightarrow w_j$, where w_j is the j^{th}. In subsequent steps the algorithm iteratively fills in all the left-over cells of the upper-right triangular portion of the matrix. Any cell (i, j) is filled by applying equation 3.1.

$$Matrix[i, j] = Matrix[i, k] \cup Matrix[k + 1, j] \tag{3.1}$$

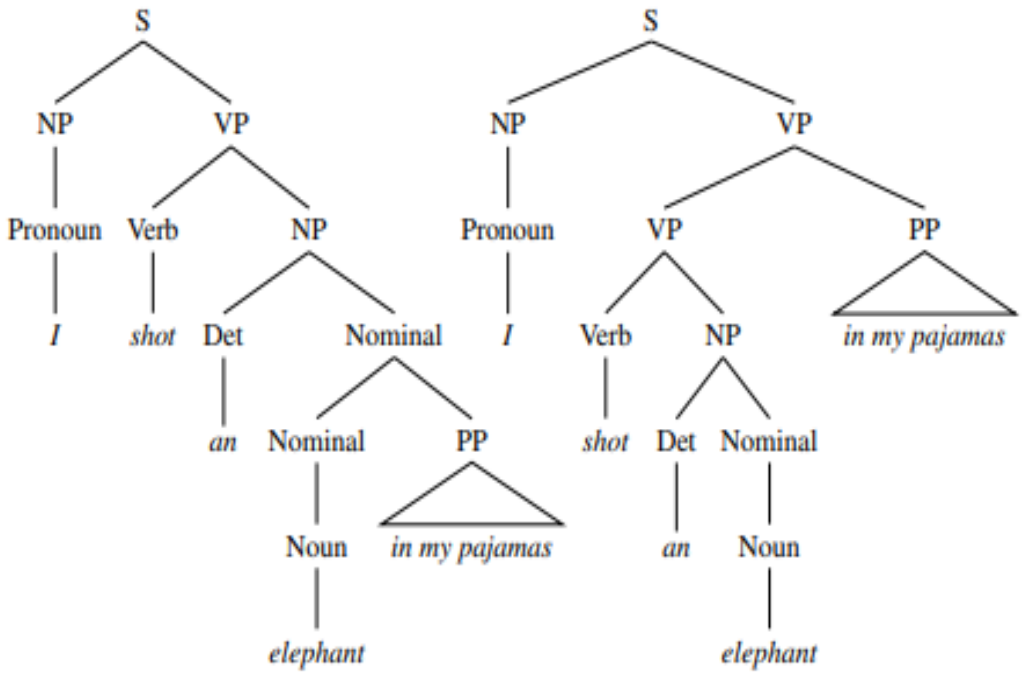


Figure 3.5: Example of ambiguity in sentence *I shot an elephant in my pajamas*

Here k is a split point such that $i \leq k < j$. As explained in section 3.3.1.2 when CFG is grammar is unavailable, it can be extracted from the treebank.
Figure 3.4 depicts the slot-filling process in CKY algorithm applied to an example sentence '*Book the flight through Houston*' with CFG as shown by Martin (2021a).

3.3.1.3 Ambiguity

Ambiguity refers to a situation when more than one parse trees can be generated based on a given CFG for a single input sentence being parsed. For example, Figure 3.5 depicts two distinct constituency parse-tree that can be generated for same input-sentence *I shot an elephant in my pyjamas* based on a common CFG.
Ambiguity is one of the most severe issues to be addressed by constituency parsing task. **Probabilistic CKY** is an extension of standard CKY algorithm which can generate the parse-tree of a given input sentence while effectively addressing the ambiguity issue. Section 3.3.1.4 will describe the probabilistic CKY in details.

3.3.1.4 Probabilistic CKY

As already explained, the statistical approaches to the constituency-parsing typically involves a two-step process. Firstly given a training treebank corpus in any language L, these approaches extract CFG grammar of it. Subsequently in the second step,

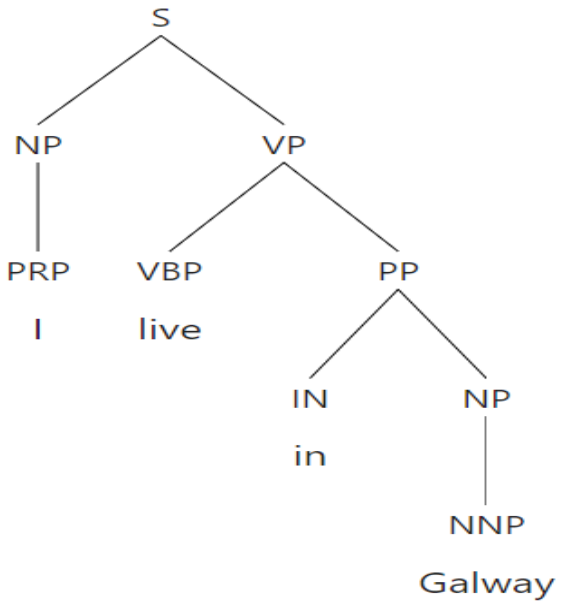


Figure 3.6: Parse tree of an example sentence *I live in Galway*

for an input sentence S, these approaches use CKY algorithm to generate parse-tree for the input sentence. However if S has ambiguity, the algorithm will generate more than one parse-trees.

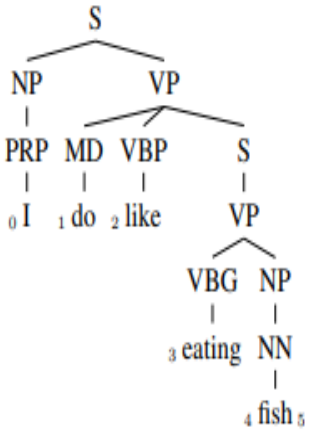
The Probabilistic CKY Booth (1969); Baker (1979) on the other hand, aims to extract the ***Probabilistic CFG*** instead of standard CFG from the training corpus. For any CFG rule $A \rightarrow B\ C$, its probability can be calucted by applying equation 3.2

$$Pr(A \rightarrow B\ C) = \frac{Count(A \rightarrow B\ C)}{\Sigma_{All\ X,Y \in CNF_L} Count(A \rightarrow X\ Y)} \tag{3.2}$$

Here $Count(A \rightarrow B\ C)$ is the number of occurrences of rule $A \rightarrow B\ C$ in the training dataset. $\Sigma_{All\ X,Y \in CNF_L} Count(A \rightarrow X\ Y)$ indicates the total number of occurrences of all the rules in CFG with A on the left-hand side.

For any given tree, its probability can be computed as the product of the probability of all its nodes. For example, Figure 3.6 depicts the constituency parse-tree T of an example sentence ***I live in Galway***. The probability of this tree can be computed by applying equation 3.3.

$$\begin{aligned} Pr(T) = Pr(PRP \rightarrow I)\ &*\ Pr(VBP \rightarrow live)\ *\ Pr(IN \rightarrow in) \\ &*\ Pr(NNP \rightarrow Galway)\ *\ Pr(PP \rightarrow IN\ NP) \\ &*\ Pr(VP \rightarrow VBP\ PP)\ *\ Pr(S \rightarrow NP\ VP) \end{aligned} \tag{3.3}$$



steps	structural action	label action	stack after	bracket
1–2	sh(I/PRP)	label-NP	$_0\triangle_1$	$_0$NP$_1$
3–4	sh(do/MD)	nolabel	$_0\triangle_1\triangle_2$	
5–6	sh(like/VBP)	nolabel	$_0\triangle_1\triangle_2\triangle_3$	
7–8	comb	nolabel	$_0\triangle_1\triangle_3$	
9–10	sh(eating/VBG)	nolabel	$_0\triangle_1\triangle_3\triangle_4$	
11–12	sh(fish/NN)	label-NP	$_0\triangle_1\triangle_3\triangle_4\triangle_5$	$_4$NP$_5$
13–14	comb	label-S-VP	$_0\triangle_1\triangle_3\triangle_5$	$_3$S$_5$, $_3$VP$_5$
15–16	comb	label-VP	$_0\triangle_1\triangle_5$	$_1$VP$_5$
17–18	comb	label-S	$_0\triangle_5$	$_0$S$_5$

Figure 3.7: Parsing of sentence *I do like eating fish* by Cross and Huang (2016)

However, the product of all probabilities can lead to the problem of arithmetic underflow. Thus, for practical purposes, instead of the product of all probabilities, the sum of log probability values are used instead as evident in equation 3.4.

$$Score(T) = \log(Pr(T)) = \log(Pr(PRP \rightarrow I)) \ + \ \log(Pr(VBP \rightarrow live)) \\ + \ \log(Pr(IN \rightarrow in)) + ... + \log(Pr(VP \rightarrow VBP \ PP)) \ + \ \log(Pr(S \rightarrow NP \ VP)) \tag{3.4}$$

For a sentence with ambiguity, the approach usually outputs the tree with maximum probability/log-probability sum as is correct parse-tree. Weighted CKY Mohri and Pereira (1998) is similar to Probabilistic CKY, where each CFG rule is assigned with the weights. Subsequently, the maximum weighted tree is outputted as the correct tree to resolve ambiguity.

3.3.2 Neural approaches

In the previous section we described CKY based statistical approaches to monolingual constituency parsing. However, with the advancement of neural networks, modern state-of-the-art approaches to CP task are NN based approaches that significantly outperform statistical approaches. This section describes the significant NN approaches to CP task in details. On the other hand, section 3.5 will describe various cross-lingual and unsupervised approaches to CP task.

Recurrent Neural Network Grammar (RNNG) Kuncoro et al. (2016) is the first significant LSTM based approach to CP task that outperformed the state-of-the-art CKY based approach. It is inspired by the *Transition-based approaches* to Dependency parsing task discussed in section 5.1.3.1. The approach comprises a **Stack** which

stores the incomplete parse-tree, a **Buffer** which stores the sentence tokens and the set of all possible **Actions**. The approach is an iterative approach that chooses best action at each time-step and update the stack and buffer accordingly. The action is chosen based on current states of stack and buffer as well as action-history. The process is continued until the Buffer becomes empty and Stack consists of completed parse-tree. We will describe the RNNG approach in more detail in section 3.6 as our proposed cross-lingual approach to CP called **UniRNNG** (section 3.4) is based on the RNNG approach indeed.

Similar to RNNGs, Cross and Huang (2016) is another approach which is inspired by and modifies the *Transition-based approaches* to DP. The approach uses a stack and a buffer in a similar way as RNNG but the stack contains sentence spans (word-sequences) with no requirement to be the part of an incomplete parse tree. Furthermore, the action set of Cross and Huang (2016) comprises of two types of action namely the **structural** actions and the **label** actions. The **Structural actions** includes two actions namely ***Shift*** which is similar to the *Shift* action in RNNG and the ***Combine*** action which merges the top-two sentence spans into one. The ***Combine*** action is similar to the *Reduce* action of RNNG but it does not aim to create an incomplete parse-tree structure and is non-directional. The **Label actions** assign a label to the sentence span on top of the stack. Figure 3.7 depicts an example sentence *I do like eating the fish* been parsed, as shown by Cross and Huang (2016).

Charniak et al. (2016) is another significant approach which aimed to extract the parse-tree of a sentence by language-modelling. Language Modeling (LM) Chelba and Jelinek (2000) typically is a probability distribution over all possible sentences (word-sequences) in a language. Given an input sentence $x = x_1, x_2, ...x_N$ an LM computes the probability value Pr(x) of sentence x by applying equation 3.5.

$$P(x) = P(x_1, x_2, ...x_N) = \prod_{t=1}^{N} P(x_t | x_1, x_2, ...x_{t-1}) \tag{3.5}$$

Inspired by the Language-modeling task, Charniak et al. (2016) approach represented entire parse-tree $T(x)$ of an input sentence x as a sequence of symbols $T(x) = T_1, T_2, \ldots T_M$. For example, the representation of an example parse-tree (of example sentence *I live in Galway*) shown in Figure 3.6 can be depicted as follows

$$(S\ (NP\ I\ )_{NP}\ (VP\ live\ (PP\ in\ (NP\ Galway\ )_{NP}\ )_{PP}\ )_{VP}\ )_{S}$$

. For a given training treebank, the authors first represented all the trees within the dataset as sequence of symbols. Subsequently they trained an LSTM based Language-

model over all these sequences. During inference, for a new input-sentence the approaches calculate the probability of all possible parse tree using this pre-trained Parse-tree LM to output the most probable tree.
Kitaev and Klein (2018) introduced the first Transformer based approach to CP task (section 2.1.1.4). The model adopts a simple encoder-decoder framework. In this framework, the entire model architecture comprises an encoder NN which encodes any input sentence s into a representation matrix, and a decoder network which takes-in the encoder output as its input and probabilistically builds the desired parse-tree $T(s)$.
The architecture of the encoder NN is inspired by the Transformer model Vaswani et al. (2017). Given an input sentence $x = x_1, x_2, \ldots, x_N$ of length N, the approach represents it as a feature-matrix $e \in R^{N*d}$ where $e = [e_{w_1}, e_{w_2}, \ldots, e_{w_N}]$. Here $e_{w_i} \in R^d$ is the contextual word-embedding. The embedding-matrix is appended with the POS-tag representation matrix $m = [m_1, m_2, \ldots, m_N]$ of dimension R^{N*1} where m_i is the tag-id of POS-tag of word w_i.
Hence, the input to the encoder is a feature-matrix of dimension $R^{N*(d+1)}$. This input feature-matrix is appended with positional encoding to embed word-sequence knowledge. Similar to Vaswani et al. (2017), the encoder NN architecture comprises eight layers. Each layer comprises of a self-attention layer and a fully connected layer. Finally, the encoder outputs $z \in R^{N*k}$ which is inputted into the decoder network. The decoder network can assign the tree-score $Score(T)$ to a given parse-tree T by applying equation 3.6.

$$Score(T) = \Sigma_{(i,j,l)\in T} score(i, j, l) \tag{3.6}$$

Here $score(i, j, l)$ is the node-score of any constituent that is located between the positions i and j of the input sentence and has label l. The approach introduces a dummy label ϕ to represent the non-existence of a node. For a new input sentence $\hat{s}$, it's parse tree $\hat{T}$ is computed by applying equation 3.7.

$$\hat{T} = argmax_T(Score(T)) \tag{3.7}$$

3.4 UniRNNG Introduction

As explained in chapter 2, Noam Chomsky proposed the hypothesis of **Universal Grammar (UG)** which states that all human languages, while being superficially as diverse as they are, share some fundamental similarities. Thus he argues that

deep down the specific grammars of various natural languages, there exists a *Universal Grammar*. Since then many linguists Baker (2008); Fodor and Sakas (2004); Tomasello (2005); Pinker (1995); Fodor (2001) attempted to outline the *principles and parameters* of this *Universal Grammar* manually, but with very limited success. If it is nearly impossible to identify and outline UG manually due to its anticipated large size and complexity Roberts and Holmberg (2005); Kayne (2012); Cinque and Rizzi (2010); Shlonsky (2010); we can use a neural network to learn these automatically.
RNN based models such as *Recurrent Neural Network Grammars (RNNG)*Dyer et al. (2016) (explained in section 3.6) are proven to do excellent job in automatically learning and encoding (as model-parameters) the grammar of any language directly from its tree-bank corpus. This inspired us to make following assumption:

A Recurrent Neural Network based multi-lingual parser trained on a diverse polyglot treebank corpus would learn and encode the *Universal Grammar* as its model-parameters.

Based on this assumption, we proposed and evaluated **Universal Recurrent Neural Network Grammar (UniRNNG)** which is a multi-lingual variant of the Dyer's RNNG model Dyer et al. (2016).
The architecture of **UniRNNG** is indeed inspired by the *Principle and Parameter* framework Chomsky (1993) explained in section 2.2.2.1. Hence unlike Dyer's RNNG, our proposed model comprises two sets of model-parameters α and β. α would encode the *Universal Principles* which are shared by all the languages and β would encode *Parameters* which are tuned to specific language of the sentence being parsed during run-time.
In order to generalize a mono-lingual constituency parsing model to multi-lingual settings, we utilized the knowledge of *Language typology* which is available as various typological feature-values in ***World Atlas of Language System (WALS)*** Haspelmath (2009) database.
As discussed in section 2.1.1.1, the CLT based approaches do not perform well if the source and target languages are typologically very distinct Ruder et al. (2019a). But since *UniRNNG* explicitly models over the typological features (as inputs) and is trained on a sufficiently diverse polyglot corpus, it is comparatively more robust to the typological differences between source and target languages. In other words, once

being trained on sufficiently large and typologically diverse corpus it can be applied to any natural-language thus making it *Language-Agnostic*.

3.5 Approaches to CP for low-resource languages

Section 3.3 described various mono-lingual supervised approaches to the CP task, including the state-of-the-art neural-network approaches. However, all these approaches are supervised approaches which require a large amount of labelled training data-set, thus limiting their utility to only a handful of high-resource languages. In this chapter, we describe our proposed **Universal Recurrent Neural Network Grammars (UniRNNG)** which is a multi-lingual variant of the **Recurrent Neural Network Grammars (RNNG)** model for constituency parsing (section 3.6), to address this issue of data-sparsity for low-resource languages.
UniRNNG is a *Cross-lingual Transfer Learning* based approach to CP task. The architecture of UniRNNG is inspired by *Principle and Parameter* theory proposed by Noam Chomsky. Furthermore, UniRNNG utilises the linguistic typology knowledge available as feature-values within WALS database, to generalize over multiple languages. Once trained on sufficiently diverse polyglot corpus **UniRNNG** can be applied to any natural language thus making it *Language-agnostic* constituency parser. Section 3.4 provides the introduction to out proposed *UniRNNG* model. In section 3.5, we provide a brief literature-review of previously proposed cross-lingual approaches to the CP task.

3.6 RNNG model

Recurrent Neural Network Grammar (RNNG) Kuncoro et al. (2016) model is a probabilistic RNNG based parser which models the hierarchical and nested relationships between words and phrases of an input sentence. RNNGs are indeed reminiscent of PCFG (section 3.3.1.4) but the grammar is represented as RNN model parameters instead of CFG rules.
The proposed RNNG based approach is a top-down variant of the standard transition-based parsing which is commonly used for the dependency parsing task (section 5.1.3.1). Formally the authors defined RNNGs with a tuple (N, Σ, θ) where N is the set of non-terminals, Σ is the set of terminal nodes and θ is the optimized parameters of the RNN which performs the generation or the discrimination task. The authors provide two variants of the transition-based parser namely the **Generative**

parser and the **Discriminative parser**.
As already explained, similar to the standard transition-based approach to dependency-parsing task (section 5.1.3.1), the RNNG approach also comprises of a Stack S, a buffer B and the action-set A. For a given input sentence x to be parsed, the stack S would comprise an incomplete parse-tree and the buffer B would comprise of tokens of x. At each time-step t, the approach chooses the best action $a_t \in A$, given the current state of stack S_t, buffer B_t and history of actions $a_{<t}$. Depending upon the chosen action a_t, the Stack and Buffer are updated to S_{t+1} and B_{t+1} respectively. The process is continued until the Stack consists of completed parse-tree.

3.6.1 Discriminative vs Generative

For a given input sentence x and its parse tree $T(s)$, the Generative RNNG computes the probability of generating a complete tree T(s) along with non-terminal nodes as $Pr(T(s))$. On the other hand, the Discriminative RNNG computes the probability of assigning the tree $T(s)$ to input sentence s as $Pr(T(s)|s)$. During the inference time therefore the *Discriminative RNNG* model predicts the correct parse-tree $T*$ of an input sentence s as follows

$$T* = \arg\max_{T \in T'} Pr(T(s)|s)$$

Whereas the *Generative RNNG* predicts the correct parse-tree $T*$ of an input sentence s as follows

$$T* = \arg\max_{T \in T'} Pr(T(s))$$

Here T' is the set of all possible trees for input sentence s.
Both *Discriminative* and *Generative* RNNGs follow *Transition based* parsing framework but differ in the set of possible actions. Tables 3.6 and 3.7 lists the action-set for *Discriminative* and *Generative* RNNGs respectively.

3.7 UniRNNG Model

This section describes our proposed **Universal Recurrent Neural Network Grammar (UniRNNG)**. As being a multi-lingual variant of *DiscRNNG* (section 3.6), the **UniRNNG** is also a transition based parser consisting of a *Stack* S, *Buffer* B and *action-set* A. At any time-step, the *Stack* stores incomplete parse-tree and *Buffer* stores token-sequence. At each time-step t, model predicts best action $a_t \in A$ given current state of Stack (S_t), Buffer (B_t) and Action-history ($a_{<t}$). Subsequently *Stack* and *Buffer* are updated as S_{t+1} and B_{t+1}, according to action a_t.

Action	**Description**
NT(X)	Opens a non-terminal node 'X' and puts it on top of *Stack*. eg: NT(VP)==>(VP
SHIFT	Removes topmost token from the *Buffer* B and pushes onto the Stack
REDUCE	Repeatedly pops completed sub-trees or terminal symbols from the stack until an open non-terminal is encountered, and then this open NT is popped and used as the label of a new constituent that has the popped sub-trees as its children. This new completed constituent is pushed onto the stack as a single composite item.

Table 3.6: Action Set for *Discriminative RNNG* Dyer et al. (2016)

Action	**Description**
GEN(w)	Generates a new word w (terminal node) and puts it at the end of buffer
NT(X)	Opens a non-terminal node 'X' and puts it on top of *Stack*. eg: NT(VP)==>(VP
REDUCE	Repeatedly pops completed sub-trees or terminal symbols from the stack until an open non-terminal is encountered, and then this open NT is popped and used as the label of a new constituent that has the popped sub-trees as its children. This new completed constituent is pushed onto the stack as a single composite item. It can only be applied when the top of the stack is not an open non-terminal symbol and size of stack is greater than one.

Table 3.7: Action-set of Generative RNNGs

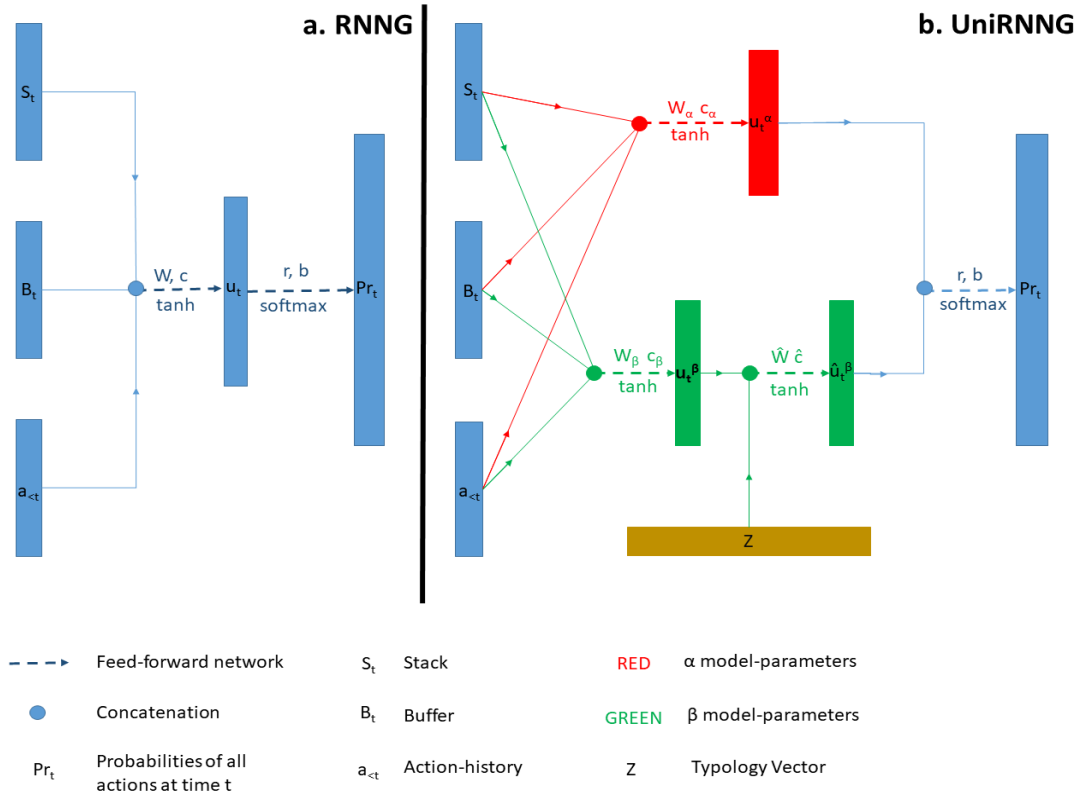


Figure 3.8: a. Recurrent Neural Network Grammar (RNNG) architecture. b.Universal Recurrent Neural Network Grammar (UniRNNG) architecture.

3.7.1 Architecture

Figure 3.8b depicts the architecture of the **UniRNNG**. At each time-step t the proposed model computes the Stack-encoding S_t, Buffer encoding B_t and action-sequence encoding $a_{<t}$ using stack-LSTM and RNN respectively, in similar way as *DiscRNNG*. (Section 3.6). However for **UniRNNG** *Cross-lingual Word-Embeddings* are used instead of Word-Identifier vectors during encoding of Stack and Buffer.

Once having computed S_t, B_t and $a_{<t}$ the model computes two distinct vector-representations of the entire model-state at time t namely α-*vector* (${u^\alpha}_t$) and β-*vector* (${u^\beta}_t$), unlike *DiscRNNG* which computes single representation u_t. The ${u^\alpha}_t$ and ${u^\beta}_t$ are computed through equations 3.8 and 3.9.

$${u^\alpha}_t = tanh(W^\alpha[S_t; B_t; a_{<t}] + c^\alpha) \tag{3.8}$$

$${u^\beta}_t = tanh(W^\beta[S_t; B_t; a_{<t}] + c^\beta) \tag{3.9}$$

A *typology aware version of* β-*vector* $\hat{u}_t^\beta$ is computed by applying equation 3.10 (computation simply involves concatenation and dimension reduction through feed-forward

network).

$$\hat{u}_t^{\beta} = tanh(\hat{W}[{u^{\beta}}_t; Z] + \hat{c}) \tag{3.10}$$

Here $Z \in R^{|Z|}$ is a *Linguistic-typology* vector. Each value within Z represents a single typology-feature from *WALS* Haspelmath (2009) database having specific value as integer for the language being parsed. Both ${u^{\beta}}_t$ and $\hat{u}_t^{\beta}$ have same dimensions i.e. R^d. Final state-representation at time t is given as concatenation of *α-vector* (${u^{\alpha}}_t$) and *typology aware version of β-vector* ($\hat{u}_t^{\beta}$) as equation 3.11.
Missing features for any language are assigned *zero* indicating no dominant value for it.

$$u_t = [{u^{\alpha}}_t; \hat{u}_t^{\beta}] \tag{3.11}$$

To summarize *UniRNNG* is very similar to Dyer's *Discriminative RNNGs (DiscRNNG)* model 3.6 with the following modifications.

1. Cross-lingual Word-embeddings are used instead of unique word-identifiers
2. At each time-step t, two distinct model-state representations are computed namely *α-vector* ${u^{\alpha}}_t$ and *β-vector* ${u^{\beta}}_t$.
3. Final model-state representation u_t is computed as the concatenation of *α-vector* and *typology aware version of β-vector*. This is unlike original *DiscRNNG* where u_t is computed directly from S_t, B_t and $a_{<t}$
4. Model is trained on a typologically diverse polyglot corpus.

The proposed architecture is inspired by the *Principle and Parameter framework* Chomsky (1993) framework proposed by linguists *Noam Chomsky* and *Howard Lasnik* Chomsky (1993). The central idea behind the PP framework is that a person's syntactic knowledge can be modelled with two formal attributes namely a finite set of fundamental ***Principles*** that are shared by all languages (e.g.: A sentence must always have a subject) and a finite set of ***Parameters*** whose values characterize syntactic variability amongst various languages (eg: *Subject-Verb-Object* (S-V-O) order within a sentence).
Inspired by this PP theory, our proposed *UniRNNG* architecture comprises of distinct α (W^{α},c^{α}) and β (W^{β},c^{β}) parameters to encode the universal and language specific features.

Language	**Tree-bank**	**Family**
en	Penn tree-bank Marcus et al. (1993)	Germanic
sd	Talbanken05 Nivre et al. (2006b)	Germanic
fr	FrenchTreebank Abeillé et al. (2003)	Romance
es	Spanish UAM Treebank Moreno et al. (1999)	Romance
jp	Tüba-J/S Kawata and Bartels (2000)	Altic
ar	Arabic PENN Treebank Maamouri et al. (2004)	Afro-asiatic
hu	Hungarian Szeged Treebank Treebank	Uralic

Table 3.8: List of source languages and their corpora used during experimentation. corpora are used to train both *Word-Embeddings* and *Parsers*

Language	**Tree-bank**	Family
de	Negra Treebank Skut et al. (1997)	Germanic
da	Arboretum Treebank Bick (2003)	Germanic
it	ISST Treebank Montemagni et al. (2003)	Romance
ct	Catalan AnCora Treebank Taulé et al. (2008)	Romance
kr	Korean Penn Treebank Han et al. (2002)	Altic
hb	Heberew Treebank Sima'an et al. (2001)	Afro-asiatic
et	Estonian Arborest Treebank Bick et al.	Uralic
hi*	Hindi-Urdu Treebank Bhat et al. (2017)	Indo-aryan
vt*	Vietnamese Treebank Nguyen et al. (2009)	Austroasiatic

Table 3.9: List of target languages and their corpora used during experimentation. corpora are used to train both *Word-Embeddings* and *Parsers*. * these languages are used only in zero-shot settings

3.8 Experiments

This section describes the experiments conducted to evaluate the performance of proposed **UniRNNG**. There are two key novel objectives of these experiments, listed as follows:

1. To evaluate whether the Monolingual Recurrent Neural Network Grammar (RNNG) based model for the CP task is effective within the Cross-lingual settings.

2. To evaluate how the linguistic typology knowledge induction impact the performance of RNNG based Constituency parser within various cross-lingual settings.

As far as we are aware, this is first work to explore the use of linguistic typology knowledge for the cross-lingual CP task. We compared the performances of both cross-lingual variant of RNNG as well as our proposed UniRNNG model with numerous baselines, in both few-shot and zero-shot learning settings.
Each of the experiments comprises of a set of source languages L_s and a single target language l_t.

3.8.1 Experimental Settings

We evaluated the performance of **UniRNNG** under two experimental setups namely *Few-shot learning* and *Zero-shot learning* setups.
Few-shot Learning Wang and Yao (2019) is applied when only few training examples are available in the *target language*. In this setup, the cross-lingual models (baseline and **UniRNNG**) are trained on a mixed corpus comprising of source-language sentences (covering over 80% corpus) and few available target language sentences. Hence for *Few-shot Learning* setup $l_t \in L_s$.
Zero-shot Learning Xian et al. (2017) is applied when no labelled dataset is available in the *target language*. Hence $l_t \notin L_s$.

3.8.2 Baselines

This section describes the baselines used to compare the performance of our proposed *UniRNNG*.

3.8.2.1 Mono-lingual Models trained on Sparse Dataset

We used this baseline to compare the performance of our proposed *UniRNNG* only in the *Few-shot* learning settings. As our *UniRNNG* model is intended to be applied for low-resource languages, we compare the performance of it with that of the state-of-the-art mono-lingual models trained on a sparse dataset. We experiment with three mono-lingual constituency parsers namely *DiscRNNG* 3.6, Kuncoro et al. (2016) and Transformer Vaswani et al. (2017).
These models provide over 95% F-Score when trained with sufficiently large dataset. But they would not show such high performance when trained on sparse dataset.

3.8.2.2 Unsupervised Recurrant Neural Network Grammar (URNNG)

Its a state of the art approach to *unsupervised constituency parsing*. We used this baseline to compare the performance of our proposed *UniRNNG* only in the *Zero-shot* learning settings.

3.8.2.3 Cross-lingual RNNG Parser trained on single source language (CL-RNNG-Mono)

It is the baseline Dyer's RNNG model Dyer et al. (2016) evaluated within the cross-lingual settings. To evaluate the model in cross-lingual settings we made two key modifications described as follows. Firstly the *Cross-lingual Word Embeddings* Ruder et al. (2019b) are used rather than unique word-identifier vectors as used by Dyer et. al. Secondly the model is trained on a single source language *English* (UniRNNGs are trained on polyglot corpus) and tested on multiple target language. Within *Few-shot learning*, the training corpus also include small number of labelled target language sentences.

3.8.2.4 Cross-lingual RNNG Parser trained of multiple source languages (CL-RNNG-Poly)

It is the same model as described in 3.8.2.3, but trained on a mixed polyglot corpus of high-resource source languages.(*CL-RNNG-Mono* is trained on a single source language *English*). Similar to 3.8.2.3, a small number of labelled target-language l_t sentences are included as part of the training corpus within the *Few-shot* settings.

Hyper-parameter	**Value**
WE dims	768
S_t,B_t,$a_{<t}$ dims	450
${u^{\beta}}_t$, ${u^{\alpha}}_t$ dims	450
Dropout prob.	0.01
Bach-size	32
Number of steps per epoch	Size of training corpus / 32
Epochs	150
BERT Model	bert_multi_cased_L-12_H-768_A-12

Table 3.10: Hyper-parameter settings

3.8.3 Dataset

Tables 3.8 and 3.9 list all the *Source* and *Target* languages as well as their tree-bank corpora used during experimentation. We evaluated our proposed *UniRNNG* model and all the baseline models on each of the target languages listed in Table 3.9 independently.

As already explained in section 3.8.1, the *CL-RNNG-Mono* parsers (3.8.2.3) are always trained on the single source-language *English*, whereas the *CL-RNNG-Poly* and the *UniRNNG* Parsers are always trained on a mixed polyglot corpus (in both *few-shot* and *zero-shot* setups). For each experiment, the source-language training corpus size is always fixed to 700,000 tokens to ensure controlled experiment-settings.

We created the source-language training-corpus for *CL-RNNG-Mono* parsers by randomly sampling sentences from the English-PTB corpus (one at a time), until the token-size becomes approximately equal to 700,000. On the other hand, to create the source-language training-corpus for *CL-RNNG-Poly* and *UniRNNG* models, we randomly sampled sentences from each of the seven source-language corpora listed in Table 3.8 until the token-size becomes approximately equal 100,000, concatenated all these sampled datasets and randomly shuffled the order. Hence all the seven source-languages listed in Table 3.8 are equally represented in the training-corpus for *CL-RNNG-Poly* and *UniRNNG* models.

3.8.3.1 Short tree-bank corpora

As explained in section 3.8.1, within *Few-shot learning* settings, only sparse target-language dataset should be used to train both *UniRNNG* and *Baselines*. Hence we

extracted a small subset of entire large treebank corpus for each target language listed in Table 3.9.
We extracted this subset by randomly sampling sentences from the target-language tree-bank corpus until the token-size becomes approximately equal to 3000. This is inspired by the works of Ammar et al. (2016) who used same yardstick to evaluate their *Multi-lingual Dependency Parser (MALOPA)*. This small target-language language corpus is added to the source-language training corpus for each experiment, within *Few-shot Learning* setup.

3.8.4 Universal Annotation

There are numerous tree-bank corpora for a diverse range of languages being developed during the years (some listed in Tables 3.8 and 3.9). But unlike *Dependency Parsing* tree-banks which are mostly annotated with the *UD Annotations* McDonald et al. (2013) (for most languages), in case of *Constituency Parsing* various existing tree-bank corpora have their own independent tag annotations, thus making the application of multi-lingual approaches to it as impossible.
However Han et al. (2014) proposed a *Universal Phrase tag-set* with 9 common Phrase-tags. Furthermore Han et al. (2014) also provides a mapping table to map tags of popular constituency tree-banks (including all treebanks used by us in our experiments) to these *Unversal Phrase Tags*.
We used this mapping table to replace all tags within all the tree-banks listed in Tables 3.8 and 3.9, with the universal tags. Subsequently we trained and evaluated all approaches (including baseline mono-lingual approaches) on these *Universally Tagged* tree-bank versions.

3.8.5 Cross-Lingual Word Embedding

As our model is a polyglot, we use *Cross-lingual Word-embeddings* during the encoding of Stack and Buffer state at any time-step t. We use a simple *Linear transformation based approach* Ruder et al. (2019b) to compute such *Cross-lingual Word-embeddings*.
Given two languages l_1 and l_2, the simple *Linear Transformation* based approach first trains the mono-lingual WE for both l_1 and l_2 independently. Subsequently, it uses a bi-lingual lexicon to learn a transformation matrix W^{l_1,l_2} to project embeddings of words of l_1 to the embedding-space of l_2 (considering l_2 as reference language).
To ensure that all WE are within the same space, we use *English* as a reference

Model	de	da	it	ct	kr	hb	et
Transformers Vaswani et al. (2017)	34.34	33.08	34.71	33.74	35.58	35.60	35.57
DiscRNNG 3.6	34.49	33.52	35.01	34.15	36.02	35.74	35.94
Kuncoro et al. (2016)	34.98	33.68	35.53	34.46	36.3	36.42	36.23
CL-RNNG-Mono+Skip-Gram	65.63	70.85	54.59	58.05	22.95	30.44	53.43
CL-RNNG-Mono+Fast-text	67.13	72.55	56.39	60.35	24.75	31.94	55.83
CL-RNNG-Mono+Glove	68.73	74.15	57.29	61.15	25.45	33.84	55.93
CL-RNNG-Mono+ELMo	69.13	74.75	58.49	61.64	26.65	33.94	56.73
CL-RNNG-Mono+BERT	**71.03**	**77.35**	60.39	63.05	27.75	39.84	59.93
CL-RNNG-Poly+SkipGram	61.94	62.89	64.0	64.53	61.88	63.19	62.76
CL-RNNG-Poly+Fast-text	63.57	64.51	65.78	66.53	64.3	64.84	65.55
CL-RNNG-Poly+Glove	65.1	66.17	66.5	67.4	64.72	66.59	65.51
CL-RNNG-Poly+ELMo	65.48	66.86	67.61	68.16	65.89	66.64	66.01
CL-RNNG-Poly+BERT	67.48	69.41	69.55	70.46	69.18	69.88	69.19
UniRNNG+SkipGram	64.92	65.95	66.79	67.35	65.05	66.24	65.83
UniRNNG+Fast-text	66.42	67.65	68.59	69.64	67.05	67.74	68.23
UniRNNG+Glove	68.03	69.25	69.49	70.45	67.55	69.64	68.33
UniRNNG+ELMo	68.42	69.85	70.69	70.94	68.75	69.74	69.13
UniRNNG+BERT	70.33	72.44	**72.59**	**73.35**	**71.85**	**72.64**	**72.33**

Table 3.11: F1 Score in *Few-shot* learning settings. *Top:* Results for supervised approaches trained on sparse dataset. *Middle:* Results for baseline Cross-lingual Transfer Parser (CLT-P). *Bottom:* Results for proposed **UniRNNG**

Model	de	da	it	ct	kr	hb	et	hi	vt
URNNG Kim et al. (2019)	11.84	11.58	10.53	12.43	9.97	10.46	8.52	9.36	3.12
CL-RNNG-Mono+BERT	**68.13**	**70.94**	61.99	56.85	20.91	27.82	52.61	48.66	37.61
CL-RNNG-Poly+BERT	64.43	64.13	64.5	66.37	63.32	64.99	63.5	56.2	57.21
UniRNNG+BERT	67.62	67.03	**67.19**	**69.14**	**66.25**	**68.14**	**66.63**	**59.23**	**60.11**

Table 3.12: F1 Score in *Few-shot* learning settings.

language. Mono-lingual WE of any other language l are thus transformed into the English space by learning the transformation matrix $W^{l,e}$ from word-pairs extracted from *English-l* bi-lingual lexicon.

We experiment with five common Word-embeddings namely *Skip-gram Word2vec* Mikolov et al. (2013b), *Fast-text* Grave et al. (2018), Glove Pennington et al. (2014), ELMo Peters et al. (2018) and BERT (section 3.8.5.1). We use bi-lingual seed dictionaries provided by WOLD Haspelmath and Tadmor (2009), ASJP Wichmann et al. (2013) and IDS Key and Comrie (2015) which are elaborate multi-lingual lexical semantic databases.

3.8.5.1 BERT Word Embeddings

We computed language-independent BERT-Embeddings to be fed into UniRNNG using pre-trained Multilingual BERT (mBERT) Wu and Dredze (2019) model. mBERT is a multilingual variant of original BERT model Devlin et al. (2019) trained on text from Wikipedia in 104 languages.
The Embeddings are calculated in same way as in Kondratyuk and Straka (2019a). Given a sentence S, we tokenised the whole sentence using WordPiece tokeniser Wu et al. (2016). Subsequently we fed this token-sequence into pre-trained mBERT provided by Turc et al. (2019). Embedding of any word $w \in S$ i.e. e_w is computed by taking average of mBERT outputs of all Wordpiece tokens corresponding to word w. Thus, mBERT based Word-embeddings do not require any Linear transformation.

3.8.6 Typology and Hyper-parameters

Table 3.10 outlines the hyper-perameters used during experiments. These values are obtained by minimizing the training loss on *Development* dataset (Dev set) for *Penn Treebank Corpus* Marcus et al. (1993).
Typology vector Z includes feature-values of all word-order and constituency features in WALS Haspelmath (2009) database excluding trivially redundant features as excluded by Takamura et al. (2016).

3.9 Results

Table 3.11 outlines results obtained from experiments conducted within the *Few-shot Learning* settings. Best results for *CL-RNNG-Mono*, *CL-RNNG-Poly* and proposed *UniRNNG* models are obtained with BERT Embedding. Table 3.12 outlines results obtained for experiments conducted under *Zero-shot* learning settings. As we obtained best results with BERT Embeddings within few-shot settings, we experimented with only BERT-embeddings (3.8.5.1) in *Zero-shot* settings indeed.

3.10 Analysis

In this section we analyse the results outlined in section 3.9 to address the research questions **RQ1**, **RQ2** and **RQ3** listed in section 1.1.1 as follows.

RQ1: Can the state-of-the-art Recurrent Neural Network Grammar (RNNG)

approach to monolingual Constituency parsing be applied for cross-lingual Constituency parsing ?

It is evident in Tables 3.11 and 3.12 that our proposed Cross-lingual variant of RNNG (CL-RNNG) model trained on both monolingual and mixed polyglot corpora (referred in Tables as **CL-RNNG-Mono** and **CL-RNNG-Poly**) indeed significantly outperformed key state-of-the-art monolingual supervised approaches to CP trained on sparse dataset, on all the seven target languages on which these were tested within *Few-shot* learning settings, as well as the state-of-the-art unsupervised approach to CP task namely the **Unsupervised RNNG (URNNG)** model on all the nine target languages on which these were tested, within *Zero-shot* settings.

RQ2: Within the cross-lingual transfer-learning settings, does mixed polyglot training lead to improvement in performance of the RNNG model, as compared to single source language training ?

As *CL-RNNG-Mono* is trained on the single source language English, it is expected to perform comparatively better on the target languages which are typologically closer to English and poorer on the target languages which are typologically apart from English. On the other hand, *CL-RNNG-Poly* and *UniRNNG* are expected to perform almost uniformly on all the target languages as these are trained on typologically diverse polyglot corpora. These expected trends are in-fact observed in both the *Few-shot* and the *Zero-shot* learning settings as evident in Tables 3.11 and 3.12.
Hence for languages *da* and *de*, *Cl-RNNG-Mono* outperformed both *CL-RNNG-Poly* and *UniRNNG* as these languages belong to the same language-family as English namely *Germanic* and are indeed typologically very close to English. Whereas, on the other five target languages which are typologically and genealogically distinct from the source language English namely *it*, *ct*, *et*, *hb* and *kr*, it under-performed *CL-RNNG-Poly*.
Based on these observed trends we can infer that the polyglot training training does load to increase in the performance the cross-lingual transferring ability of the RNNG based Constituency Parser (CL-RNNG) only when the target-language is typologically very distinct from all source languages as it allows the model to better generalize over a diverse set of languages, but does not help when the source and target languages are typologically close. Such trends are observed in both *Few-shot* and *Zero-shot* settings.

RQ3: Does the performance of RNNG model within cross-lingual transfer-learning settings be improved by injecting the linguistic typology knowledge into it ?

In both *Few-shot* and *Zero-shot* settings, *UniRNNGs* significant outperformed *CL-RNNG-Poly* on all the seven target languages namely *da, de, it, ct, et, hb* and *kr* as evident in Tables 3.11 and 3.12. Hence it can be inferred inducing linguistic typology indeed leads to further improvement in Cross-lingual transferring ability of the RNNG based Constituency Parser to a typologically distinct and unseen target language.
Furthermore, in *zero-shot* learning settings, we evaluated our models on two additional target languages namely *hi* and *vi* (rightmost column in Table 3.12). Languages *hi* and *vi* belong to linguistic families *Indo-aryan* and *Austro-asiatic* respectively. None of the source languages listed in Table 3.8 belong to these linguistic families. Thus languages *hi* and *vi* are typologically very distant form all the source languages in the polyglot training corpus of *UniRNNGs*. Hence scores obtained on these languages indicate true Language Agnostic nature of **UniRNNG** architecture.
Although the performance of **UniRNNG** for these two languages is comparatively lower than its performance on other target languages listed in Table 3.9, yet it is better than the performances of *CL-RNNG-Mono* and *CL-RNNG-Poly* models. This provides an even stronger evidence that the typology knowledge injection does lead to improvement in performance, specifically on the typologically distinct unseen target languages. In other words, once trained on significantly diverse polyglot corpus, **UniRNNG** is *Language-Agnostic*.

3.11 Conclusion

In this work we evaluated the performance of state-of-the-art Recurrent Neural Network Grammar model for monolingual within the cross-lingual few-shot and zero-shot settings. As far as we are aware, this is the first work to explore a neural-network based model in cross-lingual setting. We also provided a framework to train and test cross-lingual models to CP task, despite each corpus having distinct annotation. Furthermore, this is the first work to explore the use of linguistic typology knowledge to aid a neural-network based model for cross-lingual CP task.
In this work, we proposed and evaluated *Universal Recurrent Neural Network Grammar (UniRNNG)* which is a multilingual variant of Dyer's RNNG model. The archi-

tecture of *UniRNNG* is inspired by *Principles and Parameters* theory proposed by linguist Noam Chomsky. The *UniRNNG* model is trained on a mixed polyglot corpus and utilises linguistic typology knowledge available in WALS database to improve its cross-lingual transferring ability. We evaluated the performance of *UniRNNG* in both *Few-shot* and *Zero-shot* learning settings.

The results achieved by our experiments show that cross-lingual variant of RNNG model does significantly outperform state-of-the-art unsupervised parsers as well monolingual parsers trained of sparse datasets. The results show that both polyglot training and the linguistic typology knowledge injection leads to significant improvement in performance specifically when source and target languages are typologically apart as it allows model to generalise the model over unseen languages.

Future work, would involve exploring the changes in performances of baseline and *UniRNNG* models with the varying degree of diversity in the training corpus.

Chapter 4

End-to-end Model for Typology Feature Prediction

This chapter is based on our research work published as following paper:

- **NUIG: Multitasking Self-attention based approach to SigTyp 2020 Shared Task.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at EMNLP 2020

In this chapter, we describe our proposed Multitasking model to predict the WALS Haspelmath (2009) typology features for various languages. The proposed model is a simple neural-network based architecture inspired by the Transformers Vaswani et al. (2017) model, which uses multitasking to simultaneously compute values for all WALS features for a given input language. The model is proposed as part of the *SigTyp (Special Interest Group for Typology) 2020 Shared task* Bjerva et al. (2020). The model represents each language as a five-dimensional vector comprising phylogenetic and geographical attributes namely *Longitude*, *Latitude*, *Genus-index*, *Family-index* and *Country-index*, and does not use any of the known WALS features of the respective input language, to compute its missing WALS features.
In chapters 5 and 6, we describe our proposed end-to-end multitasking neural-network models to the cross-lingual dependency-parsing and enhanced dependency-parsing tasks that utilise the linguistic typology knowledge. All of these models to be described in subsequent chapters, will be inspired by the typology-feature prediction model described in this chapter.
Section 4.1 describe the *SigTyp 2020 Shared Task*. Section 4.2 describe our proposed model architecture while subsequent sections will describe the training, experiments and results achieved.

Lang code	Name	Lat	Long	Genus	Family	Country	Features
Training Examples							
mhi	Marathi	19.0	76.0	Indic	Indo-European	IN	order of subject, object, and verb=SOV — number of genders=three
jpn	Japanese	37.0	140.0	Japanese	Japanese	JP	case syncretism=no case marking — order of adjective and noun= demonstrative-Noun
Testing Example							
abd	Abidji	5.67	-4.59	Kwa	Niger-Congo	CI	order of subject, object, and verb=SOV — number of genders=?

Table 4.1: Examples of dataset examples for SigTyp 2020 Shared Task Bjerva et al. (2020)

4.1 SIGTYP 2020 Shared Task

The SIGTYP 2020 Shared Task involved predicting the values of numerous typology features from the World Atlas of Language Structures (WALS) Haspelmath (2009) database. For the shared-task, the participants were required to build models to predict typology-feature values for languages unseen during the training time. The shared-task comprised of two sub-tasks namely the *Constrained* and the *Unconstrained* feature-prediction tasks. In *Constrained* settings, the participants were required to use only the provided training-dataset, whereas in *Unconstrained* settings the participant can use any external resources (such as additional text, pre-trained language-models etc.) in addition to the provided training dataset.

Table 4.1 depicts the structure of both the training and the test dataset. For each language, the data provides five key phylogenetic and geographical attributes namely *Longitude*, *Latitude*, *Genus*, *Family* and *Country*. Furthermore, for each language the datasets provide feature-values of all WALS typology features as a single string as evident in table 4.1 In test dataset, some feature-values are missing (indicated by ?). The models are required to predict these missing values.

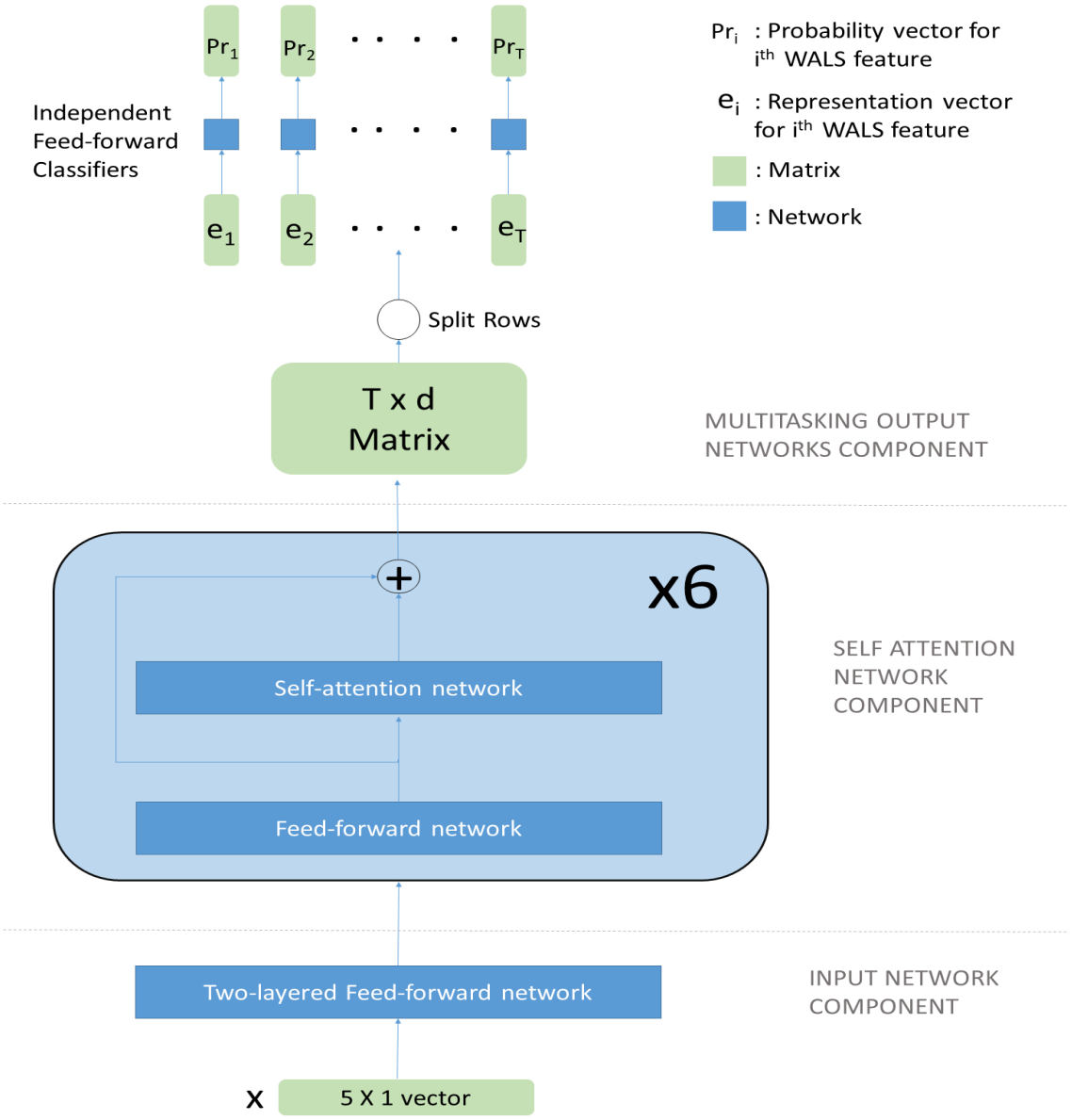


Figure 4.1: Architecture of proposed model

4.2 Model

Figure 4.1 depicts the architecture of our proposed model that computes values of all WALS typology features for a given language simultaneously. As evident in Figure 4.1, our proposed model architecture comprises three components namely *Input Network Component*, *Self-attention Network Component* and *Multitasking Output Networks Component* described in sections 4.2.1, 4.2.2 and 4.2.3 respectively.

4.2.1 Input Network Component

The input component is a simple two layered feed-forward neural network. The input of the network is a 5-dimensional vector x comprising values of five key attributes of any language, namely *Longitude*, *Latitude*, *Genus-index*, *Family-index* and *Country-index* as these are the attributes provided by train and test datasets (for all languages within the datasets) for Sigtyp 2020 Shared Task. We computed Genus-index, Family-index and Country-index from *genus*, *family* and *countryCode* attributes provided within dataset using respective name-index dictionaries.
This two layered feed forward network computes the output vector $o \in R^{T*d}$ where T is the total number of WALS typology features to be predicted by applying equations 4.1 and 4.2.

$$\hat{o} = tanh(A_1 * x + a_1) \tag{4.1}$$

$$o = tanh({A_2}^T * \hat{o} + a_2) \tag{4.2}$$

Here $A_1 \in R^{d*5}$, $A_2 \in R^{T*1}$ are weights and $a_1 \in R^d$ and $a_2 \in R^{T*d}$ are biases.

4.2.2 Self-attention Network Component

The architecture of this component is inspired by the Transformers model Vaswani et al. (2017). The model architecture comprises stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple fully connected feed-forward network. Hence the input to layer i is always the output from layer $i - 1$. Input to the first layer is the output of the previous Input Network Component.
For i^{th} layer within architecture, its *Feed-forward* and *self-attention* sub-layers are given by equations 4.3 and 4.4.

$$h_i = tanh(W_i * y_{i-1} + b_i) \tag{4.3}$$

$$k_i = attention(h_i, h_i) \tag{4.4}$$

Here $h_i \in R^d$ and $k_i \in R^d$ are outputs of *feed-forward* and *self-attention* layers respectively. We used the same attention mechanism as used by Vaswani et al. (2017). Final output of i^{th} layer y_i is computed by adding h_i and k_i (equation 4.5).

$$y_i = h_i + k_i \tag{4.5}$$

The input to the first layer y_0 is the output from the previous *Input Network Component*. The output of the *Self-attention Network Component* is the output of the final layer y_N.

It is been observed that there is a correlation between various WALS typology features. Thus, to predict the missing value of a particular typology feature for a specific languages, knowledge about other typology features for that languages would be useful. Such knowledge is ensured by the self-attention layers.

4.2.3 Multitasking Output Networks Component

The multitasking Output Networks Component comprises T independent feed-forward neural-network classifiers. The component splits the output of previous Self-attention Network Component i.e $y_6 \in R^{T*d}$ into T d-dimensional vectors $e_1, e_2, \ldots . e_T$. Each corresponds to one of the T typology features to be predicted.

The value of the j^{th} typology feature is computed by applying equation 4.6.

$$Pr_j = Softmax(W_j * e_j + c_j) \tag{4.6}$$

Here $1 <= j <= T$, Pr_j provides the probability of each of the possible values for j^{th} typology feature being the true-value. Dimensions of weights and biases are unique for each classifier as number of possible values for each of the typology features is unique.

4.3 Training

The parameters of model described in section 4.2 are trained by optimizing the loss function given by equation 4.7.

$$Loss = \Sigma_{t=1}^{T} CE(Pr_t, OH_t) \tag{4.7}$$

Here OH_t is the one-hot encoding of true-value for t^{th} typology feature. CE is the Cross-entropy loss.

Table 4.2 lists the hyper-parameters used during training. These are computed by minimizing the loss over Validation set.

Hyper-parameter	Value
d	548
drop_out probability	0.1
learning_rate	0.1
reduce lr on plateau	Yes
reduce factor	0.001
batch-size	20
steps-per-epoch	50
epochs	200
Number of features (T)	185

Table 4.2: Hyper-parameters

4.4 Results

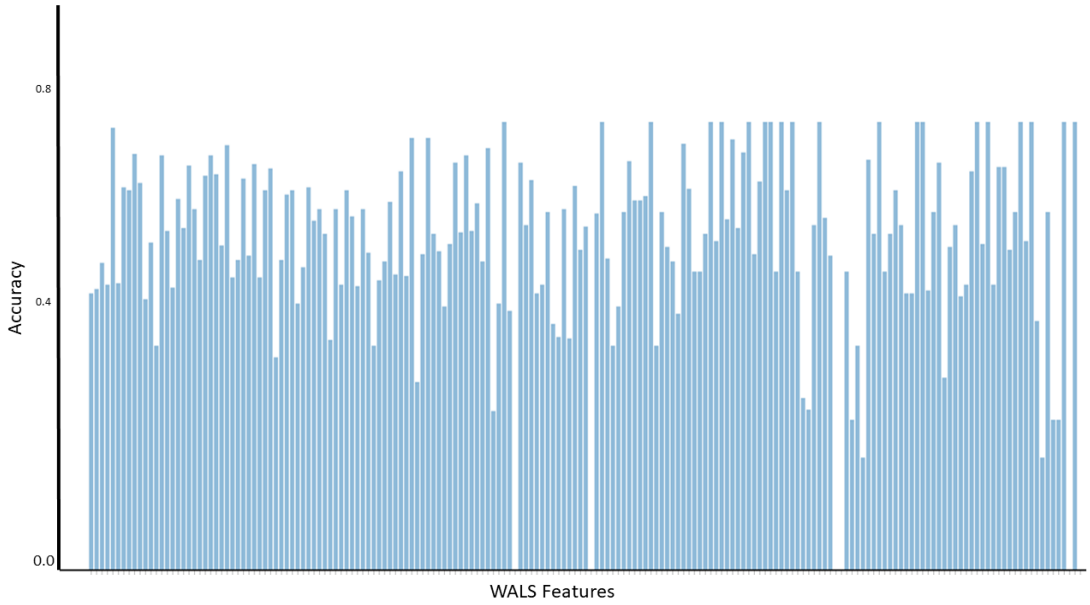


Figure 4.2: Plot depicting trend in accuracy values achieved on all WALS features

Table 4.3 compared the accuracy achieved by our proposed model with two baselines provided namely *frequency-baseline-constrained* and *knn-imputation-baseline-constrained*.

It is evident from table that our model performs at par with baselines, even though it utilizes only five attributes of the input language, namely *Longitude*, *Latitude*, *Genus-index*, *Family-index* and *Country-index* (model doesn't utilize any known WALS feature values, provided within test dataset for various languages).

Figure 4.2 is bar-plot that depicts the trend in accuracy achieved by our model on

Model	Accuracy
frequency-baseline_constrained	**0.514**
knn-imputation-baseline_constrained	0.508
NUIG_constrained	0.487

Table 4.3: Overall Accuracy of baseline and proposed models

various WALS features. Precise accuracy score achieved by our model on all 185 WALS typology features is provided in Appendix A.

4.5 Analysis and Conclusion

In this work we evaluated our proposed transformer-based model for the prediction of linguistic-typology feature-value of the specific language. The results showed that our proposed model performed at par with the state-of-the-art models.

Our model is much simpler in design than other state-of-the-art models. Furthermore all other state-of-the-art models that utilise other already known typology feature-values for the respective language to predict unknown typology feature-values. This requirement is not satisfied for most of the very low-resource languages. On the other hand, our model requires only four features namely namely *Longitude*, *Latitude*, *Genus-index*, *Family-index* and *Country-index*.

Moreover, our model's performance being at par with state-of-the-art also prove that predicting multiple typology feature-values simultaneously within the multitasking settings, does lead to improvement in performance on all. The proposed and evaluated model will provide the basis of the multitasking end-to-end dependency parser and the multitasking end-to-end enhanced dependency parser, proposed and evaluated by us in chapters 5 and 6 respectively. We describe more details in these chapters.

Chapter 5

Cross-lingual Dependency Paring with Linguistic Typology Knowledge

This chapter is based on our research work published as following papers:

- **Multitasking End-to-end BERT based Cross-lingual Dependency Parser.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at EACL 2023 (Under Review)
- **Improving the Performance of UDify with Linguistic Typology Knowledge.** In Proceedings Of SPECIAL INTEREST GROUP OF LINGUISTIC TYPOLOGY (SIGTYP) at NAACL 2021

In chapter 3, we discussed the constituency parse-tree representation scheme and the context free grammar. Subsequently, we proposed and evaluated a framework to cross-lingual constituency parsing with linguistic typology knowledge from the WALS database Dryer and Haspelmath (2013). In this chapter we discuss another significant representation of the syntax of a natural-language sentence namely *Dependency parse-tree*. In the dependency-tree framework, the syntactic structure of a sentence is represented as a set of binary head-dependent relationships between its words, instead of various phrasal structures.

Section 5.1 provides a detailed introduction to dependency-parsing (DP) framework. Section 5.2 discusses background-work related to DP for low-resource languages. In section 5.3 we outline our research objective. Finally, in sections 5.4 and 5.5 we describe in details our two proposed end-to-end cross-lingual DP models which utilise

linguistic typology knowledge. These sections will also provide details of experimentation during the evaluation of these models and compare the results achieved with other state-of-the-art approaches.

5.1 Dependency Parsing

This section provides a general background on the dependency-parsing task. Section 5.1.1 provides advantages and disadvantages of dependency parsing over constituency parsing. Section 5.1.2 provides structure of a dependency parse-tree and the issue of projectivity in dependency-parsing. Finally, in section 5.1.3 we outline various approaches to monolingual dependency parsing including the state-of-the-art NN based approaches.

5.1.1 Dependency Parsing vs Constituency Parsing

The dependency grammar (DG) of a language comprises all the possible word-level binary relationships that can exist in the language. DG is significantly different from constituency grammar (CG) which comprises of the phrase grouping rules as described in chapter 3 Thus, the dependency parsing task aims to define the syntactic structure of an input sentence by extracting all binary head-dependent relationships that exist between its words based on such DG rules. This is unlike the constituency parsing task which aims to group words of the sentence into various levels of phrase-constituents. Figure 5.1 depicts the word-level dependency relationship structure and constituency the phrase-structure analysis of an example sentence '***I prefer the morning flight through Denver***' as done by Martin (2021b). Although the lack of knowledge about phrase-structure of a sentence may lead to loss of performance in some downstream tasks, dependency grammars also have several advantages over constituency grammar.

A major advantage of DG over CG is its ability to deal with the languages with free word-order such as Czech, Russian etc. These languages are morphologically rich and have much more granularity in the word-order typology features. For example, unlike in English where the subject-verb-object (SVO) order is mostly fixed, it varies a lot from sentence to sentence in Czech. A constituency Grammar would require a separate set of rules and sub-rules accommodating each of these variations. On the other hand, since DG comprises of only binary relationships between word-pairs it is not affected by the word-orders.

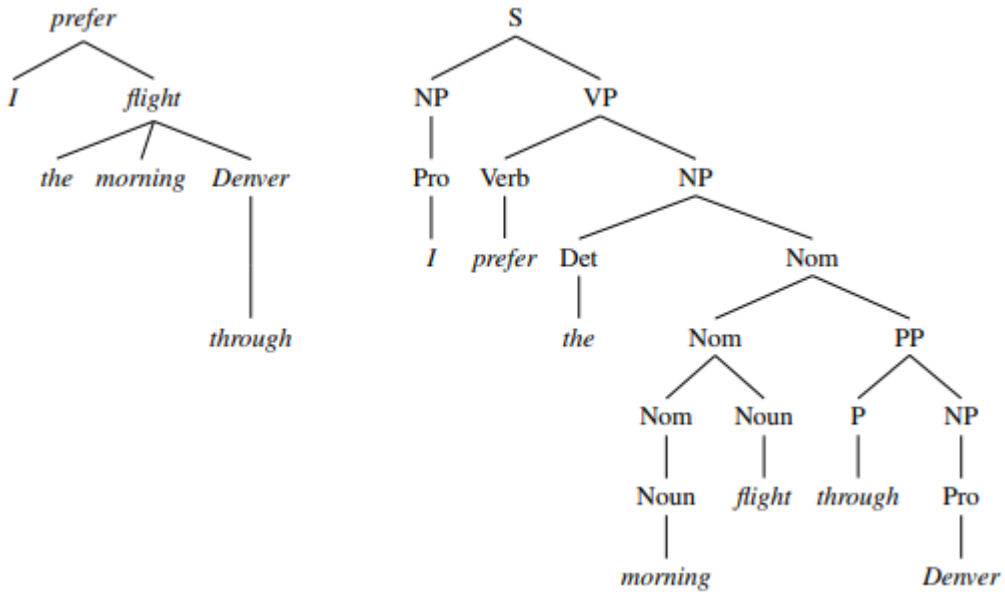


Figure 5.1: The word-level dependency relationship structure and constituency phrase-structure analysis of an example sentence '***I prefer the morning flight through Denver***'. Figure from Martin (2021b)

Another advantage of DG over CG is that since DG comprises of possible head-dependent relationships, they provide approximations of semantic properties of the language as well, such as the relationship between various predicates and their arguments. It is hard to distil such semantic knowledge from CG of a language.

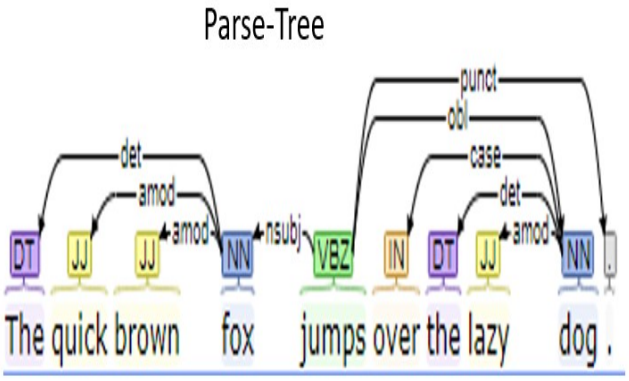
5.1.2 Dependency Tree Formulation

As already explained in section 5.1, dependency parsing is the task of creating the dependency parse-tree representation of an input sentence based on the dependency grammar rules of the language being parsed. Figure 5.2 (top) depicts the dependency-parse tree of an example sentence '***The quick brown fox jumps over the lazy dog.***'

Here, each connection between a pair of words represents a single dependency-relation. The direction of relation is from the head-word to the dependent-word. The label of each connection indicates the type of relations. Table 5.1 lists most common dependency relationships and their representative tags under the most widely used UD annotation scheme described in section 5.2.1 In Figure 5.2 (top), the base verb-form (*jumped*) is the root-node and therefore does not have any incoming arch. The root node is identified as the centre of clause structure while all other words in the sentence are either directly or indirectly connected to the root verb through the

Tag	Relationship
Nominal Relationships	
nsubj	Noun Subject
obj	Object
iobj	Indirect Object
obl	Oblique Nominal
vocative	Vocative
expl	Expletive
dislocated	Dislocated Elements
nmod	Nominal Modifier
appos	Appositional Modifier
nummod	Numeric Modifier
Clausal Relationships	
csubj	clausal subject
ccomp	Casual Component
xcomp	Open Clausal Complement
advcl	Adverbial Clause Modifier
acl	Clausal Noun Modifier
Modifier Word Relationships	
advmod	Adverbial Modifier
discourse	Discourse Element
amod	Adjectival Modifier
Function Word Relationships	
aux	Auxiliary
cop	Copula
mark	Marker
det	Determiner
clf	Classifier
case	Case Marking
Other Notable Relationships	
conj	Conjunct
cc	Coordinating Conjunction
compound	Compound word
punct	Punctuation
root	Root Word

Table 5.1: Common Dependency Relationships defined under UD Annotation scheme (5.2.1).



CONLL-U Format

```
text = The quick brown fox jumps over the lazy dog.
1   The     the     DET     DT   Definite=Def|PronType=Art   4   det     _   _
2   quick   quick   ADJ     JJ   Degree=Pos                  4   amod    _   _
3   brown   brown   ADJ     JJ   Degree=Pos                  4   amod    _   _
4   fox     fox     NOUN    NN   Number=Sing                 5   nsubj   _   _
5   jumps   jump    VERB    VBZ  Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   0   root    _   _
6   over    over    ADP     IN   _                           9   case    _   _
7   the     the     DET     DT   Definite=Def|PronType=Art   9   det     _   _
8   lazy    lazy    ADJ     JJ   Degree=Pos                  9   amod    _   _
9   dog     dog     NOUN    NN   Number=Sing                 5   nmod    _   SpaceAfter=No
10  .       .       PUNCT   .    _                           5   punct   _   _
```

Figure 5.2: Example of a dependency parse tree (top) and its CONLL-U representation (bottom). Tree is generated by CoreNLP Manning et al. (2014) parser

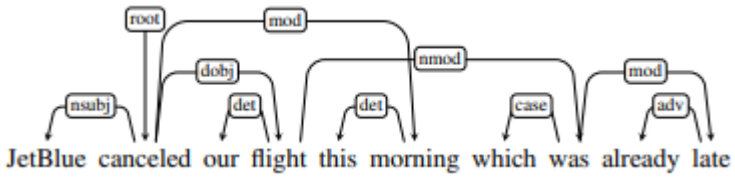


Figure 5.3: Example of Non-projective Parse-tree. Figure from Martin (2021b)

directed dependency links.
More Formally, a dependency parse-tree of an input sentence S is a directed graph $G = (V, E)$ with following constraints:

1. The total Number of nodes $|V|$ is always equal to $|S| + 1$, and the total number of edges $|E|$ is equal to $|S|$. Here $|S|$ is the length of the sentence (usually includes punctuation).

2. The graph should comprise of only one root node.

3. The root node should have no incoming edge.

4. Each node should have exactly one incoming edge (except the root node).

5. There should be a unique path from the root node to every other node in V

6. There should be no cycles in the graph.

These constraints ensure that each word only has a single head, and that the dependency structure is connected.
Figure 5.2 (bottom) depicts the CoNLL-2007 template Nivre et al. (2007) of computationally representing a dependency tree as a sentence. The conllu template stores not only the dependency relationships but other word-level knowledge about the sentence such as POS-tags, XPOS-tags etc.

5.1.2.1 Projectivity

Apart from the constraints listed previously, *Projectivity* is another constrain that is imposed by modern transition-based approaches to the DP task discussed in section 5.1.3.1. A dependency relationship arc is projective, if there is a path from its head-word to every word that lies between the head-word and its dependent-word within the sentence. A dependency tree is said to be *Projective* if all its dependency relationship arcs are Projective. In other words, a dependency tree is Projective if no two of

its arcs cross each other. Figure 5.3 depicts a non-projective dependency tree of an example sentence ***'JetBlue canceled our flight this morning which was already late'*** from Martin (2021b). It is evident in Figure 5.3 that the arc from head-word *canceled* to dependent-word *morning* indicating modifier relationship (mod) is crossing with arc from head-word *flight* to dependent-word *was* indicating noun-modifier relationship (nmod), thus making the entire tree as non-projective.
Most of the dependency trees in English are projective however there are many languages (specifically ones with flexible word-order like Russian), in which non-projective trees are legitimate and common.

5.1.3 Approaches to DP

Conventional statistical approaches to monolingual dependency-parsing task can be categories into two key classes namely the *Transition-based* approaches and the *Graph-based* approaches. Sections 5.1.3.1 and 5.1.3.2 will describe these approaches including the state-of-the-art neural network ones in details. Section 5.1.3.3 will then describe the end-to-end approach to DP on which most of our research work is based.

5.1.3.1 Transition-based approaches

The *Transition based* or *Shift-reduce* parsing approach which was originally developed to analyse the programming languages Alfred and Ullman (1972), has effectively been applied to the dependency-parsing task. The transition based approach to DP Covington (2001) is very similar to the *Transition based* approach to CP discussed in Chapter 3. A typical *Transition based* parser comprises of a **Stack**, a **Buffer** and an **Oracle**.
Table 5.2 depicts the steps involved in *Transition based* parsing of an example sentence. As evident in table 5.2, in the beginning of the transition-based parsing process (at time t = 0), the buffer consists of all the words in the sentence and Stack consists of node ***root***. At each time-step t, the **Oracle** selects the best action to be taken, from all possible actions listed as follows.

1. **LEFT-ARC**: Push top two words from the stack, assign a head-dependent relation from the topmost word in the stack (as head) to the second topmost word in the stack (as dependent), and finally push back the topmost word onto the stack.

2. **RIGHT-ARC**: Push top two words from the stack, assign a head-dependent relation from the second topmost word in the stack (as head) to the topmost

Step	Stack	Buffer	Action	Relations
0	[root]	[the, quick, brown, fox, jumps, over, the, lazy, dog, .]	SHIFT	
1	[root, the, quick]	[brown, fox, jumps, over, the, lazy, dog, .]	SHIFT	
2	[root, the, quick, brown]	[fox, jumps, over, the, lazy, dog, .]	SHIFT	
3	[root, the, quick, brown, fox]	[jumps, over, the, lazy, dog, .]	SHIFT	
4	[root, the, quick, fox]	[jumps, over, the, lazy, dog, .]	LEFT arc	(fox → brown)
5	[root, the, quick, fox]	[jumps, over, the, lazy, dog, .]	LEFT arc	(fox → brown)
6	[root, the, fox]	[jumps, over, the, lazy, dog, .]	LEFT arc	(fox → quick)
7	[root, fox]	[jumps, over, the, lazy, dog, .]	LEFT arc	(fox → the)
8	[root, fox, jumps]	[over, the, lazy, dog, .]	SHIFT	
9	[root, jumps]	[over, the, lazy, dog, .]	LEFT arc	(jumps → fox)
10	[root, jumps, over]	[the, lazy, dog, .]	SHIFT	
11	[root, jumps, over, the]	[lazy, dog, .]	SHIFT	
12	[root, jumps, over, the, lazy]	[dog, .]	SHIFT	
13	[root, jumps, over, the, lazy, dog]	[.]	SHIFT	
14	[root, jumps, over, the, dog]	[.]	LEFT arc	(dog → lazy)
15	[root, jumps, over, dog]	[.]	LEFT arc	(dog → the)
16	[root, jumps, dog]	[.]	LEFT arc	(dog → over)
17	[root, jumps]	[.]	RIGHT arc	(jumps → dog)
18	[root, jumps, .]	[]	SHIFT	
19	[root, jumps, .]	[]	SHIFT	(jumps → .)
20	[root]	[]	DONE	

Table 5.2: Example of Transition based parsing applied to an example sentence

word in the stack (as dependent), and finally push back the second topmost word onto the stack.

3. **SHIFT**: Remove the word from the front of the buffer and push it onto the stack.

Oracles of conventional parsers Yamada and Matsumoto (2003); Nivre et al. (2006a) typically used a machine learning classifier to choose the correct action given the current state of Stack and Buffer as well as action history. Modern transition-based approaches such as Elkaref and Bohnet (2017); Kırnap et al. (2018); Kiperwasser and Goldberg (2016) utilise LSTM or stack-LSTM Dyer et al. (2015) to encode the stack, buffer and action-history. Subsequently they train neural classifier to predict the correct action to be taken at each time-step.

5.1.3.2 Graph-based approaches

Graph-based approaches constitute another distinct class of approaches to the DP task that search through all possible parse-trees for a given sentence to find the maximum scoring tree that satisfies all the constraints listed in section 5.1.2. Hence, given an input sentence S a graph-based parse extracts the dependency tree of it $\hat{T}_s$ by applying the following equation

$$\hat{T}_s = argmax_{T_s} Score(T_s)$$

The Score(T_s) of any sentence parse-tree T (of sentence S) is simply computed as the sum of scores all its edges as equation

$$Score(T) = \Sigma_{e \in E_T} score(e)$$

Here E_T is set of all edges of the tree T and $score(e)$ is the edge of any tree-edge e. Figure 5.4 depicts the process of graph-based parsing. As evident in Figure 5.4 a typical graph-based dependency parser implements following steps:

1. It builds a fully connected directed graph (with an outgoing edge from each node to every other node) with each word in the input sentence as a node as well as an extra root node.

2. Assign an edge score to each edge of the fully connected graph. This edge-score of each edge is predicted by a ML/DL based algorithm, which makes prediction based on various features of head (outgoing) and dependent (incoming) node of the direct edge.

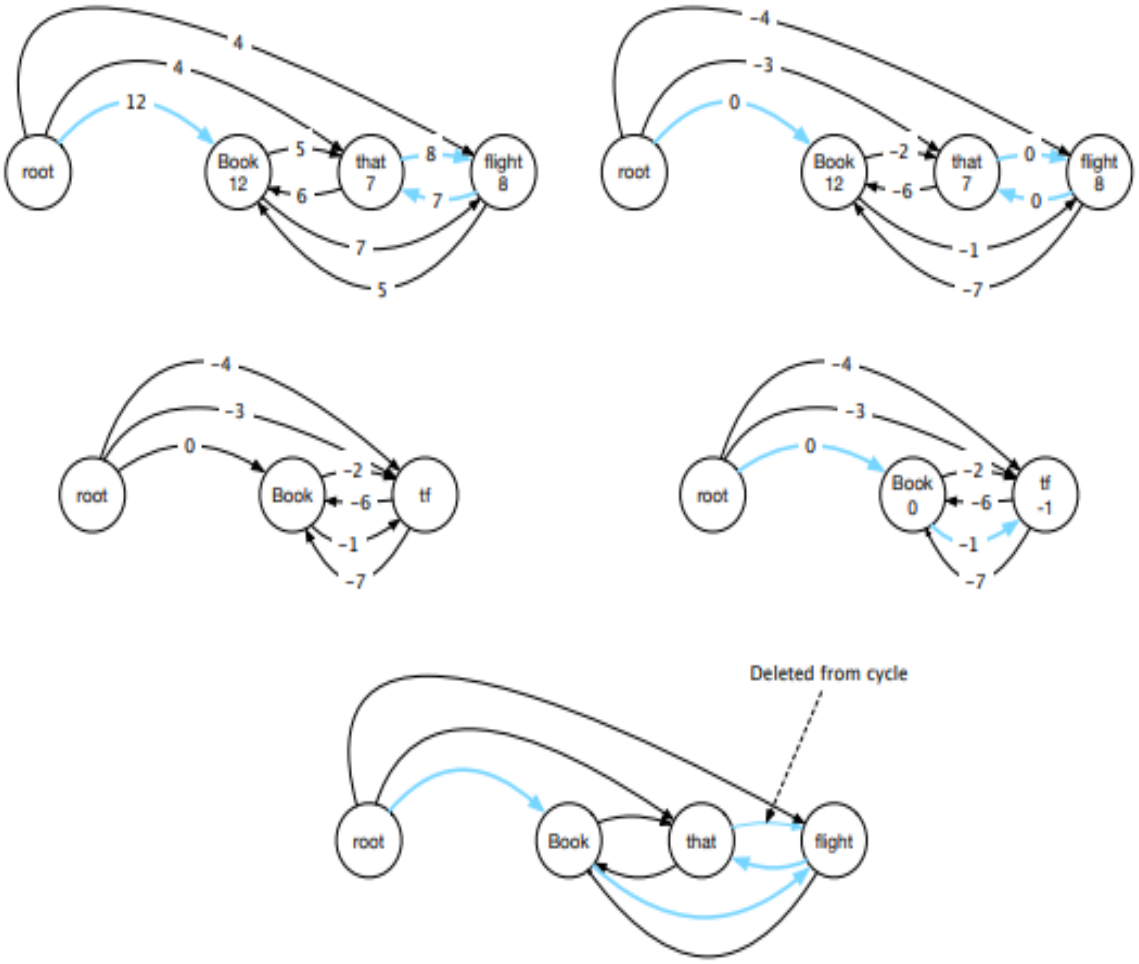


Figure 5.4: Graph-based dependency parsing algorithm Chu (1965) applied to an example sentence ***Book that flight***. Figure from Martin (2021b)

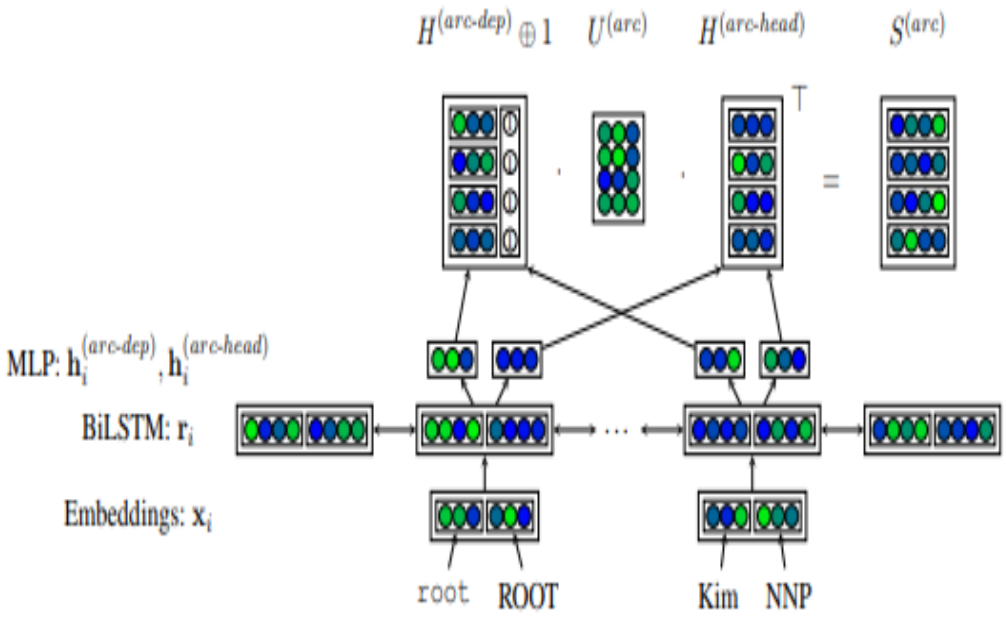


Figure 5.5: Deep Biaffine Network architecture proposed by Dozat and Manning (2016)

3. Finally, the maximum spanning tree is extracted from this fully connected graph using standard graph-theory algorithms Greenberg (1998).

The aim of training a graph-based DP model is therefore to enable the ML/DL based scorer to assign edge-scores to all edges of the connected graph in such a way that its maximum spanning tree is always the correct dependency parse-tree of the respective sentences.

Conventional ML based approaches to graph-based DP involves computes the edge score of an edge as following equation:

$$score(e) = \Sigma_{i=1}^{N} w_i * f_i$$

Here, f_i is the feature-value of any I^{th} and w_i is the weight to be learnt through training. The features used by various approach in edge-weight scorer include head and dependent word-forms, lemmas, POS-tags, contexts, distance between head and dependent etc.

The model is trained by minimizing the Loss L given by equation 5.1.

$$L = Score(\hat{T}_s) – Score(T*_s) \tag{5.1}$$

Here $\hat{T}_s$ is the True dependency-tree and $T*_s$ is the maximum scoring dependency-tree out of all possible dependency-trees for sentence S excluding the true one.

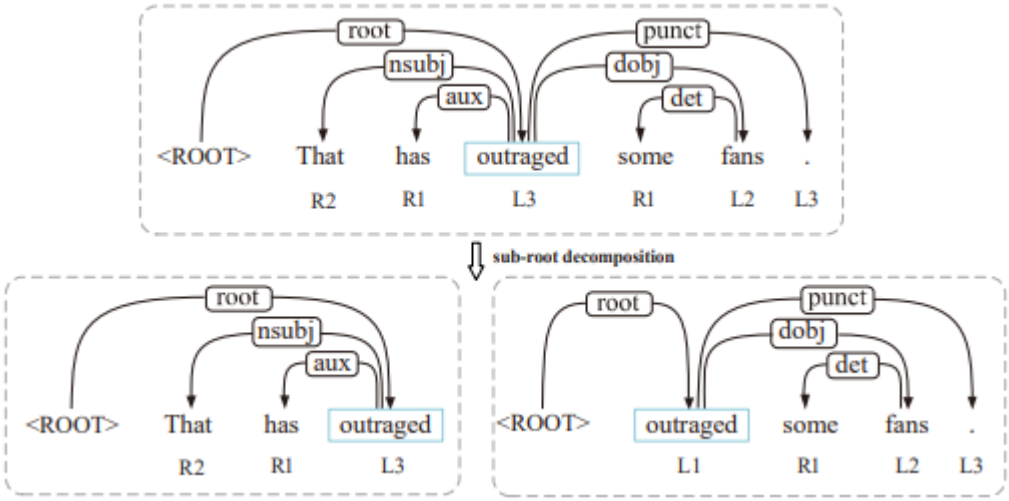


Figure 5.6: Sub root decomposition as performed by Li et al. (2018)

Modern state-of-the-art NN based approaches to graph-based DP is very similar to conventional one, with the exception that these approaches used a neural-network to learn the features for the edge-scorer rather than manually encoding these features. First NN based approach proposed by Kiperwasser and Goldberg (2016) used Bidirectional LSTM model to encode the nodes of an edge as feature-vector representations. Final layer of the network is a feed-forward layer which takes as input these node-embeddings and output the edge-score. The *Deep Biaffine Network* proposed by Dozat and Manning (2016) is the state of the art graph-based approach to DP task. It is similar to the work of Kiperwasser and Goldberg (2016) with a biaffine scorer that assigns edge scores to all edges of the fully connected graph simultaneously. Figure 5.5 depicts the architecture of the Deep Biaffine network.

5.1.3.3 End-to-end Approaches

Li et al. (2018) proposed an end-to-end framework for DP task, which is distinct from both Graph-based and Transition-based approaches. The authors of Li et al. (2018) represented the entire dependency parse-tree as relative head-position tag sequence as shown in Figure 5.7a. Hence the goal of the model is now to predict the correct relative head-position tag-sequence $T = t_1, t_2, \ldots, t_N$ given the input sentence as a sequence of words $W = w_1, w_2, w_3 \ldots . w_N$.

Hence the DP task is now reduced to the standard sequence-tagging task (such as POS-tagging, Named Entity Recognition etc.). However, there is one slight difference than unlike in standard sequence tagging task, while predicting the head-position tag sequence, we must ensure the validity of the underlying depndency-parse tree. This

is done by imposing various heuristic constraints while predicting the head-position tag sequence for an input word-sequence.
Li et al. (2018) proposed the BiLSTM-CRF with attention model Niu et al. (2021) to perform this head-position tag sequence prediction task. Figure 5.8a depicts the architecture of this BiLSTM-CRF model. The authors claimed that this proposed model performs at par with the state-of-the-art graph-based and transition-based parsers despite being much simpler in design and therefore much easy to implement, train and deploy.
It was observed that the proposed end-to-end dependency parser is not good at capturing long distance dependencies though thus under-performing on comparatively longer sentences. Hence authors performed sub-root decomposition on all the sentences which are longer than the set threshold length. Figure 5.6 depicts the sub-root decomposition of an example sentence. We indeed utilised the same sub-root decomposition technique while training our proposed parsers as described in the subsequent sections, to address the issue of long-distance dependencies.

5.2 Low-resource Dependency Parsing

Section 5.1.3 described the state-of-the-approaches to dependency parsing. However, all three categories of DP approaches (namely, *Transition-based*, *Graph-based* and *End-to-end* approaches) are supervised approaches that require significant training datasets. Such datasets may not be sufficiently available for many low resource languages. This inspired a new line of research to build models for DP in low-resource languages. In this line of research, cross-lingual approaches discussed in section 2.1.1 (both *Data-transfer* and *Model-transfer* approaches) have been utilised very effectively. Section 5.2.1 provides a brief introduction to the Universal Dependencies project which is the key reason behind the success of cross-lingual approaches to DP task. Subsequently, section 5.2.2 will provide a brief literature review of modern approaches to cross-lingual DP. In this section, we also list numerous CL based approaches to DP that used linguistic typology knowledge to improve the cross-lingual transferring ability of the respective models.

5.2.1 Universal Dependency

Universal Dependencies (UD) project De Marneffe et al. (2021) is aimed at developing a set of dependency treebank annotations which are consistent across most of the world's languages. This annotation scheme is developed based on the evolution of

universal Stanford dependencies De Marneffe et al. (2006); De Marneffe and Manning (2008); De Marneffe et al. (2014), Google universal part-of-speech tags Petrov et al. (2011), and the morphosyntactic interlingua tagsets developed by Zeman (2008). The main philosophy behind developing such UD project is to provide universal categories and guidelines to promote consistent annotation (of similar constructions) across all languages, while also allowing necessary language-specific extensions. Table 5.1 lists common dependency labels adopted under the UD annotation scheme.
Hence, unlike in the case of CP where the various treebanks in multiple languages developed over the years have their own unique tag annotation, in DP various treebanks in multiple languages were build around a shared Universal annotation tagset. This makes the cross-lingual transfer simpler. In fact, UD project also included making various train, test and dev treebanks in many languages available online and open-source[1] for the researchers to build and test cross-lingual DP models. UD is indeed an ongoing crowd-sourced project where new treebanks are constantly added by the linguists and researchers. Latest UD version (UD v2.10) comprises of 228 treebanks in 130 languages.

5.2.2 Cross-lingual Approaches to Dependency-parsing

Numerous cross-lingual transfer-learning based approaches to Dependency parsing for low-resource languages have been proposed. These include both *Annotation-Projection approaches* such as Smith and Eisner (2009); Huang et al. (2009); Chen et al. (2011); Jiang and Liu (2010); Li et al. (2014); Xiao and Guo (2015) as well as *Model-transfer approaches* such as McDonald et al. (2011); Cohen et al. (2011); Duong et al. (2015a); Guo et al. (2016b); Vilares et al. (2015); Falenska and Çetinoğlu (2017); Mulcaire et al. (2019); Vania et al. (2019); Shareghi et al. (2019) which involve training a model on high-resource languages and subsequently adapting it to low-resource target languages.
Apart from bilingual model-transfer approaches, numerous multilingual parsers have been proposed such as Stanza Qi et al. (2020), UDpipe Future Straka et al. (2019) and UDify Kondratyuk and Straka (2019b). These parseres are trained on joint polyglot corpora. Results in these papers show that multilingual polyglot-training improves the performance of a model on most low-resource target-languages, as compared to simple monolingual training for cross-lingual model-transfer.
Participants of CoNLL 2017 shared-task Daniel et al. (2017) and CoNLL 2018 shared

[1] `https://universaldependencies.org/`

task Zeman et al. (2018) also provide numerous approaches to dependency parsing of low-resource languages.
Numerous approaches such as Naseem et al. (2012); Täckström et al. (2013); Barzilay and Zhang (2015); Wang and Eisner (2016); Rasooli and Collins (2017); Ammar (2016) also used typological information to facilitate cross-lingual transfer. Most of these approaches utilise only selected word-order typology features from WALS database Haspelmath (2009). Further, they feed this linguistic typology features into the model along with word/token representations.

5.3 Research Objective

Our research-work described in this chapter, is aimed at utilising linguistic typology knowledge to improve the performance of two state-of-the-art end-to-end cross-lingual/multilingual dependency parsers. Linguistic typology (specifically word-order typology knowledge) has successfully been used by various researchers (chapter 2) to improve the cross-lingual transferring ability of the respective models from high-resource source languages to low-resource target languages. Section 5.2.2 describe these approaches in detail. However, all these approaches directly feed-in the linguistic typology features into the respective model along with word-representations. On the other hand, we induced the linguistic typology knowledge into both of our models through Multitask learning (MTL) instead, by adding an auxiliary task of linguistic typology prediction along with DP. We injected the typology knowledge available in ***URIEL database*** described in section 5.3.2. Section 5.3.1 provides a detailed overview on MTL. Inducing typology knowledge through MTL rather than directly feeding it along with word-embeddings have following advantages.

1. The model can also be applied to low-resource languages for which many typology feature values are unknown/missing.

2. The auxiliary task should help to improve the performance on the main dependency parsing task as well, since it would make the model give special emphasis on the syntactic typology (specially word-order typology) of language being parsed while predicting the dependency relations.

Hence, our entire research-work related to cross-lingual DP with typology can be divided into two parts. Section 5.4 will describes the first part of work, in which we make following contributions.

1. We evaluated the performance an *End-to-end BERT Based Parser* which can parse a sentence by directly predicting relative head-position tag for each word in the input sentence. This is inspired by Li et al. (2018) which is an *End-to-end Seq2seq Dependency Parser*. We evaluated the performance of this BERT based end-to-end parser in both mono-lingual and cross-lingual/multilingual setups (using mBERT). We will refer to this model as ***Base E2E BERT parser***.

2. We added the auxiliary task of Linguistic typology prediction to our Base E2E BERT parser to observe the change in performance under different settings. We will refer to this model as ***Multitasking E2E BERT Parser*** in this paper.

3. We evaluated the change in performance of various mBERT based Cross-lingual Dependency Parsing models due to polygot training.

In the second part of our research-work related to cross-lingual DP with typology knowledge we make following contributions. Section 5.5 describes this part of work in detail.

1. We re-implemented the UDify model which is the state-of-the-art language-agnostic dependency parser which is trained on a polyglot corpus of 75 languages. Subsequently, we added the auxiliary task of typology prediction to it and evaluated the increase/decrease in the performance as a result.

2. We evaluated the impact of various category of typology fetures on the performance of UDify in multilingual settings.

5.3.1 Multitask Learning

Multi-task Learning (MTL) Ruder (2017) is a neural network framework which involves performing two or more tasks simultaneously leading to knowledge/parameter sharing. These tasks are closely related thus complement each other leading to improved performance on all of them.
Even in scenarios where we primarily care about a single task, using a closely related task as an auxiliary task for MTL can be useful. For example, Caruana (1998) used tasks that predict different characteristics of the road as an auxiliary tasks while predicting the steering direction in a self-driving car. Zhang et al. (2014) used head pose estimation and facial attribute inference as auxiliary tasks for facial landmark detection, Liu et al. (2015), jointly learn query classification and information retrieval, Girshick (2015) jointly predicted the class and the coordinates of an object in an

Feature-type	**Vectors**
Syntactic	syntax_wals, syntax_sswl, syntax_ethnologue, syntax_knn
Phonology	phonology_wals, phonology_ethnologue, phonology_knn, phonology_average
Inventory	inventory_ethnologue, inventory_phoible_saphon, inventory_phoible_spa, inventory_phoible_ph, inventory_phoible_ra, inventory_phoible_upsid, inventory_knn, inventory_average
Family	fam
Geography	geo
One-hot	id

Table 5.3: Various typology vector representations of a language, provided by *lang2vec* library.

Language	**Value**	**Binary Representation**					
		S_SVO	S_SOV	S_VSO	S_VOS	S_OVS	S_OSV
en	SVO	1.0	0.0	0.0	0.0	0.0	0.0
ga	VSO	0.0	0.0	1.0	0.0	0.0	0.0
hi	SOV	0.0	1.0	0.0	0.0	0.0	0.0
mg	VOS	0.0	0.0	0.0	1.0	0.0	0.0

Table 5.4: Binary representations of WALS feature **Order of *Subject-Verb-Object*** in URIEL Database Littell et al. (2017)

image, Arik et al. (2017) jointly predicted phoneme duration and frequency profile for text-to-speech. In this work, we use linguistic typology feature prediction task as auxiliary task for cross-lingual DP

5.3.2 URIEL Database

As explained in section 5.3, for Linguistic typology feature prediction auxiliary tasks we used Linguistic typology feature values provided by URIEL database Littell et al. (2017). The URIEL database is a collection of binary features extracted from multiple typological, phylogenetic, and geographical databases such as WALS Haspelmath (2009), PHOIBLE Moran et al. (2014), Ethnologue M. Paul Lewis and Fennig (2015) and Glottolog Hammarström et al. (2017).
Let a typology-feature f has a set of values as V within its original database. Then feature f can be converted to $|V|$ features such that for every value $v \in V$ the corresponding binary feature f_v is computed as equation 5.2.

Lang-code	Families				
	Indo-European	Germanic	West-Germanic	Romance	North-Germanic
de	1.0	1.0	1.0	0.0	0.0
en	1.0	1.0	1.0	0.0	0.0
fr	1.0	0.0	0.0	1.0	0.0
sw	1.0	1.0	0.0	0.0	1.0
mg	0.0	0.0	0.0	0.0	0.0

Table 5.5: Representation of genealogical properties of example languages in the URIEL database Littell et al. (2017)

$$\begin{aligned} f_v &= 1 \quad if \ f == v \\ &= 0 \quad otherwise \end{aligned} \tag{5.2}$$

Table 5.4 depicts the process of binarization of a prominent WALS feature ***Subject-Verb-Object***. In similar fashion, the authors of Littell et al. (2017) binarized all typology features of all databases listed previously, thereby creating many comprehensive binary vector representations of each language.

The authors also binarized and encoded the genealogical properties of all the languages as shown in table 5.5. Finally, the authors also encoded the geographical representation of each language as a unique vector of fixed dimension. Each feature in the geography vector of a language comprises of the orthodromic distance—from the specific language to a fixed point on the Earth. These distances are expressed as a fraction of the Earth's antipodal distance. Thus the value would be 0.0 and 1.0.

Hence the URIEL database provides numerous typology vectors listed in table 5.3. All these vectors can be accessed through the Pyton PyPi library called *lang2vec*[2]. For the experiments within this paper, we used only syntactic binary features generated from WALS database (categorised as *Syntax-WALS* within URIEL database).

5.4 Multitasking End-to-end BERT based Cross-lingual Dependency Parser

In this section, we describe our research-work which involved the evaluation of an end-to-end BERT based multilingual dependency parser which is inspired by the E2E

[2]https://pypi.org/project/lang2vec/

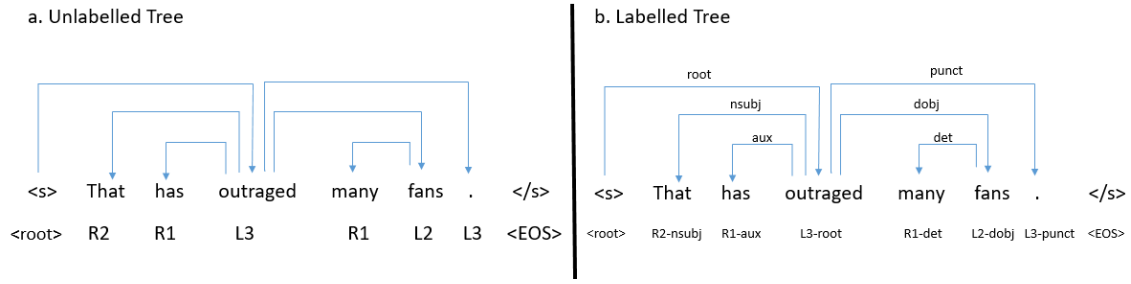


Figure 5.7: Examples of dependency parse tree being represented as relative head-position tag sequence

Seq2seq DP model proposed by Li et al. (2018). Subsequently, we added the auxiliary task of prediction of typology feature-values of the URIEL database. Section 5.4.1 describes our proposed Base End-to-end BERT model. In section 5.4.2 we describe the modification of the base model architecture to inject typology knowledge. Section 5.4.3 describes the training process, section 5.4.4 describes experimental details and section 5.4.5 outlines results achieved.

5.4.1 Base End-to-end BERT Parser

This section elaborates the details of our *End2End BERT based Dependency Parser* which directly predicts the relative head position tag of each word within input sentence.

Given a sentence of length T, its dependency parse-tree can be represented as a sequence of T relative head-position tags as demonstrated in Figure 5.7a.

Figure 5.8a depicts the architecture of our baseline model. The depicted architecture comprises of three components namely *BERT Encoder*, *Output Network* and *Tree-decoder* described as section 5.4.1.1, 5.4.1.2 and 5.4.1.3.

5.4.1.1 BERT Encoder

It is a BERT based network which takes as input, the entire sentence as sequence of tokens. The model outputs $d-1$ dimensional word-embeddings for all words within the input sentence. Thus for a sentence of length T, it would output matrix $E \in R^{T*(d-1)}$.

We used the WordPiece tokenizer Wu et al. (2016) to tokenize input sentence and extract embeddings. For each word within input sentence, we use the BERT output corresponding to the first wordpiece of it as its embedding, ignoring the rest.

We also add pos-tag information in our parser by appending index of pos-tag of each

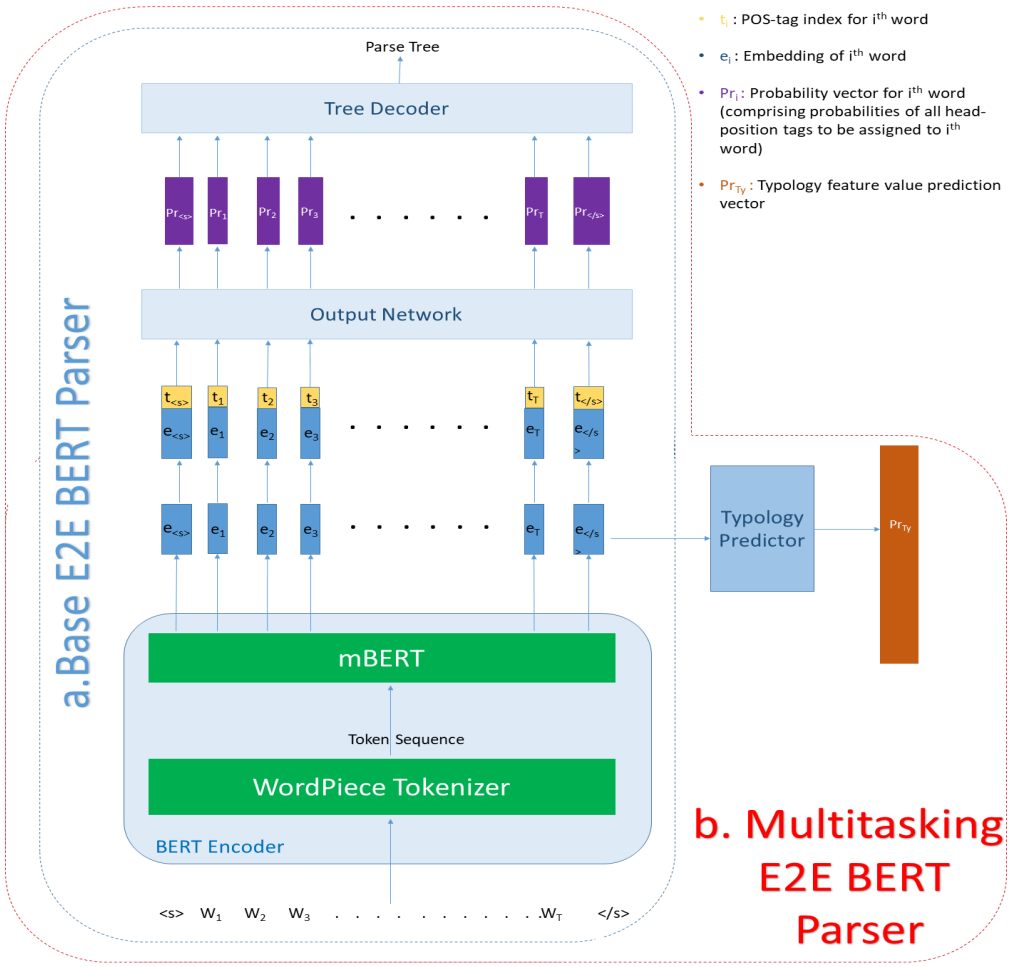


Figure 5.8: a. Base End-to-end BERT parser architecture. b.Multitasking End-to-end BERT parser architecture. Its an extension of Base End-to-end BERT parser architecture with one extra component namely *Typology Predictor*.

word, to the encodings outputted by BERT encoder as evident in Figure 5.8a. Thus the final embedding-matrix $\hat{E}$ is derived from E as:

$$\hat{E} = E; [t_1; t_2; ....; t_T]$$

Here t_i is POS-tag index of i^{th} word. $\hat{E} \in R^{T*d}$

5.4.1.2 Output Network

Its a simple feed-forward network with the *softmax* activation function. The network takes in an embedding matrix $\hat{E} \in R^{T*d}$ from the BERT encoder and outputs the probabilities of all possible relative head position tags at each word by applying the following equation.

$$Pr = softmax(\hat{E} * W + b)$$

Here W, b are weights and biases, $Pr \in R^{T*N}$ where N is the number of valid relative head-position tags.
For the sentence of length T, the set of all possible relative head position tags S_T is given as

$$S_T = [L_1, L_2, ..., L_T, R_1, R_2, ..., R_{T-1}, < root >, < EOS >]$$

Here $< root >$ and $< EOS >$ are tags to be assigned to $< s >$ and $< /s >$ tokens at the begin and end of the input sentence as shown in Figure 5.7a.
For training and evaluations, we always computed probabilities of all relative head-position tags within the tag-set for a sentence of length Max i.e. S_{Max} as the dimensions of model parameters should be fixed. Here Max is the length of largest sentence from all corpora used during the experiments.
In this paper we experimented with only Unlabeled Dependency Parsing however same architecture can be used for Labeled Dependency Parsing as well. In such case the output tags would comprise of relative head positions as well as relationship labels (eg: L2-nsubj). Hence, the set of all possible relative head position tags S would be much larger. Figure 5.7b depicts a labelled parse-tree being represented as sequence of head-position tags.

5.4.1.3 Tree-Decoder

This component decodes the most probable correct label sequence from the probabilities outputted by Output Network. The correct label sequence would satisfy following constraints.

1. The sequence should start with $< root >$ and end with $< EOS >$ tags. These tags should not appear anywhere else.

2. At each index (of word being labelled) the assigned label should be within the range of sentence. For example: the word ***That*** within sentence shown in Figure 5.7a can not have tags L_2, L_3, L_4, L_5, L_6 and the punctuation **.** in the sentence can not have any right tags as these are outside the range of sentence.

3. The label sequence should not generate any cycles within the dependency tree.

4. One of the words should have the head at $< root >$ token.

We used dynamic programming with beam-search to efficiently extract the most probable label sequence which satisfies the above listed constraints, out of all possible label sequences.

5.4.2 Multitasking End-to-end BERT Parser

Figure 5.8b demonstrates the architecture of our proposed model. The model is very similar to the *Base E2E BERT Parser* described in section 5.4.1 with one extra component namely *Linguistic typology predictor* which predicts the typology features of the language being parsed. Thus model is Multi-tasking model with hard-parameter sharing Ruder (2017).

5.4.2.1 Linguistic typology predictor

It is a simple deep feed forward neural network which takes in the embedding generated by BERT Encoder for token $< /s >$ as input and outputs probabilities of values of binary syntactic typology features for the language being parsed as 1. Such features are provided by URIEL database (section 5.3.2).
Let $\hat{N}$ be the number of syntactic typology features provided by URIEL database. The *Linguistic typology predictor* would then predict probability matrix $Pr_{ty} \in R^{\hat{N}}$ by applying equation 5.3.

$$Pr_{ty} = sigmoid(e_{</s>} * U + c) \tag{5.3}$$

Here $e_{</s>} \in R^d$ is embedding from the BERT Encoder for $< /s >$ token. $U \in R^{d*\hat{N}}$ and $c \in R^{\hat{N}}$ are weights and biases respectively.

Experimental Settings	**Source Languages**	**Target Languages**
Monolingual	en, zh	en, zh
Cross-lingual with single source language	en	de, hr, it, hi, zh, et, vi
Cross-lingual with multiple source languages	en, ur, fr, ar, ja, pl, la, ta, el, cop, kk, tr	de, hr, it, hi, zh, et, vi

Table 5.6: Source and Target Languages used during experiments

5.4.2.2 Missing Typology

As with most typology databases, URIEL also comprises several missing values of various typology-features for many languages. These missing values are indicated as '–' in typology vector provided by URIEL (rather than having values 0 or 1). A typology feature can also have value as '–' for a well-documented language if that feature has no dominant value observed within the respective language
These missing features pose a problem during training of *Multitasking BERT Parser*.

Hyper-parameter	Value
d	768
Dropout prob.	0.01
Bach-size	32
Number of steps per epoch	Size of training corpus / 32
Epochs	50
BERT dimensions	cased_L-12_H-768_A-12

Table 5.7: Hyper-parameters

We address this issue through the masking technique Vaswani et al. (2017). We mask the missing typology features and train only on available ones for each source language.

5.4.3 Training

We trained both *BERT Encoder* (fine-tuning of pre-trained BERT model) and *Output Network* components of *Base E2E BERT Parser* model jointly, by optimizing the cross-entropy loss Gómez (2018) between true relative head-position tags and probabilities outputted by the *Output Network*.
On the other hand, *Multitasking E2E BERT parser* is trained to perform tasks of *Prediction of relative heap-position tag sequence* and *Prediction of typology features* simultaneously through MTL, by optimizing the total-loss as the sum of cross-entropy loss over true head-position tag-sequence and the binary cross-entropy loss over true typology values.
Table 5.7 outlines values of hyper-parameters used during experimentation. These values are obtained by minimizing loss on *Validation* dataset for English language.

5.4.4 Experiments

In this section the monolingual and multilingual variants of our proposed models within two distinct experimental setups namely *Monolingual* and *Cross-lingual* setups. These are described as sections 5.4.4.1 and 5.4.4.2 respectively.
These experiments aim to achieve following novel objectives:

1. To evaluate the performance of our end-to-end BERT based model for dependency parsing task in both monolingual and cross-lingual settings, and compare it with the performances of state-of-the-art cross-lingual and monolingual models. Such evaluation is necessary as the end-to-end model is much simpler in design and therefore highly time and space efficient.

Languages	Corpus
en	en_ewt-ud-train
ur	ur_udtb-ud-train
fr	fr_ftb-ud-train
ar	ar_padt-ud-train
ja	ja_gsd-ud-train
pl	pl_pdb-ud-train
la	la_ittb-ud-train
ta	ta_ttb-ud-train
el	el_gdt-ud-train
cop	cop_scriptorium-ud-train
kk	kk_ktb-ud-train
tr	tr_imst-ud-train

Table 5.8: Corpora for source languages listed in Table 5.6 used during experiments. All Corpora are part of Universal Dependencies dataset.

Languages	Corpus	Dev Corpus*
de	de_hdt-ud-test	de_hdt-ud-dev
hr	hr_set-ud-test	hr_set-ud-dev
it	it_isdt-ud-test	it_isdt-ud-dev
hi	hi_hdtb-ud-test	hi_hdtb-ud-dev
zh	zh_gsd-ud-test	zh_gsd-ud-dev
et	et_edt-ud-test	et_edt-ud-dev
vi	vi_vtb-ud-test	vi_vtb-ud-dev

Table 5.9: Corpora for target languages listed in Table 5.6 used during experiments. All Corpora are part of the Universal Dependencies dataset. * A small subset of sentences are sampled from these Corpora to be added to the source Corpora in the *Few-shot* scenarios

2. To evaluate the impact of injection of linguistic typology knowledge into our proposed end-to-end parser, through multitasking.

3. To evaluate the impact of polyglot learning and few-shot learning on the performances of both base end-to-end BERT parser as well as multitasking end-to-end BERT parser.

We conducted the experiments on numerous source-target language pairs. Table 5.6 lists the languages on which experiments were conducted in both *Monolingual* and *Cross-lingual* setups.

5.4.4.1 Monolingual Setup

In this setup we conducted experiments to evaluate the performance of fully monolingual variants of our proposed *Base E2E BERT Parses* and *Multitasking E2E BERT Parser*. In these settings we experimented in two languages namely *English* and *Chinese*. These monolingual variants use pre-trained monolingual English and Chinese BERT models provided by Huggingface open-source library[3].
For all experiments within this setup, we used the *Deep Biaffine Parser* Dozat and Manning (2016) as the baseline. Its is a neural graph-based dependency parser which uses biaffine attention classifiers to predict the arcs and labels of the required parse-tree for an input sentence.

5.4.4.2 Cross-lingual setups

We conducted numerous experiments to evaluate the performance of Multilingual/Cross-lingual variants of our proposed *Base BERT Parses* and *Multitasking E2E BERT Parser* models in cross-lingual settings. These Multilingual variants use the pre-trained Multilingual BERT (mBERT) Wu and Dredze (2019) model which is trained on data from Wikipedia in 104 languages.
We evaluated the Multilingual variants of our models under following two Cross-lingual setups.

1. *Cross-lingual with single source language (CL-Single)*: In this setup, all the parsers are trained in single source language English, but tested on a diverse range of target languages

2. *Cross-lingual with multiple source languages (CL-Poly)*: In this setup, all the parsers are trained on diverse polygot corpus and tested on a diverse range of target languages. There is no overlap between source and target language sets.

Furthermore, the experiments within the *Cross-lingual with single source language (CL-Single)* and *Cross-lingual with multiple source languages (CL-Poly)* setups are conducted under both *Few-shot* and *Zero-shot* learning scenarios.
Within the *Zero-shot* learning scenario, the training corpus does not contain any sentence in the target language on which the model is being evaluated. On the other hand, within the *Few-shot* learning scenario, the training corpus consists of few sentences in the target language on which the model is being evaluated, along with other source language sentences (covering over 80% the corpus).

[3] `https://huggingface.co/models`

Model	en	zh
Deep Biaffine Network	**93.77**	**78.67**
Base E2E BERT Parser	93.00	76.87
Multitasking E2E BERT parser	93.13	78.17

Table 5.10: Unlabeled Attachment Scores (UAS) achieved in *Monolingual* experimental settings.

In Cross-lingual setups we used Graph-based mBERT parser by Wu and Dredze (2019) as baseline. It is a multilingual parser that uses same architecture as Dozat and Manning (2016) except the LSTM encoder which is replaced by mBERT.

5.4.4.3 Languages

Table 5.6 lists various source and target language used in each of the experimental settings. In *CL-Poly* setup, we trained our models on joint polygot corpus of all twelve source languages listed in Table 5.8. All these twelve languages belong to distinct linguistic families thus making the corpus typologicaly diverse.
For all experiments, the training corpus size is always fixed to 30,000 sentences. The joint polygot corpus to train *CL-Poly* is created by randomly sampling 2500 sentences from the training corpus for each of the 12 source languages listed in Table 5.8, concatenating them as one treebank and randomly shuffling the order.
Our *Cross-lingual* models are tested on seven target languages, belonging to distinct linguistic families. Three of these seven languages namely *zh*, *et* and *am* belong to a linguistic family which is distinct from language families of all the source languages listed in Table 5.8. Thus performance on these languages indicate true robustness of the evaluated models to typological variations between source and target languages.
For each experiment under the *Few-shot learning* scenario, we extracted a small set of target language sentences (on which model is being evaluated), to be added to the source training corpus before training.
We extracted this subset by randomly sampling sentences from the *dev* corpus of the respective target-language tree-bank dataset until the token-size becomes approximately equal to 3000. This is inspired by Ammar et al. (2016) who used same yardstick to evaluate their *Multi-lingual Dependency Parser (MALOPA).*

5.4.5 Results

Tables 5.11 and 5.12 outline Unlabeled Attachment Scores (UAS) obtained under the *Few-shot* and the *Zero-shot* learning scenarios respectively. The tables 5.10, 5.11

	CL-Single				CL-Poly			
	mBERT	**Base E2E**	**Multi E2E**	**Aux task***	**mBERT**	**Base E2E**	**Multi E2E**	**Aux task***
zh	**43.32**	42.98	41.74	0.01	**66.81**	66.52	65.35	0.28
hr	**72.49**	72.07	70.91	0.07	**75.28**	75.01	74.05	0.14
et	**71.05**	70.69	69.72	0.05	**67.2**	66.8	65.67	0.26
de	**78.07**	77.68	76.67	0.04	**78.85**	78.54	77.33	0.21
hi	**44.83**	44.42	43.18	0.11	**74.68**	74.4	73.32	0.22
it	**86.63**	86.32	85.23	0.04	**77.77**	77.4	76.3	0.21
vi	**40.74**	40.34	39.25	0.08	**66.89**	66.56	65.45	0.24

Table 5.11: Unlabeled Attachment Scores (UAS) achieved in both Cross-lingual settings under the *Zero-shot* scenario. *F1 values achieved on the auxiliary task of linguistic typology prediction (excluding missing values)

	CL-Single				CL-Poly			
	mBERT	**Base E2E**	**Multi E2E**	**Aux task***	**mBERT**	**Base E2E**	**Multi E2E**	**Aux task***
zh	44.04	43.69	**44.29**	0.57	67.68	67.37	**68.19**	0.76
hr	73.38	73.0	**73.46**	0.6	75.93	75.58	**76.28**	0.68
et	71.89	71.5	**71.96**	0.56	67.91	67.55	**68.45**	0.78
de	78.8	78.47	**79.08**	0.57	79.74	79.45	**80.25**	0.71
hi	45.63	45.33	**45.91**	0.61	75.59	75.16	**76.13**	0.62
it	87.44	87.12	**87.63**	0.61	78.51	78.14	**78.98**	0.66
vi	41.44	41.16	**41.62**	0.61	67.68	67.41	**68.37**	0.75

Table 5.12: Unlabeled Attachment Scores (UAS) achieved in both Cross-lingual settings under the *Few-shot* scenario. *F1 values achieved on the auxiliary task of linguistic typology prediction (excluding missing values)

and 5.12 also outline the F1-scores achieved by our *Multitasking E2E BERT parser* model on the auxiliary task of predicting linguistic-typology features in Monolingual settings as well as both *Cross-lingual with single source language* and *Cross-lingual with multiple source languages* under both *Zero-shot* and *Few-shot* scenarios. The results in these tables indicate the impact of the auxiliary task.

5.4.6 Analysis

In this section we analyse the results outlined in section 5.4.5 to address the research questions **RQ4**, **RQ5**, **RQ6** listed in section 1.1.1 as follows.

RQ4: Does an End-to-end Dependency parser performs at par with the state-of-the-art Graph-based parser, within both monolingual and cross-

lingual settings ?

Results outlined in section 5.4.5 indicate that in both *Monolingual* and *Cross-lingual settings*, our *Base E2E BERT parser* indeed performed at par with the baseline *Deep Biaffine Parser* Dozat and Manning (2016) and *Graph-based mBERT parser* Wu and Dredze (2019) models respectively, despite being much simpler in design as its end-to-end.

RQ5: Does injecting linguistic typology knowledge into an End-to-end cross-lingual dependency parser, through an auxiliary task of typology feature-value prediction, leads to improvement in performance of it ?

Results outlined in Table 5.10 show that within Monolingual setup, our *Multitasking E2E BERT parser* showed marginal improvement over *Base E2E BERT parser* for both English and Chinese. In fact the monolingual variant of our *Multitasking E2E BERT parser* outperformed the baseline *Deep Biaffine Parser* Dozat and Manning (2016) for both English and Chinese.
Hence it can be inferred that in Monolingual settings, the auxiliary task of predicting linguistic typology features does lead to improvement in parsing performance indeed, as it enables the model to emphasize on syntactic typology of language being parsed (specifically word-order features) while predicting the dependency relations within the sentence.
The results in Tables 5.11 and 5.12 indicate that under the *Cross-lingual Zero-shot learning* scenario our proposed *Multitasking E2E BERT parser* under-performed the *Base E2E BERT parser* in both *CL-Single* and *CL-Poly* settings for all the target languages, whereas it outperformed the *Base E2E BERT parser* in both *CL-Single* and *CL-Poly* settings under the *Cross-lingual Zero-shot learning* scenario.
Furthermore, it can also be observed in tables 5.11 and 5.12 that within the *Zero-shot* scenario, our *Multitasking E2E BERT Parser* performed poorly on the auxiliary task with average F1 score being 0.06 within *CL-Single* and 0.22 with *CL-Poly* settings respectively. On the other hand, within the *Few-shot training* scenario, the proposed *Multitasking E2E BERT Parser* showed comparatively better performance on the auxiliary task with average F1 score being 0.59 within *CL-Single* and 0.71 with *CL-Poly* settings respectively.
Based on these trends it can be inferred that the auxiliary task does not help the model to improve the cross-lingual transfer parsing in an unseen language (which are

not the part of training corpus). However the task does enable the model to better learn to distinctively parse in each of the languages on which it is trained, even if the training corpus consists of only few sentence in the language.

RQ6 Is the impact of adding the auxiliary task of typology feature-value prediction higher with mixed polyglot training scenerio, as compared to single source language training scenerio ?

In the *CL-Poly* setting under the *Few-shot learning* scenario, our *Multitasking E2E BERT parser* shows an average improvement of 4.6% in UAS across all target languages over the *Base E2E BERT parser*. This is much higher than the average improvement of 1.93% shown by our*Multitasking E2E BERT parser* over *Base E2E BERT parser* within *CL-Single* settings under the *Few-shot learning* scenario.
Furthermore, it is also observed that in both *Few-shot* and *Zero-shot* scenarios, our proposed *Multitasking E2E BERT Parser* performed better on the auxiliary task of linguistic typology prediction, within mixed polyglot training (CL-Poly) settings as compared to under monolingual training settings (CL-Single).
Hence, for the cross-lingual parsing task, the improvement in performance of our proposed *Multitasking E2E BERT Parser* over *Base E2E BERT parser* (improvement due to the auxiliary task of typology prediction) is higher under mixed polyglot training (CL-Poly) settings as compared to under monolingual training settings (CL-Single).

5.5 Improving the performance of UDify with Linguistic Typology Knowledge

5.5.1 Introduction

UDify Kondratyuk and Straka (2019b) is the state-of-the-art mBERT based language-agnostic dependency parser, which takes the advantage of multilingual modeling to improve its performance on low-resource languages. Section 5.5.2 describes the architecture of UDify model in detail. The authors of UDify Kondratyuk and Straka (2019b) trained it on a joint polyglot corpus created by concatenating all training treebanks available in UDv2.3, and evaluated it on all test treebanks in UDv2.3 individually. Results outlined by Kondratyuk and Straka (2019b) show that for dependency parsing task, the UDify outperforms its baseline monolingual UDPipe Future

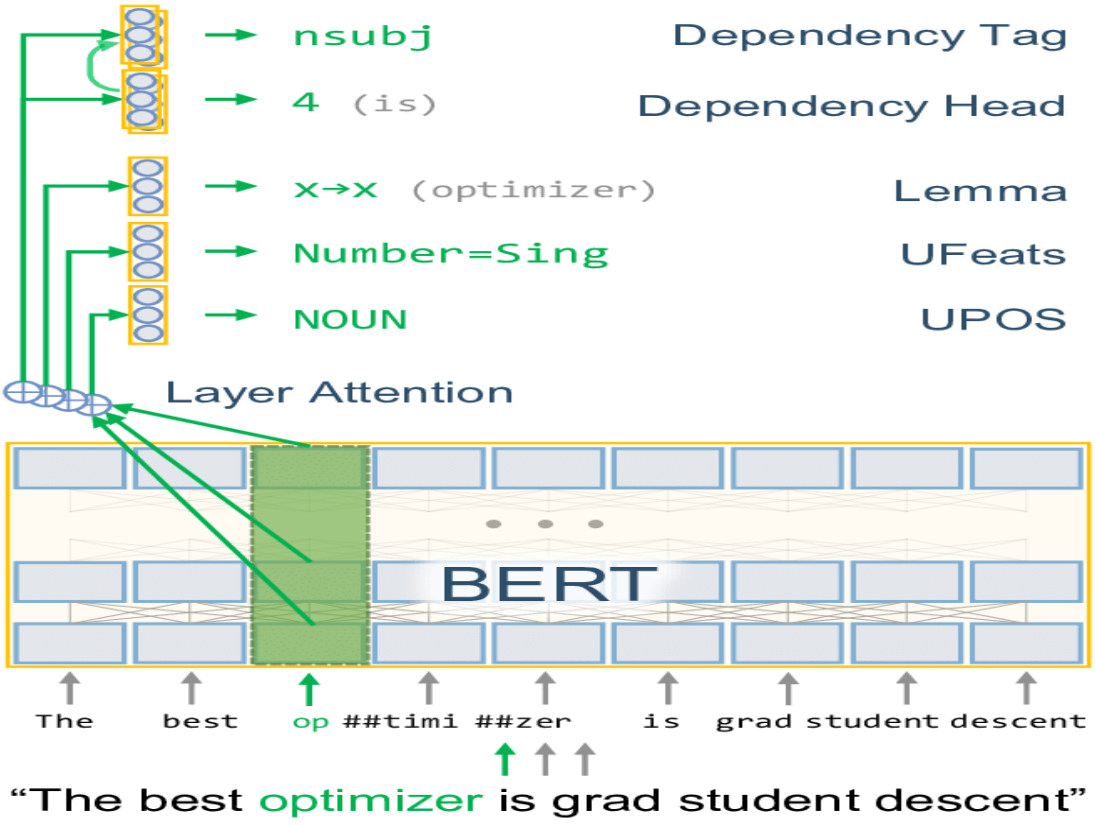


Figure 5.9: UDify Kondratyuk and Straka (2019b) model architecture.

Straka (2018) model by a large margin especially for low-resource languages, as the model benefit significantly from the cross-lingual transfer learning which occurs due to joint polyglot training.

However, the performance of UDify model on the low-resource languages (less represented in the polyglot training corpus) is still much lower than the performance of it on the high-resource languages which are well represented within the training corpus. In this work, we use linguistic typology knowledge to improve the cross-lingual transferring ability of UDify model even further, thereby significantly reducing this gap between model's performance on high-resource and low-resource languages.

We induce the linguistic typology knowledge available in URIEL Littell et al. (2017) database into the UDify model by adding an auxiliary task of linguistic typology feature prediction to it, within the multitasking framework. Section 5.5.3 will describe this knowledge induction process in details.

5.5.2 UDify model

UDify is a multitasking multilingual BERT based model which performs four key language-processing tasks simultaneously namely *UPOS-tagging*, *UFeat-tagging*, *Lemmatization* and *Dependency Parsing*. Model utilizes a single common mBERT based encoder for all the tasks, and individual task-specific decoders for each of the four tasks.
The encoder takes in the entire sentence as input, tokenizes it using BERT's pre-trained WordPiece Tokenizer Wu et al. (2016) and subsequently outputs contextual embeddings for each token. The architectures adopted by the UDify model for each of the task-specific decoders are described as follows. Figure 5.9 depicts the full architecture of the UDify model.

1. *UPOS-tagging*: For UPOS-tagging, the model adopts the standard neural sequence tagging architecture with softmax layer on the top. The decoder accepts embeddings generated from BERT encoder and outputs a probability matrix.

2. *UFeat-tagging*: The encoder takes in the entire sentence as input, and subsequently tokenizes the input sentence using BERT's pre-trained WordPiece Tokenizer. The architectures adopted by the UDify model for each of the task-specific decoders are described as follows.

3. *Lemmmatization*: The encoder takes in the entire sentence as input, and subsequently tokenizes the input sentence using BERT's pre-trained WordPiece Tokenizer. The architectures adopted by the UDify model for each of the task-specific decoders are described as follows.

4. *Dependency Parsing*: The encoder takes in the entire sentence as input, and subsequently tokenizes the input sentence using BERT's pre-trained WordPiece Tokenizer. The architectures adopted by the UDify model for each of the task-specific decoders are described as follows.

5.5.2.1 Word-embeddings

Previous studies Devlin et al. (2019) have shown that when fine-tuning mBERT on a downstream task, combining the output of the last few layers as the BERT output is more beneficial than just the last layer outputs. Hence UDify model computes the weighted sum of outputs all 12 BERT layers. It outputs this weighted sum corresponding to each token (from input sentence), as its mBERT based token-representation

vector. For each word, the model considers the representation-vector of its first token as its embedding, while ignoring the rest of its tokens.

5.5.3 Linguistic Typology prediction

To improve the cross-lingual transferring ability of UDify model, we added a fifth auxiliary task of Linguistic Typology prediction to it.

Our *Typology-predictor* is a simple deep feed-forward neural network with *sigmoid* activation function, which predicts the values of all typology features provided by the URIEL database Littell et al. (2017).

URIEL database is a collection of binary features extracted from multiple typological, phylogenetic, and geographical databases such as WALS Haspelmath (2009), PHOIBLE Moran et al. (2014), Ethnologue M. Paul Lewis and Fennig (2015) and Glottolog Hammarström et al. (2017). URIEL database can be accessed through Pyton PyPi library called *lang2vec*[4].

Let $\hat{N}$ be the number of typology features provided by URIEL database. Our *Typology predictor* would then output the probability vector $Pr_{ty} \in R^{\hat{N}}$ by applying equation 5.4.

$$Pr_{ty} = sigmoid(e_{</s>} * U + c) \tag{5.4}$$

Here $e_{</s>} \in R^d$ is the contextual embedding from the shared *mBERT Encoder* for end-token $< /s >$ of the input-sentence. $U \in R^{d*\hat{N}}$ and $c \in R^{\hat{N}}$ are weights and biases respectively. Pr_{Ty} comprises the probability of value of each URIEL binary feature being as 1, for the specific language being parsed.

The total-loss is computed by simply adding the *Typology Predictor* loss to *UDify* model's (as computed in Kondratyuk and Straka (2019b))

5.5.4 Experiments

This section describes the details of experiments conducted to evaluate our proposed model.

5.5.5 Experimental Setup

Both baseline *UDify* and the proposed *UDify+Typology-predictor* models are trained on a single large joint-polyglot corpus, created by concatenating all training datasets available in UDv2.5[5] together.

[4]https://pypi.org/project/lang2vec/
[5]https://universaldependencies.org/

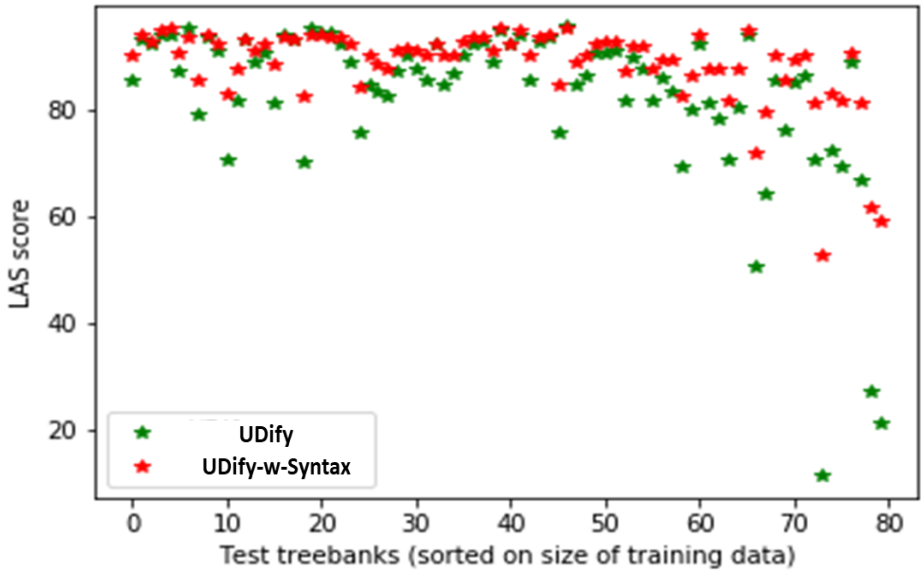


Figure 5.10: Trends in LAS achieved by *UDify* and *UDify-w-Syntax* models on all 80 test treebanks

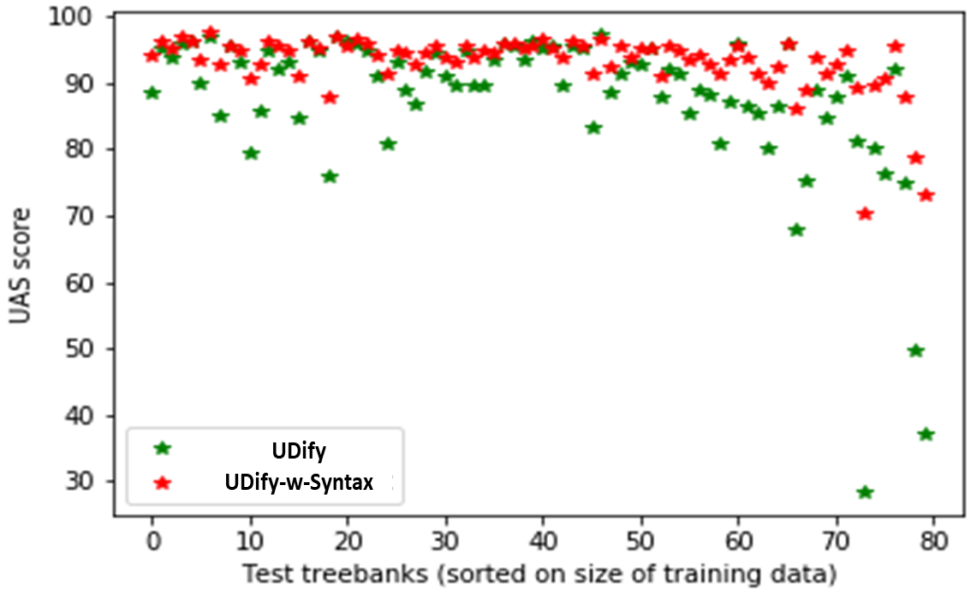


Figure 5.11: Trends in UAS achieved by *UDify* and *UDify-w-Syntax* models on all 80 test treebanks

Corpus	Model	UPOS	UFeats	Lemma	UAS	LAS	Typo F1
Overall (all UDv2.5 test-banks)	UDPipe	94.27	91.37	94.99	86.24	81.78	–
	UDify	94.03	89.33	90.92	87.84	82.83	–
	UDify-w-Lang_id	95.76	90.95	91.52	90.21	85.61	–
	UDify-w-Syntax	**95.89**	**92.05**	**91.87**	**93.18**	**88.4**	**74.6**
	UDify-w-Syntax+Semantic	94.04	88.06	87.09	89.26	83.84	73.33
	UDify-w-All	92.85	85.48	84.33	84.86	79.17	64.88

Table 5.13: Overall Results achieved by the baseline and all variants of our proposed model. These are average of all results outlayed in Appendix B.

Corpus	Model	UPOS	UAS	LAS
English-EWT (size: 25377)	UDify	97.73	94.64	90.04
	UDify+	98.32	95.73	91.41
French-GSD (size: 33399)	UDify	98.14	94.74	92.77
	UDify+	99.24	96.19	92.84
Buryat-BDT (size: 19)	UDify	60.23	36.98	21.52
	UDify+	73.73	73.25	59.1
Lithuanian-HSE (size: 2494)	UDify	90.47	80.1	70.38
	UDify+	93.56	90.14	81.6

Table 5.14: Selected results from Appendix B. **UDify+** refers to *UDify+Syntax* model

Before each training-epoch, we randomly shuffled all sentences in our polyglot training corpus, and subsequently fed mixed batches of sentences from this shuffled corpus into the model being trained, where each batch may contain sentences from any language or treebank (as done by authors of UDify Kondratyuk and Straka (2019b)).
We used a batch-size of 32, drop-out probability of 0.01 and the pre-trained mBERT model *cased_L-12_H-768_A-12* downloaded from tensorflow-hub[6]. We fine-tuned these hyper-parameters on *Dev* dataset for *English-EWT* treebank.

5.5.6 Results

We evaluated our proposed model on 80 test tree-banks available in UDv2.5 datasets individually. Appendix B provides the results achieved on each of these 80 test-

[6]https://tfhub.dev/tensorflow/bert_multi_cased_L-12_H-768_A-12/3

Distrb 1	Distrb 2	t-value	p-value
Typo F1	Diff	84.23	3.24e-23
Typo F1	Size	6.98	7.36e-11
Typo F1	UDify	1.42	0.16
Typo F1	UDify+	3.49	6.26e-4

Table 5.15: Results of t-test for correlation between various performance parmeters. **Typo F1**: Its F1 score achieved by *UDify+Syntax* for auxiliary task.; **UDify, UDify+**:UAS achieved by *UDify* and *UDify+Syntax* models; **Diff**: Improvement in UAS of *UDify+* over *UDify*

Corpus	Model	UPOS	UAS	LAS
Breton-KEB	UDify	63.67	63.97	40.19
	UDify+	62.15	60.65	34.23
Tagalog-TRG	UDify	61.64	64.73	39.38
	UDify+	62.38	63.9	38.31
Faroese-OFT	UDify	77.86	69.28	61.03
	UDify+	77.46	65.57	54.11
Naija-NSC	UDify	56.59	47.13	33.43
	UDify+	55.06	46.61	27.94
Sanskrit-UFAL	UDify	40.21	41.73	19.8
	UDify+	38.08	43.14	15.48

Table 5.16: Results achieved in zero-shot learning scenario. **UDify+** refers to *UDify+Syntax* model

treebanks, whereas table 5.13 outlines the average results on all these 80 treebanks. All scores are evaluated using the official CoNLL 2018 Shared Task evaluation script. We compared the performance of our model with two baselines namely *UDPipe Future*Straka (2018) and *UDify*.

URIEL database comprises of three categories of typology features namely *Syntactic*, *Semantic* and *Phonological* features. In this work, we evaluated three variants of our proposed model, based on the categories of features predicted by the typology-predictor within the auxiliary task, namely *UDify-w-Syntax* (predicts only syntactic typology features), *UDify-w-Syntactic+Semantic* (predicts syntactic and semantic typology-features) and *UDify-w-All* (predicts all the URIEL typology-features).

Furthermore, we evaluated the performance of *UDify-w-Lang_id* model. The architecture of it is identical to our proposed model but the linguistic-typology predictor is replaced by a simple language-id predictor.

5.5.7 Discussion

In this section we analyse the results outlined in section 3.9 to address the research questions **RQ7** and **RQ8** listed in section 1.1.2 as follows.

RQ7: For the state-of-the-art UDify parser which is a multilingual multi-tasking model that performs four key tasks simultaneously namely UPOS-tagging, UFeat-tagging, Lammetization and Dependency-parsing, when an auxiliary task of typology feature-value prediction is added to it, does it impact the performances of other four NLP tasks ?

As described in section 5.5.6, we evaluated the three variants of our proposed multitasking UDify model with added auxiliary task of linguistic typology prediction namely *UDify-w-Syntax*, *UDify-w-Syntactic+Semantic* and *UDify-w-All* variants, distinct based on the typology feature-types been included. It is evident in results outlined as Appendix B that we observed similar trends in performance on all four tasks namely *UPOS-tagging*, *UFeats-tagging*, *Lammetization* and *Dependency Parsing*.
It can be observed in the results outlined in section 5.5.6 that the *UDify-w-Syntax* variant of our proposed model outperforms the other two variants of it, for most of the test-treebanks, despite the fact that the *UDify-w-Syntax+Semantic* and *UDify-w-All* variants utilizes more typology-features than the UDify-w-Syntax variant.
The reason being that since all four tasks performed by the UDify model namely UPOS-tagging, UFeats-tagging, Lammelization and Dependency Parsing are syntactic tasks, only the syntactic typology-features are relevant to these tasks. Henderson (2004) proved that, having large number of unrelated features makes it difficult for a neural-network model to effectively learn from provided training-data, and thereby would lead to drop in performance.
It is also evident in results outlined in Appendix B (displayed as figures 5.10 and 5.11) that for high-resource languages, the *UDify+Syntax* model shows only marginal improvement in performance over *UDify* whereas for low-resource languages it shows strong improvement in performance on all four tasks. Such trends can also be observed in table 5.14. Table 5.16 on the other hand, outlines results obtained on selected languages which are not represented in the training data at all (zero-shot learning). For such treebanks, *UDify+Syntax* under-performs *UDify*.
Hence it can be inferred that the auxiliary task of linguistic typology prediction, does lead to significant improvement in performance of *UDify* in the *Few-shot* learning scenario, but does not lead to any improvement within zero-shot learning scenario.

The overall results (average) summarised in table 5.13, show that *UDify+Syntax* outperformed baselines *UDPipe Future* and *UDify* model for almost all 80 test-treebanks. Hence we can infer that adding the auxiliary task of syntactic typology prediction to UDify model does lead to the improvement in performance.
Furthermore, to ensure that the improvement is indeed due to typology knowledge injection, we compared the the performance of *UDify-w-Syntax* model with the performance of *UDify-w-Lang-id* model. The architecture of it is identical to our proposed model but the linguistic-typology predictor is replaced by a simple language-id predictor. Results in Appendix B show that the *UDify-w-Syntax* model outperformed the *UDify-w-Lang-id* model on almost all 80 target languages.

RQ8: Is there any correlation between the performance the end-to-end parser on the main dependency-parsing task and the performance of it on the auxiliary task of linguistic typology feature-value prediction ?

To ensure that the auxiliary task of linguistic typology-prediction is indeed responsible for the improvement in performance of *UDify*, we conducted numerous statistical t-tests to find the correlation between F1 scores achieved by the *UDify+Syntax* model for the auxiliary-task of typology-prediction, and various other performance parameters including the improvement in performance of *UDify+Syntax* over *UDify*. Table 5.15 outlines the results of these t-tests.
The results in Table 5.15 show that there is indeed a strong correlation between performance scores on the main DP task and the score achieved on the auxiliary task.

5.6 Conclusion

In this chapter, we proposed and evaluated the performance of an *End-to-end BERT Based Dependency Parser* which can parse a sentence by directly predicting relative head-position tag for each word within input sentence. This is inspired by a monolingual BiLSTM based End-to-end Dependency parser.
Subsequently, we added the auxiliary task of Linguistic typology prediction to our *Base E2E BERT parser* to observe the change in performance under different settings. Our results show that adding such auxiliary task leads to improvement in performance of *Base E2E BERT Parser* within Cross-lingual settings under the *Few-shot* learning scenario whereas no improvement is observed within the *Zero-shot* learning scenario. As far as we are aware this is the first work to evaluate the end-to-end Dependency

Parsing framework, within the cross-lingual settings. The future work could involve exploring same auxiliary task for other transformer based language models such as GPT-2, XLM-R etc. Further, other frameworks apart from Multitask learning such as GANs can be explored to induce linguistic typology knowledge within Multi-lingual Parser.

In this chapter, we also aimed to improve the performance of the state-of-the-art language-agnostic UDify parser by injecting the linguistic typology knowledge available in URIEL database to improve the cross-lingual transferring ability of it. We injected the typology knowledge in UDify model through an auxiliary task, in the multitasking settings.

Chapter 6

End-to-end Enhanced Dependency-parsing for Typology Feature Prediction

This chapter is based on our research work published as following paper:

- **End-to-end mBERT based Seq2seq Enhanced Dependency Parser with Linguistic Typology knowledge.** In Proceedings of SPECIAL INTEREST GROUP ON NATURAL LANGUAGE PARSING (SIGPARSE) AT ACL 2021

In chapter 5 we described the task of dependency parsing as well as two proposed and evaluated approaches to cross-lingual DP with linguistic typology knowledge injection. In this chapter we describe the task of Enhanced Dependency Parsing and describe a cross-lingual multitasking approach to EDP in detail.

The Enhanced Dependency Parsing (EDP) framework Schuster and Manning (2016); Coke et al. (2016) is an interesting extension of the standard DP framework, which provides additional significant syntactic and semantic knowledge that is missing in a standard dependency parse-tree. Such additional knowledge does lead to an improvement in performance on numerous downstream NLP tasks.

Our proposed model is an extension of our UDify based multitasking model for DP described in section 5.5.2, with an addition auxiliary task of end-to-end EDP task. The architecture of the end-to-end EDP auxiliary component is indeed inspired by the monolingual End-to-end Seq2seq Dependency-Parser proposed by Li et al. (2018).

Section 6.1 describes the Enhanced Dependency Parsing task in details. Subsequently, section 6.1.2 provides a brief literature review of various approaches to cross-lingual EDP task. In section 6.2, we describe our proposed model and subsequent sections describe the experimentation conducted to evaluate the proposed model and outline the results achieved.

6.1 Enhanced Dependency Framework

The EDP framework also commonly adopts the UD annotation scheme (section 5.2.1) similar to the basic DP framework (section 5.1.2). However, as compared to the standard DP framework, the EDP framework aims to define the relationships between head and dependent words more explicitly by adding more relationship-types or augmenting the basic UD relationship names with additional knowledge. Hence, an Enhanced Dependency-tree is an extension of the standard Dependency-tree comprising all relations of the dependency tree with a few additional attributes. Section 6.1.1 outlines these additional attributes within the EDP framework.

6.1.1 EDP Frameowork attributes

This section outlines the rules in the EDP framework which are distinct from standard DP framework as subsequent sub-sections. We use the examples provided by Schuster and Manning (2016) to explain these rules. Figure 6.1 depicts these example sentences.

6.1.1.1 Augmented Modifiers Rule

In the standard DP framework, for a modifier relationship, the head-word that is modified by a prepositional phrases (PP) is related to the prepositional complement-word rather than the preposition itself. However this modifier relationship does not provide any information about the actual preposition that is infact modifying the head-word. Such knowledge is useful for numerous downstream tasks.
For example, in example-sentence 1a in Figure 6.1, the word ***house*** in the sentence ***the house on the hill*** is modified by the preposition ***on***. Hence, in the standard dependency-tree of the sentence, the head-word ***house*** is therefore connected to prepositional complement-word ***hill*** with relationship-type ***noun-modifier (nmod)*** but no information about the preposition itself is provided in the tree. This issue is addressed in the corresponding enhanced dependency-tree depicted in Figure 6.1 where the relationship-label ***nmod*** is augmented with the preposition word ***on*** (as ***nmod:on***). Similarly, in example 1b of Figure 6.1, the noun-modifier relationship between words ***brushed*** and ***eating*** is augmented with preposition ***after***.
Like the noun-modifier (nmod), in EDP framework the adverbial clause modifier (advcl) relationship types are also augmented with respective preposition word.
If a modifier relationship comprises of multi-word preposition than entire phrase is augmented to the relationship-label as shown in example 1c id in Figure 6.1

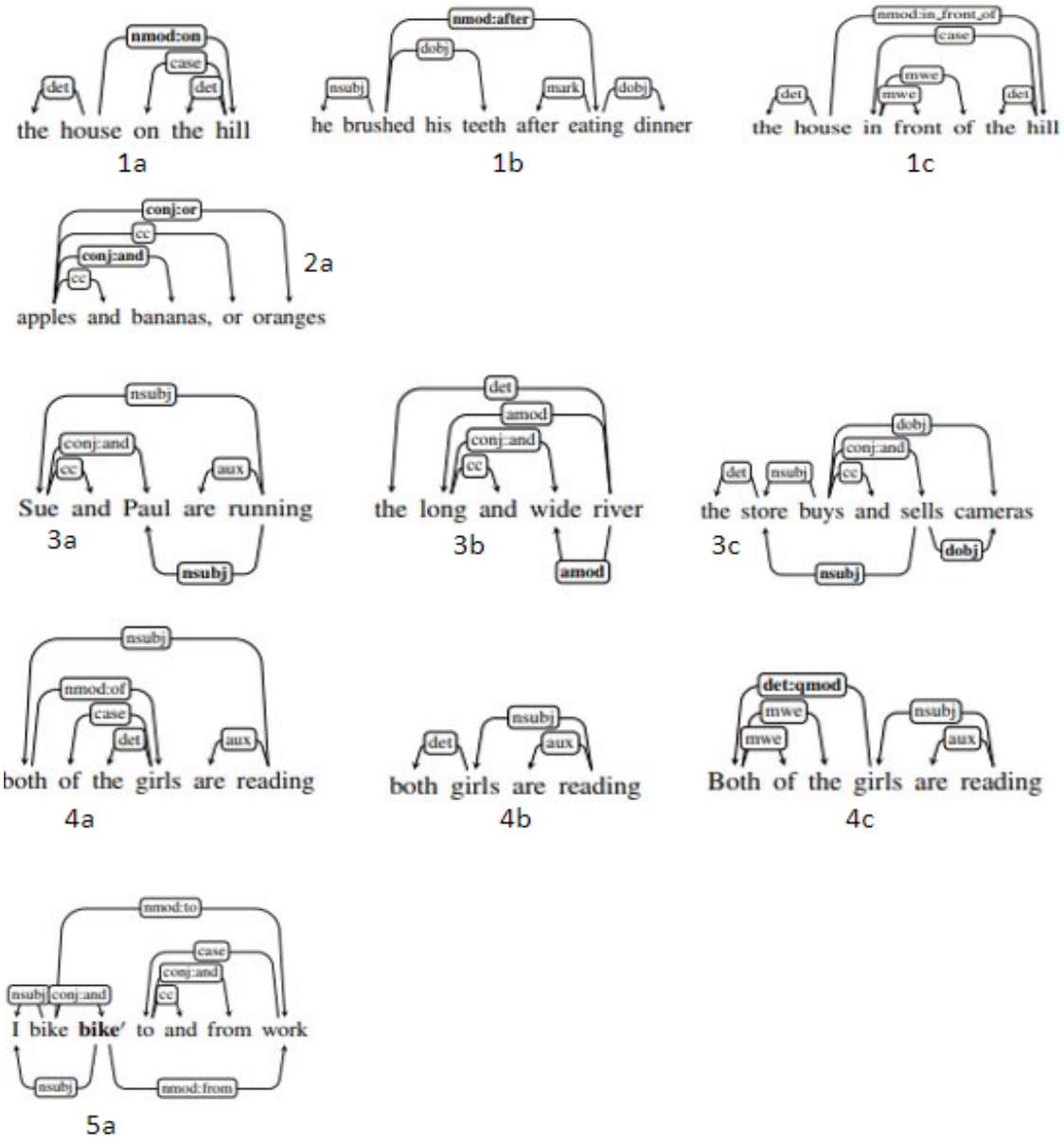


Figure 6.1: Demonstrations of EDP attributes. Examples from Schuster and Manning (2016)

6.1.1.2 Augmented Conjuncts Rule

Similar to the modifier, in the EDP framework the conjunct relationship-types are augmented with their corresponding conjunction word. For example in example 2a in Figure 6.1, the conjunction words ***apple*** and ***banana*** are augmented with the conjunction word ***and*** whereas the words ***apple*** and ***orange*** are augmented with the conjunction word ***or***.

6.1.1.3 Propagated Head or Dependents Rule

If a sentence comprises of one or more conjoined phrases and if the entire phrase has syntactic-relationships with other words in the sentence, in the standard dependency-tree of this sentence such explicit relationship are directed to/from only the first conjunct in the phrase. On the other hand, in the ED framework such relationships are marked for both conjuncts. Although such propagation violates one of the standard DP constraint of single head on each word in the sentence (section 5.1.2).
For example, in example 3a in Figure 6.1, both ***Sue*** and ***Paul*** are subjects of verb ***running*** hence both of them are connected to word ***running*** with relationship-type ***noun-subject (nsubj)***.
Similarly, if a sentence comprises of conjoined adjectival or adverbial phrases, all conjuncts are connected to the respective noun or verb as evident in example 3b of Figure 6.1.
Likewise similar rules are applied for any conjoined verb phrases as evident in example 3c in Figure 6.1.

6.1.1.4 Quantificational Determiners Rule

If a sentence comprises a multi-word construction phrase, with relations to other words in the sentence, such relationships are often not accurately depicted in its standard dependency-tree. For example, consider two sentences ***both of the girls are reading*** and ***both girls are reading***. Figure 6.1 depicts dependency-trees of these relationships (as 4a and 4b). It is evident that for the first sentence, the word ***both*** is marked as the subject of verb ***reading*** whereas in second sentence the word ***girl*** is marked as the subject, even though both sentence mean the same.
In EDP framework, any relationship with such a construction phrase is always marked at the semantically significant part of such a multi-word phrase (eg: ***girl*** not ***both***) while a quantificational modifier relationship is added between the words within the phrase. For example, in example 4c in Figure 6.1 relationship ***nsubj*** to word ***girl***

while a new relationship ***det:qmod*** is added between the words ***both*** and ***girl***.
This attribute can also violate one of the standard DP constraint of single head on each word in the sentence (section 5.1.2).

6.1.1.5 Conjoined prepositions

As explained in section 6.1.1.1, for a modifier relationship-type the EDP framework requires the preposition word (or phrase) to to be augmented to a relationship-label. However, in some scenarios, the proposition can be a conjoined phrase with both prepositions modifying the head-word independently. For example, in the example sentence ***I bike to and from work***, there are two conjunct propositions namely ***from*** and ***to*** discussing modifying head-word ***bike*** independently. In this scenario, EDP framework adds duplicates of the head-word and each one copy of the head-word connected to one of the prepositions, as evident in example 5b in Figure 6.1.

6.1.2 Approaches to Cross-lingual EDP

The task of cross-lingual EDP for low-resource languages was brought into attention as the *IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies* Bouma et al. (2020). The task typically provided a training and test datasets in multiple languages and the participants were invited to build and evaluate multilingual models that can be applied to more than one languages.
Most models proposed for the task are transformer based models (section 2.1.1.4). Approaches such as He and Choi (2020); Grünewald et al. (2021); Kanerva et al. (2020) fed the token indexes directly into a pre-trained transformers to predict ED relationships. On the other hand, rather than fine-tuning the pre-trained transformer models to predict dependencies, numerous approaches instead used word-embeddings/token-representations generated by pre-trained transformers to be fed into another BiLSTM models. These transformer based embeddings are often combined with other linguistic features (cross-lingual) such FastText Wang et al. (2020), character-based features as well as the features from predicted POS tags, morphological features and basic UD parse-tree Barry et al. (2020).
Several proposed approaches to CL-EDP such as Orange Heinecke (2020), FASTPARSE Dehouck et al. (2020), UNIPI Attardi et al. (2020), CLASP Ek and Bernardy (2020), ADAPT Barry et al. (2020) etc. are heuristic based approaches that aim to predict the ED relationships by applying hand-drafted enhancement rules to the predicted standard DP relationships. On the other hand, approaches such as Emory

NLP He and Choi (2020), ShanghaiTech Wang et al. (2020), RobertNLP Grünewald et al. (2021) are graph-based approaches that do not derive EDP relationships from standard UD relationships through enhancement (or conversion) of its dependency relationships, but instead directly produce EDP trees for a given input-sentence. Hershcovich et al. (2020) is the only transition-based system proposed that uses the stack-LSTM architecture Dyer et al. (2015).

Similar to our proposed model described in this chapter, TurkuNLP Kanerva et al. (2020) is another model that utilized UDify model Kondratyuk and Straka (2019b) for the EDP task. The TurkuNLP model aimed to represent the ED relationships into a standard DP format by combining multiple edges into a single edge with a complex labels. The authors reduced the total number of edge-labels by adopting a mechanism of delexicalising the labels of the edges. Subsequently, they fine-tuned UDify parser model to predict these modified standard DP relationships. This TurkuNLP model was subsequently outperformed by Wang et al. (2020) which used a second-order inference methods involving the *Mean-Field Variational Inference*.

The Shared task is repeated again in 2021 where numerous researches proposed distinct and improved methods to Cross-lingual EDP. TGIF Shi and Lee (2021) is the best performing model for the 2021 Shared Task and the current state-of-the-art. It is a hybrid model that performs EDP in two consecutive steps. Firstly they used a graph-based parser (section 5.1.3.2) to predict the minimum spanning tree comprising of all dependency relations. Subsequently, they predict any additional graph-edges (of EDP parse-tree) that is not present in the spanning trees. The authors also adopted a language-specific fine-tuning strategy, where they first trained the model on mixed polyglot corpus created by concatenating all available training copra in many languages and subsequently fine-tuned on each individual training language separately.

Hyper-parameter	**Value**
Dropout prob.	0.01
Bach-size	32
Number of steps per epoch	Size of training corpus / 32
Epochs	150
BERT Model	bert_multi_cased_L-12_H-768_A-12

Table 6.1: Hyper-parameters

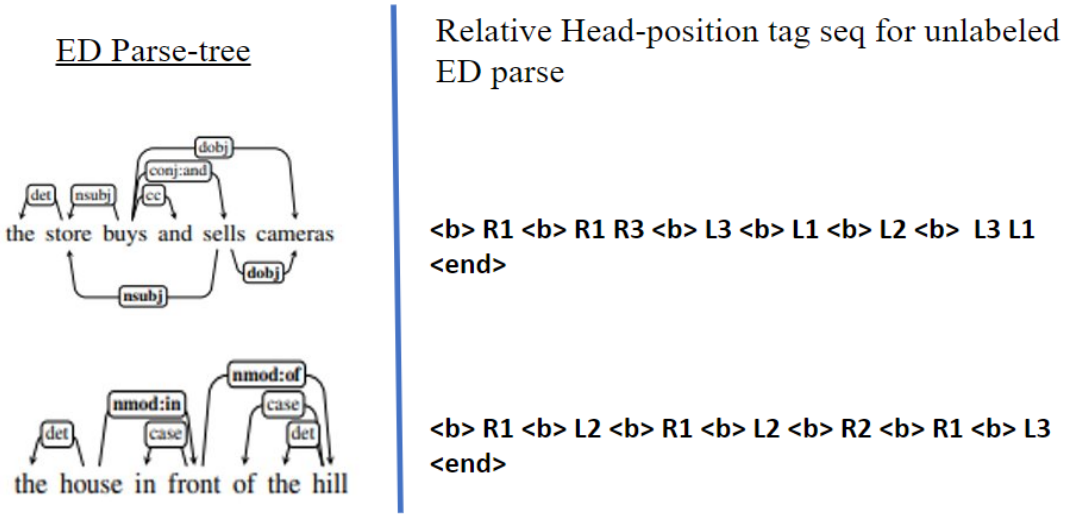


Figure 6.2: Example Enhanced Dependency Parse trees represented as *Relative Head-position tag-sequences*

Language	UPOS	UFeats	Lemmas	UAS	LAS	ELAS
bg	99.01	35.97	98.1	93.87	90.63	81.85
en	95.37	33.47	96.76	87.57	85.46	78.8
et	96.89	35.74	96.55	86.21	83.36	76.63
lv	96.62	35.91	96.55	89.51	85.89	78.97
lt	93.8	30.59	93.66	79.05	74.42	77.22
ru	98.45	36.92	98.49	93.27	92.01	79.53
sk	96.92	23.48	95.71	90.89	88.19	81.15
sv	96.45	34.06	93.06	86.54	82.78	76.02

Table 6.2: Results achieved by the Base E2E-w-Typo parser for all the tasks in *IWPT 2021 shared task*

6.2 mBERT based Seq2seq ED Parser

This section describes our proposed end-to-end model for cross-lingual EDP task which is inspired by our proposed end-to-end model for standard DP task (section 5.4). Figure 6.3b depicts the architecture of the proposed ED parser.

Our proposed *End-to-end ED Parser* is an extension of the UDify Kondratyuk and Straka (2019b) model described in section 5.5.2, with one additional component namely the *Relative Head Sequence predictor* which predicts the relative head-position of the tag-sequence representing the unlabelled enhanced-dependency parse-tree of the input sentence (as the fifth auxiliary task in the multitasking UDify model). Section 6.2.1 describes the mechanism of representing an enhanced dependency tree as a sequence of relative head-position tag sequence. Subsequently, section 6.2.2

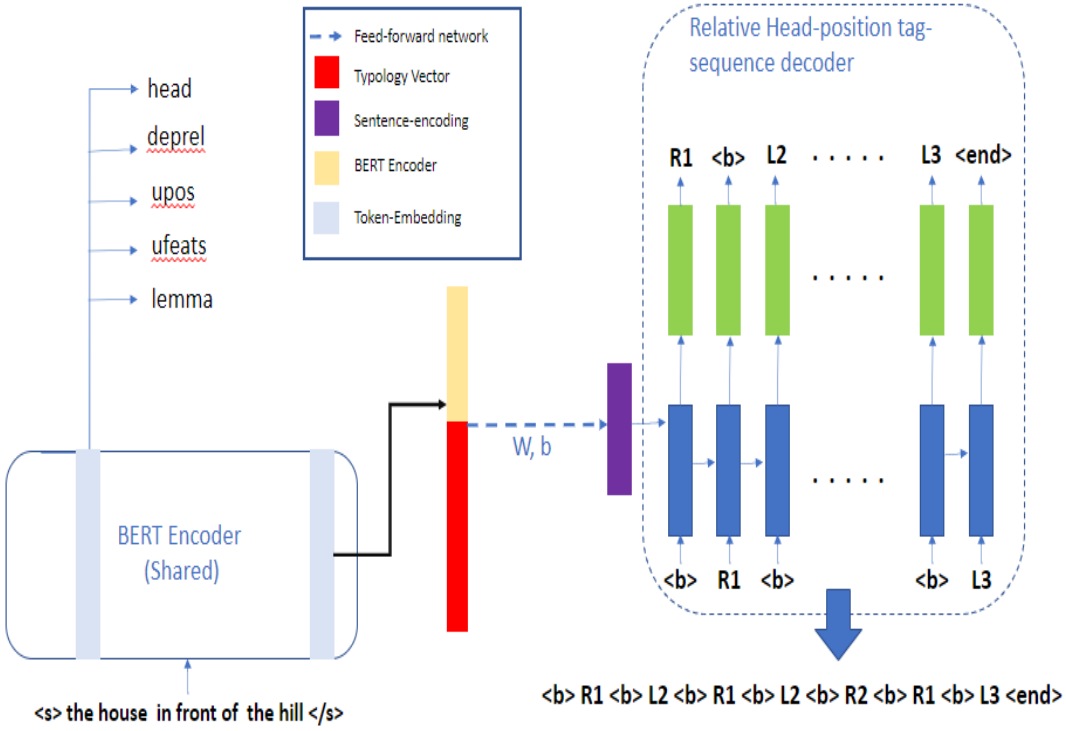


Figure 6.3: Architecture of the *Relative Head-position Sequence predictor* model for EDP task

describes architecture of the *Relative Head Sequence predictor* component of the proposed model. Our proposed model injects linguistic typology knowledge into the *Relative Head Sequence predictor* to improve its cross-lingual transferring ability.

6.2.1 ED parse-tree as relative head-position tag sequence

Given a sentence of length T, its unlabelled ED parse-tree can be represented by a relative-head tag-seq of length $\hat{T}$ such that $\hat{T} \geq 2T + 1$. Figure 6.2 depicts the representations of sample unlabelled enhanced-dependency parse-trees as their relative sequences of relative head-position tags. Here, the tag $< b >$ represents the next-token whose heads are pointed by the subsequently predicted relative-head position tags (until the next $< b >$ tag is predicted).

6.2.2 Relative Head Sequence predictor

As evident in Figure 6.3b, our *Relative Head Sequence predictor* is a standard LSTM based Seq2seq neural-network Sutskever et al. (2014) which takes in the entire input-sentence encoding vector as input, and sequentially predicts the relative head-position

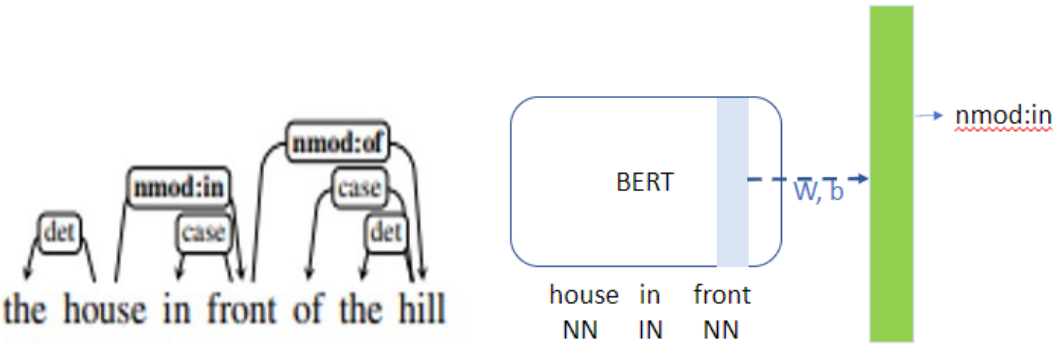


Figure 6.4: Architecture of the *Label predictor*

Model	UPOS	UFeats	Lemmas	UAS	LAS	ELAS
combo	97.62	94.95	94.39	91.55	89.14	85.01
dcu-epfl	96.32	91.81	95.15	87.44	84.3	86.89
fastparse	97.24	93.0	95.84	78.23	72.44	67.07
grew	97.24	93.0	95.84	89.6	87.03	82.95
robertnlp	97.89	94.06	0.01	93.15	90.4	88.44
shanghaitech	0.46	32.78	0.01	4.18	1.27	88.37
tgif	0.46	32.81	0.01	10.93	0.94	90.67
unipi	96.37	91.75	95.17	90.55	87.98	84.42
Base E2E	**96.36**	**32.76**	**95.17**	**87.63**	**84.54**	**76.32**
Base E2E-w-Aux	**96.93**	**32.58**	**95.71**	**87.86**	**84.67**	**78.7**
Base E2E-w-Typ	**97.54**	**33.28**	**96.22**	**88.43**	**85.28**	**79.32**

Table 6.3: Comparison of results achieved by all ED parsers in **IWPT 2021 Shared task** and the variants of proposed End-to-end parsers

tag-sequence, one tag at a time.

6.2.2.1 Input sentence-encoding

The sentence-encoding $e^X \in R^d$ of any input sentence $X = x_1, x_2, ...x_T$ is computed by applying equation 6.1.

$$e^X = W * [BERT(X); TY_l] + b \tag{6.1}$$

Here $BERT(X)$ is the output embedding-vector from the UDify's shared mBERT encoder for the end-of-sentence token $< /s >$ of input-sentence and TY_l is a *Linguistic-typology* vector of language l being parsed. Each value within the TY_l represents a single typology-feature from *WALS* Haspelmath (2009) database having a specific

Languages	Base E2E		Base E2E-w-Aux		Base E2E-w-Typo	
	LAS	**ELAS**	**LAS**	**ELAS**	**LAS**	**ELAS**
bg	90.03	78.45	90.33	80.85	90.63	81.85
en	84.46	75.4	84.96	78.1	85.46	78.8
et	82.46	74.03	83.06	76.33	83.36	76.63
lv	85.19	76.67	85.29	78.57	85.89	78.97
lt	73.52	73.52	73.72	76.92	74.42	77.22
ru	91.01	76.33	91.61	78.83	92.01	79.53
sk	87.49	77.45	87.39	80.85	88.19	81.15
sv	82.18	73.12	82.08	75.52	82.78	76.02

Table 6.4: LAS and ELAS achieved by all three variants of the proposed End-to-end Seq2seq Enhanced Dependency parsing

integer value. Equation 6.1 involves the concatenation of the *BERT-output* and the *Typology* vectors, followed by dimension reduction through a feed-forward network. Feeding typology features together with the input sentence could improve the cross-lingual transferring ability of the multilingual model, as shown by Ammar et al. (2016).
For the proposed model, we use all the word-order and constituency features in WALS Haspelmath (2009) database excluding trivially redundant features as excluded by Takamura et al. (2016).

6.2.2.2 Training

We trained our *mBERT based Seq2seq ED Parser* on a single large joint-polyglot corpus, created by concatenating all the treebanks available in the training dataset provided for the *IWPT 2021 Shared task*.
Before each training epoch, we randomly shuffle all sentences in our polyglot training corpus, and subsequently feed mixed batches of sentences from this shuffled corpus into the model being trained where each batch may contain sentences from any language or treebank (as done by authors of UDify Kondratyuk and Straka (2019b)).
We optimized the weights of our multitasking model by minimizing the total loss as the sum of sparse cross-entropy losses for all five tasks namely *UPOS-tagging*, *UFeat-tagging*, *Lemmatization*, *Dependency Parsing* and *Relative Head-position Sequence prediction*.

6.2.2.3 Predicting

The ED parsing of any unknown input-sentence $X = x_1, x_2, ...x_T$ can be performed by extracting the most probable correct relative head-position tag-sequence. The correct relative head-position tag-sequence would satisfy following constraints.

1. The sequence should start with $< b >$ and end with $< end >$.
2. For each word in $x_i \in X$, the relative head-position tag assigned to it should be within the range of the sentence. For example, within the sentence **"the house in front of the hill"**, the word '*the*' can not have tags L_2, L_3, L_4, L_5, L_6 and the word '*hill*' can not have any right tags, as these are outside the range of the sentence.
3. The label sequence should not generate any cycles within the dependency tree.
4. One of the words should have the head at $< root >$ token.
5. The sequence should contain the number of $< b >$ tags equal to number of tokens in the input sentence X.

We used dynamic programming with beam-search to efficiently extract the most probable relative head-position tag-sequence which satisfies the above listed relative head-position tag-sequence, out of all possible sequences.

6.2.3 Label Predictor

Figure 6.4 depicts the architecture of our *Label predictor* model. It is an mBERT based multi-class classifier with a softmax layer on top. The model takes as input the token-seq segment from the input sentence ranging from head to tail, as well as its corresponding predicted POS-tag sequence. The model outputs the probabilities of all possible ED dependency labels to be assigned to the given relation.
The *Label-predictor* is trained on all ED relationships available in training dataset for *IWPT 2021 Shared task*. The parameters of the mBERT encoder of our *Label predictor* are initialized with the parameters of the fine-tuned mBERT encoder of our *Relative Head-position tag-sequences*.

6.3 Experiments

In this section we aim to evaluate the proposed End-to-end BERT based Enhanced Dependency Parser. These experiments aim to fulfill the following novel objectives:

1. To evaluate and compare the performance of our proposed cross-lingual end-to-end model for EDP task with the other state-of-the-art more complex models for the cross-lingual EDP. Such evaluation is significant as it is much simpler in design and therefor much time and space efficient.

2. To evaluate the impact of injection of linguistic typology knowledge into the end-to-end model. Furthermore, we aim to determine the best framework for such typology knowledge injection.

We experimented with three variations of our proposed *End-to-end Seq2seq ED-parser* namely **Base E2E**, **Base E2E-w-Aux** and **Base E2E-w-Typ** models. The **Base E2E** model has similar architecture as depicted in Figure 6.3 but without the typology vector. Thus, **Base E2E** does not use linguistic typology knowledge. On the other hand, the architecture of **Base E2E-w-Aux** is similar to **Base E2E** with an addition auxiliary task of predicting URIEL features of type *WALS-Syntax*, similar to *UDify-w-Syntax* model for standard DP described in section 5.5.6. Finally the *Base E2E-w-Typo* model feeds-in the typology features directly as shown in Figure 6.3.
All variants of the *End-to-end Seq2seq ED-parser* are trained on a large joint polyglot corpus created by concatenating all the treebanks in the provided training dataset for *IWPT 2021 Shared Task*. We evaluated the parsers on the test corpora provided for the *IWPT 2021 Shared Task* in eight distinct languages namely *bg*, *et*, *en*, *lv*, *lt*, *ru*, *sk* and *sv*. We outline the results achieved by our proposed model in detail in section 6.4. Table 6.1 outlines hyper-parameters used in the experiments. These values are obtained by minimizing the training loss for *English-EWT Corpus* provided in the *dev* dataset provided for the *IWPT 2021 Shared Task* for **Base E2E-w-Typ** variant.

6.4 Results and Analysis

This section outlines the results obtained by the experiments described in section 6.3 while addressing the research questions **RQ9** and **RQ10** as follows. All the results are calculated using the evaluation script provided by the *IWPT 2021 Shared task*.

RQ9: Does the cross-lingual mBERT based End-to-end Enhanced Dependency Parser perform at par with various state-of-the-art cross-lingual approaches to the enhanced dependency parsing task?

Table 6.3 compares the average results (average of all languages) by all three variants of our proposed end-to-end parser with all other participant models of *IWPT 2021 Shared Task*, on all five tasks namely *UPOS-tagging*, *UFeat-tagging*, *Lemmalization*, *DP* and *EDP*. It is evident that the proposed End-to-end EDP parser performed at par with state-of-the-art approaches to EDP task including other participant approaches to *IWPT 2021 Shared Task*, while being much simpler in design.

RQ10: Does linguistic typology knowledge injection into a cross-lingual mBERT based End-to-end Enhanced Dependency Parser improves its performance ? Is it better to feed-in the linguistic typology knowledge into the model directly along with word-representations, or to inject typology knowledge though an auxiliary task ?

Table 6.4 compares the *Enhanced Unlabelled Attachment Score (EUAS)* and *Enhanced Labelled Attachment Score (ELAS)* achieved by all three variants of the proposed End-to-end Seq2seq Enhanced Dependency parsing. It is evident from the results in tables 6.4 and 6.3 that the linguistic typology knowledge induction indeed led to improvement in performance on the ED task, as both *Base E2E-w-Aux* and *Base E2E-w-Typ* models outperformed the *Base E2E* model for all the target languages. The results also show that directly feeding-in the typology knowledge into the end-to-end parser leads to better performance then injecting this knowledge through the auxilliary task, as the *Base E2E-w-Typ* model outperformed the *Base E2E-w-Aux* for all the target languages.
Table 6.2 outlines results achieved by the *Base E2E-w-Typ* model on all eight blind test-corpora on which the model is evaluated, for all the six tasks. Appendix C outlines all the results achieved by all the participants of *IWPT 2021 Shared tasks* for reference.

6.5 Conclusion

Enhanced Dependency Parsing framework in an interesting extension of standard Dependency Parsing framework, such that an Enhanced Dependency parse-tree com-

prises of additional syntactic and semantic information which is missing in the standard dependency parse-tree. Such additional knowledge is useful in numerous downstream tasks.

In this work we proposed and evaluated a multitasking end-to-end mBERT based model for cross-lingual EDP task. As far as we are aware, this is the first work that evaluated an end-to-end approach to EDP task. Subsequently we injected linguistic typology knowledge into the proposed framework to examine the impact of such knowledge injection on its performance. We also evaluated various frameworks for such typology knowledge injection. This is the first work that aimed to utilise linguistic typology knowledge available in an external database to improve the performance of an Enhanced Dependency Parser.

Our results show that the proposed end-to-end model preformed at par with the state-of-the-art models while being much simpler in design and therefore much more time and space efficient than the state-of-the-art models. Furthermore, the results also proved that injecting the linguistic typology knowledge does indeed lead to improvement in performance of the parser significantly.

Chapter 7

Cross-lingual Semantic Role Labelling with ValPal Database Knowledge

This chapter is based on our research work published as following paper:

- **Cross-lingual Semantic Role Labelling with the Valpal database knowledge.** In Proceedings of THE 3RD WORKSHOP ON KNOWLEDGE EXTRACTION AND INTEGRATION FOR DEEP LEARNING ARCHITECTURES (DEEP-LIO) AT ACL 2022

Semantic role labeling (SRL) is the task of identifying various semantic arguments (such as Agent, Patient, Instrument, etc.) for each of the target verb (predicate) within an input sentence. SRL is useful as an intermediate step in numerous high level NLP tasks, such as information extraction Christensen et al. (2011); Bastianelli et al. (2013), automatic document categorization Persson et al. (2009), text-summarizing Khan et al. (2015) question-answering Shen and Lapata (2007) etc. However state-of-the-art neural-network approaches to SRL task are supervised approaches that require large annotated training dataset, thus leading to data-sparsity issue in low-resource languages. Similar to dependency parsing task (chapter 5), various cross-lingual approaches are applied for SRL as well. Cai and Lapata (2020) is the state-of-the-art approach to cross-lingual SRL which is trained on English and can be utilised for other low-resource languages. In this work we inject the knowledge available in ***ValPal database***, which is a comprehensive semantic typology database, into this state-of-the-art cross-lingual semantic role labeller. Such knowledge injection should improve the performance of the model.

Section 7.1 provides a high-level overview of the semantic role labelling task while section 7.2 provides a review of various cross-lingual approaches to SRL task. Section 7.3 describes Valpal database whereas section 7.5.1 will describe the process of injecting this valpal database knowledge into the state-of-the-art Cai and Lapata (2020) model. Subsequent sections provide experimental details and discuss the results obtained.

7.1 Semantic Role Labelling

Semantic Role Labeling/Shallow semantic-parsing is the task of assigning distinct labels to words and phrases in a sentence that indicate their semantic role within the sentence. These semantic roles include roles such as *Agent*, *patient*, *instrument*, *beneficiary* etc. In other words, SRL task aims to identify who did what to whom and how in a sentence.
The task of semantic-parsing Kamath and Das (2018) aims to represent the entire meaning of a sentence either as a first-order-logic rule or as a semantic graph. Such representations can significantly differ based on the word order. For example, consider the three example sentences listed as follows.

1. Jane baked the cake for Harry
2. Harry enjoyed the cake by Jane
3. The cake was prepared by Jane for Harry

Although all three sentences convey the same meaning i.e. *Jane* is the baker, the *Cake* is baked and *Harry* ate it, yet the sophisticated syntax-based meaning-representations such as *Elementary Dependency Structures* representation Buys and Blunsom (2017), Prague Tectogrammatical Graphs Zeman and Hajic (2020) etc. of all three sentences are significantly different. Semantic Role Labeling is a word-level meaning-representation which resolves this issue. Figure 7.1 which depicts the semantic role labels for the three sentences listed previously. If a sentence has more than one verbs, each word in that sentence is assigned a distinct semantic role label with respect to each verb predicate. Hence a unique semantic role label sequence is extracted for each verb-predicate in the sentence independently. Generally, SRL sequences of all verbs are represented in the conllu format (section 5.2.1) as shown in figure 7.2.

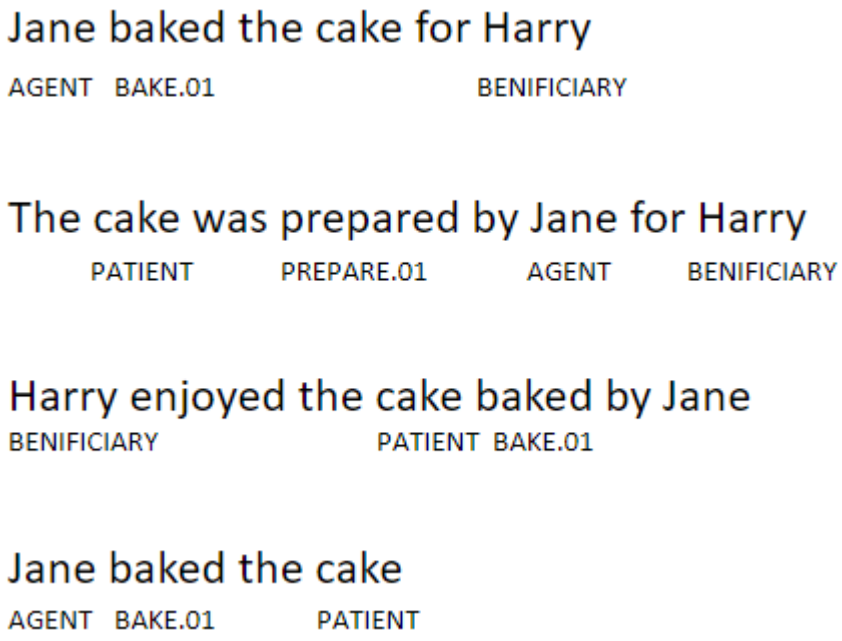


Figure 7.1: Semantic Role labels in the example sentences.

```
# sent_id = weblog-blogspot.com_nominations_20041117172713_ENG_20041117_172713-0002
# text = President Bush on Tuesday nominated two individuals to replace retiring jurists on federal courts in the Washington area.
1       President       President       PROPN   NNP     Number=Sing     5       nsubj   5:nsubj _       _       ARG0    _
2       Bush    Bush    PROPN   NNP     Number=Sing     1       flat    1:flat  _       _       _       _
3       on      on      ADP     IN      _       4       case    4:case  _       _       _       _
4       Tuesday Tuesday PROPN   NNP     Number=Sing     5       obl     5:obl:on        _       _       ARGM-TMP        _
5       nominated       nominate        VERB    VBD     Mood=Ind|Tense=Past|VerbForm=Fin        0       root    0:root  _       nominate.01     V       _
6       two     two     NUM     CD      NumType=Card    7       nummod  7:nummod        _       _       _       _
7       individuals     individual      NOUN    NNS     Number=Plur     5       obj     5:obj   _       _       ARG1    ARG0
8       to      to      PART    TO      _       9       mark    9:mark  _       _       _       _
9       replace replace VERB    VB      VerbForm=Inf    5       advcl   5:advcl:to      _       replace.01      ARG2    V
10      retiring        retire  VERB    VBG     VerbForm=Ger    11      amod    11:amod _       _       _       _
11      jurists jurist  NOUN    NNS     Number=Plur     9       obj     9:obj   _       _       _       ARG1
12      on      on      ADP     IN      _       14      case    14:case _       _       _       _
13      federal federal ADJ     JJ      Degree=Pos      14      amod    14:amod _       _       _       _
14      courts  court   NOUN    NNS     Number=Plur     11      nmod    11:nmod:on      _       _       _       _
15      in      in      ADP     IN      _       18      case    18:case _       _       _       _
16      the     the     DET     DT      Definite=Def|PronType=Art       18      det     18:det  _       _       _       _
17      Washington      Washington      PROPN   NNP     Number=Sing     18      compound        18:compound     _       _       _       _
18      area    area    NOUN    NN      Number=Sing     14      nmod    14:nmod:in      SpaceAfter=No   _       _       _
19      .       .       PUNCT   .       _       5       punct   5:punct _       _       _       _
```

Figure 7.2: Example of *Semantic Role Labelling* of a multi-predicate sentence represented in the conllu format

Verb: disagree
Args: Arg0: Agreer
Arg1: Proposition
Arg2: Disagreed entity
Arg3: Other
Eg: [*Arg0* The people] disagreed [Arg1 with the policies] of the [Arg2 Minister].
[*ArgM-TMP* Usually] [Arg0 Harry] disagrees [Arg2 with Jane] [Arg1 on everything].

Verb: fall
Args: Arg0: Subject/Patient
Arg1: Extent
Arg2: Start point
Arg3: End point
Eg: [*Arg0* The stock-price] fell [Arg2 from $27] [Arg1 to $25].
[Arg0 Their average sale] fell [Arg1 by 4.2%].

Figure 7.3: Examples of Propbank Annotations

7.1.1 SRL Datasets and Label-sets

Most SRL models are trained on a fixed pre-defined set of semantic-labels. **FrameNet** Baker et al. (1998) and **PropBank** Kingsbury and Palmer (2003) are two manually annotated SRL datasets. Both of these datasets also provide a unique set of semantic role-labels which are both derived from distinct linguistic principles. FrameNet and Propbank label-sets are indeed the most widely used within the research community. Section 7.1.1.1 will describe *PropBank* and section 7.1.1.2 will describe *FrameNet* in details.

7.1.1.1 PropBank

The **Proposition Bank** (PropBank) is a publically available dataset of sentences which are manually annotated with the semantic roles. The PropBank dataset is available in a number of languages with the standard semantic role label schema across all the languages. The English PropBank comprises all the sentences in the popular Penn TreeBank corpus Taylor et al. (2003) whereas the Chinese PropBank comprises all the sentences from the Penn Chinese. Subsequently, copra in numerous other languages including it, de, es, hi etc. were annotated with PropBank label scheme and were released publicly by various researchers.
As languages are typologically distinct it is very difficult to define a universal set of

ArgM	Description	Examples
ArgM-TMP	Temporal Argument (when?)	yesterday, tomorrow, next week
ArgM-LOC	Location Argument (where?)	at the market, in Dublin
ArgM-DIR	Direction Argument (where?)	to home? down
ArgM-MNR	Manner Argument (how?)	clearly, with much eagerness
ArgM-CAU	Causal Argument (why?)	due to , in response to
ArgM-REC	Recipient Argument	audiences, each other, him
ArgM-ADV	Miscellaneous	
ArgM-PRD	Secondary Predication Argument	... ate the fruit raw

Table 7.1: Common ArgM labels in the PropBank label set

semantic role labels which are applicable to most of the world's languages. Hence, in the PropBank schema, all the semantic roles are indicted simply by only the numbers rather than names (such as Arg0, Arg1, Arg2 etc.). In general, Arg0 indicates the semantic role PROTO-AGENT, and Arg1 indicate the PROTO-PATIENT role. The semantic-roles of the other labels are less consistent, and vary with the predicate verb. Although, in most cases the Arg2 label indicates the benefactor, instrument, attribute or the end state. Figure 7.3, depicts the examples of sentences annotated with the PropBank labels.

Apart from numbered annotations, PropBank schema also comprises of a number of non-numbered arguments called ***Modifier Arguments (ArgMs)*** (eg: ArgM-TMP, ArgM-LOC, etc.) as these represent roles that aim to modify or adjunct the semantics of a sentence. These labels are indeed consistent across predicates and languages. Table 7.1 lists the most common ArgM type PropBank labels.

7.1.1.2 FrameNet

FrameNet is a research project based at the International Computer Science Institute (ICSI) in Berkeley, California. The project aimed to create a lexical database, based on the linguistic theory of frame-semantics Lakoff et al. (1986). The lexical database also provides a fixed set of arguments for each frame. Hence, the FrameNet labels are more consistent and machine-readable as compared to PropBank, across languages. Consider three example sentences labelled with PropBank annotations, listed as follows.

1. [$_{Arg1}$Twitter-stock price] rose by [$_{Arg2}$6%.]
2. [$_{Arg1}$Twitter-stock price] increased by [$_{Arg2}$6%]
3. There is a [$_{Arg1}$6%] rise in [$_{Arg1}$Twitter-stock price].

Argument	Description
ATTRIBUTE	The scalar property of the item that is changing
. DIFFERENCE	The distance by which the item property changes on the scale.
FINAL STATE	Description of the final state of item after the change on scale
FINAL VALUE	The final value of item property after change.
INITIAL STATE	Description of the initial state of item after the change on scale
INITIAL VALUE	The initial value of item property after change.
ITEM	The description of the item that is experiencing change.
VALUE-RANGE	Range on scale on which the value of the ATTRIBUTE fluctuates

Table 7.2: Example Arguments of the frame **change_position_on_scale** on English Framenet

Frame	Verbs
apply_heat	Cook, roast, boil, barbecue, fry etc.
change_position_on_scale	increase, decrease, rise, fall, went up etc.

Table 7.3: Example frames and their predicate verbs in English FrameNet

All three sentence convey the same information, but consists of three different verb-predicates namely *rise*, *rose* and *increased*. FrameNet assigns all three verbs to a common frame called **change_position_on_scale**. The frame **change_position_on_scale** consists of a fixed set of arguments, some of which are listed in Table 7.2 Table 7.3 provides some example frames and their example verb-predicates.
Hence, SRL with FrameNet labeling involves first identifying frame of the predicate and subsequently identifying arguments of that frame within the sentence. All arguments may not be present in the sentence.

7.1.2 Monoligual Approaches to SRL

The task of SRL typically involves predicting the semantic roles of each word for each predicate within an input sentence. Modern approaches to SRL are supervised ML based approach, that often adopt either FrameNet (section 7.1.1.2) of PropBank (section 7.1.1.1) labeling scheme. This section provides a high level overview of these approaches.

7.1.2.1 Feature based approach

For each predicate in an input sentence, a feature-based semantic role labeller implements following steps:

1. **Pruning**: For any given predicate within a sentence, only a small number of words are arguments while most of the words are to be labelled as *NONE*. Hence, in the very first pruning step, the labeller filters out unlikely words in the sentence using various heuristics.

2. **Identification**: For each word that is filtered from the heuristic based pruning in step 1, the model then runs a binary classifier on it to classify if the specific word is part of an argument or has to be assigned label *NONE*.

3. **Classification**: Finally for words that are filtered from step 2, the labeller runs a multi-class classifier which classifies the argument type of it, out of all possible argument types in PropBank and FrameNet.

Although the multi-class classifier (in step 3) classifies each word's argument type separately, thus making a simplifying assumption that each word can be labeled (as argument of the predicate) independently. However, there is a global constraint on the correct label-sequence, that there should be no overlap between argument types for a single predicate (eg: A verb can not have two subjects in the same sentence).
Hence, for most feature-based labellers, the multi-class classifier generally outputs the probability of each semantic role label being assigned to each word (instead of the most probable label). Subsequently, a dynamic programming algorithm (such as Viterbi algorithm Forney (1973)) is used to find the most probable role-label sequence that satisfies the required constraints.
Popular semantic role labellers such as Gildea and Jurafsky (2002) use numerous hand-crafted features (of the specific word being labelled) to be inputted into the ML based multi-class and binary classifiers. These features include predicate-word, head-word, tense, POS-tag, linear position of word phrase-type etc.

7.1.2.2 Neural based approach

State-of-the-art approaches to SRL are neural-network based approaches, that treat the SRL ask as a sequence labelling task. Generally, neural-network approaches use the BIO labelling scheme to label all words within the sentence simultaneously (for each predicate independently though). In the BIO scheme Ratinov and Roth (2009)

each tag is augmented with *Begin*, *Intermediate* or *End* indicators indicating the start and end of an argument (if its multi-word).
Most common architectures for sequence-tagging tasks are Bi-LSTM based architectures, which is widely adopted by NN based approaches to SRL such as Shi and Lin (2019); He et al. (2017); Zhang et al. (2019). However state-of-the-art approaches have also adopted transformer based architecture as it has become widely popular for other similar sequence-tagging task Mohammadshahi and Henderson (2021) such as POS-tagging, NER tagging.

7.2 Cross-lingual Approaches to SRL

Similar to other NLP tasks, as already explained, the state of the art approaches to SRL (section 7.1.2) are supervised approaches which require large annotated datasets to be trained on thus limiting their utility to only high-resource languages. This issue of data-sparsity (in low-resource languages) has been effectively addressed with numerous cross-lingual approaches to SRL including *Annotation Projection* approaches such as Padó and Lapata (2009); Kozhevnikov and Titov (2013); Akbik et al. (2015); Aminian et al. (2019), *Model Transfer* approaches such as McDonald and Nivre (2013); Swayamdipta et al. (2016); Daza and Frank (2019); Cai and Lapata (2020) and the *Machine Translation* approaches such as Fei et al. (2020).
In this work, we provide an overview of **Valency Patterns Leipzig (ValPal) online database**[1] Hartmann et al. (2013) which is a multilingual lexical database, originally created by the linguistic research community to study the similarities and differences in verb-patterns for various world's languages. Furthermore, we provide a framework to utilise the knowledge available in **Valpal** database to improve the performance of the state-of-the-art cross-lingual approach to SRL task.

7.3 ValPal Database

Valency Patterns Leipzig (ValPal) is a comprehensive multilingual lexical database which provides semantic and syntactic information about different verb-forms in various languages, including many low-resource languages. The ValPal database provides values of following features for each verb-form:

1. *Valency:* the total number of arguments that a base verb-form can take.

[1] http://ValPal.info/

2. *Argument-pattern*: the type and order of arguments taken by a base verb-form in its most common usage.

3. *Alterations:* the alternate *argument-patterns* that can be taken by either the base verb-form or any of its morphological variant.

Table 7.4 depicts the information about three lexical units namely ***cook***, ***kochen*** and ***cuocere*** as provided in the ValPal database. Please note that Table 7.4 lists only a few of all the alterations provided for these verb-forms in ValPal database due to space constraints. Lexical units ***cook***, ***kochen*** and ***cuocere*** are *en*, *de* and *it* words representing base verb-form for verb activity COOKING.

7.3.1 Coding of Argument-patterns

In ValPal database each argument-pattern (including alteration) is coded with a unique coding-frame. For example in Table 7.4, the argument-pattern of English base verb-form ***cook***, is coded as follows

$$1 - nom > V.subj[1] > 2 - acc$$

The code indicates that the base verb-form ***cook*** takes 2 arguments in its most common usage (valency of 2). The first argument is *cooker* (indicated as *1-nom*) and the second one is *Cooked-food* (indicated as *2-acc*). $V.subj[1]$ indicates the verb with the first argument as its agent. The order of arguments are **cooker–V–cooked_food** (eg: She is cooking the fish.).
Verb-form ***cook*** also has an *alteration* called **Causative-Inchoative** with the derived argument-pattern as follows.

$$2 - acc > V.subj[1]$$

This argument pattern indicates that verb-form can also have order of arguments as **cooked_food–V** with *Agent* argument missing from the sentence (eg: The fish is cooking).

7.3.2 Coding-sets

ValPal provides a unique coding-set for each language. The codes in these coding sets indicate various argument-types including modifier argument-types. For example, codes NP-Nom, NP-acc and LOC-NP indicate the AGENT (Arg0), PATIENT (Arg1) and modifier LOCATION (ArgM-LOC) arguments respectively in the coding-sets

of all languages. The codes with+NP and mit+NP-dat indicate INSTRUMENT argument in English and German coding-sets. Similarly codes UTT-NP indicate the argument TEMPORAL in most coding-sets. In these codes, the *NP* indicates the index of valency occupied the respective argument within the argument pattern (eg: code $2 - acc$ in argument pattern $2 - acc > V.subj[1]$ indicates argument-type PATIENT with the valancy-index of 2).

7.3.3 Alteration Types

As already explained, the ValPal database also provides a list of alternate argument-patterns (called alterations) for each verb-form. Some of these alterations are *morpho-independent* as they can be taken by the respective base-verb in any morphological form, whereas others are *morpho-dependent* as they can be taken by the respective verb only in a specific morphological form.
For example, both the *Reflexive-Passive* and *Impersonal Passive* alterations of the italian base verb-form *cuocere*, outlined in Table 7.4 are morpho-dependent alterations as these alterations are observed only when the verb-form possesses morpheme si.
The ValPal database is originally created by the linguistic research community, typically to study the similarities and differences in verb-patterns for various world languages. However this knowledge can also be used by NLP research community for building the models for data-sparse languages.

7.3.4 FrameNet to aid ValPal

One shortcoming of the Valpal database is that its vocabulary is limited for many languages. If we encounter a verb in the training-set that is missing in ValPal, we utilised the *FrameNet* database to extract the desired *argument-pattern* and *alterations* of it from ValPal itself.
To extract this knowledge about the missing verb, firstly we extracted the frame of the missing verb from the respective *FrameNet* database. Subsequently, we extracted a replacement-verb that belongs to the same frame (as that of the missing verb) and is available in ValPal database. Finally, we assigned the argument-pattern and alterations of this replacement-verb to the missing verb. For example, the verb **barbecue** is missing from the ValPal database. Yet, the verb **barbecue** belongs to frame **COOKING-45.1** in *English FrameNet* Barkley. Another verb-form called **cook** belong to the same frame (**COOKING-45.1**) and is available in ValPal database. Thus

Verb-form	**Lang**	**Argument-pattern**	**Alterations** (Alteration-name:Arg-pattern (example))
cook	English	$1 - nom > V.subj[1] > 2 - acc$	***Understood Omitted Object***:$1 - nom > V.subj[1] > 2 - acc$ (She walked in while I was cooking.) ***Causative-Inchoative*** : $2 - acc > V.subj[1]$ (The soup is still cooking.)
kochen	German	$1 - nom > V.subj[1] > 2 - acc$	***Benefactive Alternation***:$1 - nom > V' > subj[1] > 3 - dat > 2 - acc$ (Ich koche meiner Mutter eine Suppe.) ***be-Alternation***:$1 - nom > beV'.subj[1] > 4 - acc > mit + 2 - dat$ (Die Großmutter bekocht die Kranke mit Suppe.) ***Ambitransitive Alternation***:$2 - nom > V'.subj[2]$ (Das Wasser kocht.)
cuocere	Italian	$1 > V.subj[1] > 2$	***Reflexive-Passive***:$2 > siV'.subj[2] > daParteDi + 1$ (La carne si cuoce con attenzione.) ***Impersonal Passive***:$siPassV' > da + 1$ (Quando si è (stati) cotti dal sole si diventa di color rosso intenso.)

Table 7.4: Sample verb-form knowledge in Valpal database

we use argument-patters provided in ValPal for verb-form **cook** as the argument-patterns for **barbecue**.

7.4 FOL rules from ValPal

To inject the entire ValPal database knowledge about any low-resource target-language l in a Cross-lingual Neural Network model, we represented this knowledge as a set of First-order-logic (FOL) rules F_l. The process of generating this set of FOL rules involves two steps namely *Translating ValPal Argument-patterns to Propbank label orders* and *Writing Propbank-label order as FOL rule* described as sections 7.4.1 and 7.4.2.

In ValPal database, the argument-pattern for verb-form ***tie*** is outlined as equation

7.1 (as Q).

$$Q = 1 - nom > V.subj[1] > 2 - acc > LOC - 3(> with + 4) \tag{7.1}$$

We use this as an example to demonstrate the process of converting an argument-pattern to a FOL rule.

7.4.1 Translate argument-patters to Propbank Order

In this step, we translate all the Valpal's argument-patterns (including alterations) for all lexical verb-forms in any target-language l, to the Propbank Orders. The entire process of translating a ValPal argument-pattern P of the language l into a *Propbank Label-order* involves two simple text-processing sub-steps described as sections 7.4.1.1 and 7.4.2.

7.4.1.1 Replace modifier argument-types

As already explained in section 7.3.2, Valpal database provides a unique coding-set for each language. In this subset, we examined the entire coding-set for language l to identify the codes that refer to a modifier argument-type (eg: LOC-NP and UTT-NP etc. in English coding-set for LOCATION and TEMPORAL modifier-arguments), and created a mapping table that maps these modifier-argument codes to the corresponding Propbank annotations (eg: LOC-NP mapped to ARGM-LOC; UTT-NP mapped to ARGM-TMP etc.). The coding-set of any language in the ValPal database is small thus making it feasible to manually create such a mapping table.
Subsequently, we used this mapping table to replace all modifier argument-patterns (if any) in the argument-pattern P being translated, with the corresponding Propbank label.
After replacing the modifier argument-types we reduce the valancy-index of all the arguments following the replaced modifier argument, in the argument-pattern being translated, by one. For example, the argument-pattern outlined as equation 7.1 comprises only one modifier argument-type namely LOC3.
We replaced this with the corresponding Propbank label namely ARGM-LOC and reduced the valency-index of all argument-types following this replaced argument-pattern by 1 (thus $(with+4)$ is re-written as $(with+3)$). Hence the argument-pattern in Equation 1 would be re-written as equation 7.2.

$$Q = 1 - nom > V.subj[1] > 2 - acc > ARGM - LOC(> with + 3) \tag{7.2}$$

7.4.1.2 Rewrite all non-modifier argument types

After replacing all modifier argument-types in the argument-patterns by the process described in section 7.4.1.1, we simply replaced all left over arguments in the ValPal argument-pattern P as 'ARGx' where x is $valancyIndex-1$. Hence argument $1-nom$, $2-acc$ and $with+3$ (with valancy Indexes as 1, 2, 3 respectively) in equation 7.2 would be replaced by *Arg0*, *Arg1* and *Arg2* respectively.
Finally we replaced $Vsubj[NP]$ with V and removed all bracket symbols. Hence argument-pattern outlined as equation 7.2 would be translated as following equation 7.3.

$$Q = ARG0 > V > ARG1 > ARG - LOC > ARG2 \tag{7.3}$$

7.4.2 Write Propbank Label order as FOL rule

Having represented all argument-patterns (including alterations) for all lexical verb-forms of language l as allowed Propbank Label-orders, we rewrite each verb-form and Propbank Label-order pair as a FOL rule. For example the pair of verb-form ***tie*** and its corresponding allowed Propbank Label-order outlined as equation 7.3, is represented by the FOL rule indicated as the following equation 7.4.

$$f = baseForm(V, tie) \vee pattern(Y, Q) \tag{7.4}$$

Here Q is the Propbank label-order outlined in equation 3, and Y is the sequence of Propbank tag-sequence predicted by a neural-network model for any input token-seq. The logic-constraint in equation 7.4 would be true if the verb for which the arguments are being predicted is a variant of base verb-form *tie* and the predicted SRL tag sequence Y satisfies the label order Q
While checking whether a predicted SRL tag sequence follows a specific order, we ignore the **'O'** annotations (**'O'** indicates semantic role label **'NULL'** in the Propbank Annotation scheme). For example, the SRL tag sequences ***ARG0, ARG0, O, O, V, ARG1, ARG-LOC, O, ARG2*** follows the argument-pattern.
To check if the verb for which the arguments are being predicted is a morphological variant of the specific base verb-form, we perform stemming of both base verb-form and the token from sentence which is tagged 'V' by the model. If the stem strings are equal we consider the verb token to be a variant of base verb-form.
If an argument-pattern (represented as Propbank label-order) is for a morpho-dependent

alteration, then the morphological constraint is also added to the FOL rule representing the argument-pattern. For example, in Table 7.4 the argument-pattern *Reflexive-Passive* is a morpho-dependent alteration. This argument-pattern is represented as FOL defined by equation 7.5.

$$f = baseForm(V, cuocere) \vee morphoForm(V, si) \vee pattern(Y, \hat{Q}) \tag{7.5}$$

Here $\hat{Q}$ represents the corresponding label-sequence for Argument-pattern. The rule $morphoForm(V, si)$ constraints the verb V to have morpheme si for the rule to be true.

Hence we obtain a set of FOL rules F_l representing the entire Valpal database knowledge about language l (with each verb-form and argument-patterns pair provided in the Valpal database for the language l as a single FOL-rule $f \in F_l$). These FOL rules are used during the fine-tuning of a cross-lingual neural-network model for SRL in target-language l. During fine-tuning, the model is always rewarded if it predicts an SRL tag-seq Y which satisfies atleast one of the FOL rule $f \in F_l$, and penalised otherwise. Section 7.5.1 will explain the fine-tuning process in more detail.

7.5 Model

7.5.1 Labeler fine-tuning with ValPal

This section describes the framework adopted by us to induce the target-language specific ValPal database knowledge expressed as a set of FOL rules F_l, into the pre-trained *Semantic Role Labeler*. Our framework is inspired by the *Deep Probabilistic Logic* (DPL) framework proposed by Wang and Poon (2018). The framework assumes the availability of only an unlabelled target-language corpus. Hence, for the *Labeler fine-tuning* sub-step, we randomly sample a batch from the already available parallel source-target data and utilised only the target language part of it.

Let $X = x_1.....x_T$ be an input sentence and $Y = y_1.....y_T$ be any SRL-tag sequence. Further, let Ψ be the pre-trained Bi-LSTM based *Semantic Role Labeler* such that $\Psi(X, Y)$ denotes the conditional probability $P(Y|X)$ as outputted by the final softmax layer of Ψ.

The fine-tuning of this pre-trained Ψ to specific target-language l requires an unlabelled target-language training corpus. Given such an unlabelled target-language-corpus X_{targ}, for each $X \in X_{targ}$ we input sentence X into the pre-trained Ψ to compute the most probable SRL-tag sequence Y as $Y = argmax_{\hat{Y}}(\Psi(x, \hat{Y}))$. Subsequently we input both the sentence X and its predicted most-probable SRL tag-seq

Y in all the FOL rules in F_l to compute their value (as 0.0 or 1.0). DPL framework defines the conditional probability distribution $P(F_l, Y|X)$ as equation 7.6.

$$P(F_l, Y|X) = \prod_{f \in F_l} \frac{exp(w.f(X,Y)).\Psi(X,Y)}{exp(w)} \tag{7.6}$$

The framework assumes the Knowledge-constraints to be log-linear and thus defines each knowledge-constraint as $exp(w.f(X,Y))$ where $f \in F_l$ is the FOL rule representing the respective knowledge-constraint. Here w is the pre-decided reward-weight assigned to all constraints. Hence the predicted output-sequence Y would be rewarded (as its likelihood would increase by a factor of $exp(w)$) if it follows one of the defined argument-patterns in ValPal database for the respective verb for which the arguments are being predicted ($f(X,Y) = 1.0$). However no penalty is awarded for not following the correct Argument-pattern.

7.5.1.1 Learning

The ideal way to optimize the weights (fine-tune) of the model Ψ is by minimizing $P(F_l|X)$ and updating the parameters through back-propagation. We can compute $P(F_l|X)$ by summing over all possible SRL-tag sequences as $P(F_l|X) = \Sigma_Y P(F_l, Y|X)$. However computing $P(F_l, Y|X)$ by equation 7.6 with all possible output-sequences, and subsequently back-propagating through it, for each training example is computationally very inexpensive. Thus DPL framework also provides a more efficient EM-based approach Moon (1996) to the parameter fine-tuning which is adopted by us.

The full process of learning the parameters of Ψ (initialized with parameters pre-trained on source language) is outlined as Algorithm 2. For each training-example $X \in X_{targ}$, the Algorithm 2 implements three steps. In the first-step, it predicts the most probable SRL-tag sequence Y for the given training-example X as $Y = argmax_{\hat{Y}}(\Psi(x, \hat{Y}))$ with current parameter values for Ψ.

In the E-step, it compute $q(Y) = P(F_l, Y|X)$ by applying equation 4 with current parameters of Ψ. Finally in M-step it keeps $q(Y)$ as fixed and update parameters of Ψ by minimizing the KL-divergence Kullback and Leibler (1951) loss between $q(Y)$ and the probability of Y from $\Psi(X,Y)$ (i.e. $P(Y|X)$).

In other words, in each epoch step the model first computes the joint likelihood of F_l and Y i.e $P(F_l, Y|X)$ with current model parameters, and subsequently it updates the parameters to predict likelihood of Y i.e., to be as close to $P(F_l, Y|X)$ as possible.

Algorithm 2 Fine-tuning of Semantic Role Labeller

Require: Target Language corpus X_{targ}; set of FOL rules F_l representing entire Valpal db knowledge; Pre-trained LSTM based Semantic Role Labeller Ψ; Number of Epochs N;

repeat

for each $X \in X_{targ}$ **do**

▷ **E-Step**

$Y \leftarrow argmax_{\hat{Y}}(\Psi(X, \hat{Y}))$

$q(Y) \leftarrow P(F_l, Y|X)$ ▷ by equation 7.6

▷ **M-Step**

$\Psi \leftarrow argmin_{\hat{\Psi}}(D_{KL}(q(Y)||\hat{\Psi}(X, Y)))$

end for

until convergence

7.6 Experiments

This section described the experiments performed by us to evaluate the proposed model. Subsequently, section 7.7 will discuss the results achieved. These experiments aim to address following novel objectives:

1. To evaluate the impact of injection of VALPAL database knowledge on the performance of a cross-lingual simple Bi-LSTM based model for the SRL task.

2. To evaluate the impact of Polyglot and Few-shot training on the performance of the proposed BiLSTM based Semantic Role Labeller with VALPAL typology knowledge.

3. To evaluate the impact of VALPAL vocabulary expansion with FrameNet databse, on the performance of the proposed BiLSTM based Semantic Role Labeller with VALPAL typology knowledge.

7.6.1 Dataset

We experimented with four languages namely *en*, *de*, *zh* and *it* as these languages are covered in both the ValPal database as well as in the CoNLL 2009 Shared task Hajic et al. (2009) dataset. The *Semantic Role Labeller* requires a fully-annotated training dataset in the high-resource source-language. We utilized the Universal Proposition Banks provided at [2] provided for CoNLL 2009 Shared task, for training of the *Semantic Role Labeller* and the evaluation of various systems.

[2] `https://github.com/System-T/UniversalPropositions`

Hyper-parameter	Value
Dropout prob.	0.01
Bach-size	32
Epochs	150
embeddings size	768
predicate indicator embed size	16
Bi-LSTM hidden states size	400
BiLSTM depth	3
hidden biaffine scorer size	300
Bi-LSTM hidden states size	256
BiLSTM depth	2
compressed role rep size	30
hidden biaffine scorer size	30

Table 7.5: Hyper-parameter settings for input and training (first block), semantic role labeler (second block) and semantic role compressor (third block). Semantic role labeller and Semantic role compressor are same as Cai and Lapata (2020)

On the other hand, the *Semantic Role Compressor* component requires sentence-paired parallel copra in source and target languages. We used the Europarl parallel text-corpus Koehn et al. (2005), and the large-scale EN-ZH parallel corpus Xu (2019) to train the *Semantic Role Compressor*, as used by Cai and Lapata (2020).
We used the target-language part of the same parallel-corpora independently for the Valpal knowledge induction, as the Valpal database knowledge induction simply requires unlabelled text-corpus in the target-language.

7.6.2 Model-configurations

We computed the language-independent BERT-Embeddings to be fed into the networks using pre-trained Multilingual BERT (mBERT) Wu and Dredze (2019) model. These embeddings are calculated in same way as computed in the work of Kondratyuk and Straka (2019a). Given a sentence S, we tokenised the whole sentence using *WordPiece tokeniser* Wu et al. (2016). Subsequently, we fed this token-sequence into pre-trained mBERT provided by *HuggingFace* [3].
Embedding of any word $w \in S$ i.e. e_w is computed by taking average of mBERT outputs of all Wordpiece tokens corresponding to word w. Subsequently, these word-embeddings are frozen during the training of the networks. Table 7.5 outlines the hyper-parameters used during training.

[3] `https://huggingface.co/bert-base-multilingual-cased`

Algorithm 3 Full Training process. The function *FineTune* represents the process outlined as 2 and function *CrossTrain* represents the cross-lingual training procedure adopted by Cai and Lapata (2020). L_{CE} is cross-entropy loss and L_{KL} is KL divergence loss

Require: Annotated Source language corpus $\{X_{Tagged}, Y_{Tagged}\}$; Parallel Source-target Corpus $\{X^S_{Parallel}, X^T_{Parallel}\}$; set of FOL rules representing entire Valpal db knowledge of target language F_l; batch-size b; Number of Epochs E Semantic Role Labeler Ψ; Semantic Role Compressor Φ

$steps \leftarrow |X_{Tg}|/b$
for $epoch \leftarrow 1$ to E **do**
  for $step \leftarrow 1$ to $steps$ **do**
    $X, Y \leftarrow$ Sample($\{X_{Tg}, Y_{Tg}\}$,b)
    $X^S, X^T \leftarrow$ Sample($\{X^S_{Pr}, X^T_{Pr}\}$,b)
    ▷ **Labeler pre-training**
    $\Psi \leftarrow argmin_{\hat{\Psi}}(D_{CE}(Y||\hat{\Psi}(X)))$
    ▷ **Labeler Fine-tuning**
    $\Psi \leftarrow FineTune(X^T, F_L, \Psi, b)$
    ▷ **Compressor training**
    $\Phi \leftarrow argmin_{\hat{\Phi}}(D_{KL}(\Psi(X)||\hat{\Phi}(X)))$
    ▷ **Cross-lingual training**
    $\Phi, \Psi \leftarrow CrossTrain(X^S, X^T, \Psi, \Psi)$
  end for
end for

7.6.3 Baselines

We compared the performance of our proposed model against the base-model Cai and Lapata (2020) as well as numerous other state-of-the-art baselines. These baselines include two annotation projection based models namely *Bootstrap* Aminian et al. (2017) and *CModel* Aminian et al. (2019), as well as two strong mixture-of-experts models namely *MOE* Guo et al. (2018) which focus on combining language specific features automatically as well as *MAN-MOE* Chen et al. (2018) which learns language-invariant features with the multi-nominal adversarial network as a shared feature extractor. We also compared with PGN Fei et al. (2020) which is the state-of-the-art translation-based model which translates the source annotated corpus into the target language, performs annotation projection, and subsequently trains the model on both source and the translated corpus. We utilised the source-code provided by the authors of each of these baselines to train and test them.

Model	it	de	zh	avg
Bootstrap	51.7	55.2	58.4	55.1
CModel	55.5	57.0	61.1	57.9
MAN-MOE	57.1	64.0	64.7	61.9
MoE	56.7	63.2	65.2	61.7
PGN	57.9	65.3	65.9	63.0
Base-wo-Compressor	37.1	49.7	45.3	44.0
Base-wo-Compressor+ Valpal	37.8	54.2	49.9	47.3
Increase	0.7	4.5	4.6	3.3
Base-full	57.2	65.1	68.8	63.7
Base-full+ Valpal	**57.9**	**69.5**	**73.4**	**66.9**
Increase	0.7	4.4	4.6	3.2

Table 7.6: Results for Monoloingual settings (with extended vocab for de and zh)

Model	it	de	zh	en	avg
MAN-MOE	57.7	66.2	65.9	66.0	63.9
MoE	57.1	63.5	66.1	64.1	62.7
PGN	**58.0**	**65.7**	**66.9**	**67.8**	**64.6**
Base-wo-Compressor	37.6	50.2	48.9	49.9	46.6
Base-wo-Compressor + Valpal	38.5	54.7	53.6	54.8	50.4
Increase	0.9	4.5	4.7	4.9	3.8

Table 7.7: Results in Polygot settings

7.7 Results

7.7.1 Monolingual training

In the first set of experiments we trained the models on a single source language *en* and tested these on the target languages *zh*, *it* and *de*. In these settings, we trained the models on English UPB train-dataset and tested them on the UPB test-sets of the target-languages. Table 7.6 shows the labeled F-scores achieved on each of these target-languages. In Table 7.6, the *Base-wo-Compressor* refers to the base model without the SRL compressor, whereas *Base-full* refers to the full base model.

7.7.2 Polyglot training

Table 7.7, outlines the results obtained under the polyglot training settings. For each experiment within these settings, the models are trained on a joint polyglot corpus of the three out of four languages namely *en*, *it*, *de* and *zh*, excluding the target language for which the results are outlined. For each experiment within these settings, the training corpus size is always fixed to 600,000 tokens to ensure controlled

	it	de	zh
Vocab	125	128	122
Ext-vocab	–	975	415
Base-full	57.2	65.1	68.8
Base-full+ ValPal	57.9	65.9	68.7
Increase	0.7	0.8	0.9
Base-full+ ValPal-ext	–	69.5	73.4
Increase	0.7	4.4	4.6

Table 7.8: Results with and without ext-vocab

experiment-settings.
We created such a polyglot corpus by randomly sampling sentences from UPB train-set for each of the three source-languages until the token-size becomes approximately equal 100,000, concatenated all these sampled datasets and randomly shuffled the order. *Alignment-projection* based approaches and the *Base-full* are not evaluated in the polyglot settings as these approaches require parallel-aligned source and target language sentence-pairs.

7.8 Analysis

In this section we analyse the results outlined in section 3.9 to address the research questions **RQ11**, **RQ12** and **RQ13** listed in section 1.1.1 as follows.

RQ11: Does the performance of a simple BiLSTM model for the Semantic Role Labelling task improve, when the semantic typology knowledge of the target-language available in the ValPal database, is injected into it, within monolingual and polyglot training training scenerio ?

RQ12: Does the impact of injecting the ValPal database knowledge into the state-of-the-art cross-lingual BiLSTM based model for the Semantic Role Labelling task increases due to joint polyglot training as compared to the mono-lingual training ?

Results in Table 7.6, show that for both *Base-wo-Compressor* and *Base-full* model, adding Valpal database knowledge improved the performance of it for all three target languages. Furthermore, for all three target-languages, the improvement in performance of both *Base-wo-Compressor* and *Base-full* models due to Valpal knowledge

injection are same i.e 0.7 for it, 4.4 for *de* and 4.6 for *zh* (average 3.2). This provides the evidence that the improvement is indeed due to the Valpal Knowledge injection. Similar trends are observed for polyglot training Results show that adding Valpal knowledge improves the performance of *Base-wo-Compressor* model, even within the polyglot settings, Furthermore, it is observed that although *Base-wo-Compressor* model performs better in the polyglot training settings as compared to monolingual settings for most of the target languages, the improvement in performance of *Base-wo-Compressor* due to Valpal knowledge injection is the same in both settings. This is because the fine-tuning of model with Valpal database knowledge is performed only with the unlabelled target-language corpus.

RQ13: Does extending the verb-inventory of ValPal database for a specific target-language with other lexical databases (such as FrameNet and VerbNet) before injecting this ValPal knowledge into the cross-lingual BiLSTM based model for the Semantic Role Labelling, increases the impact of this knowledge injection ?

It can be observed in Tables 7.6 and 7.7 that the improvement on target-language it is much lower than the improvements observed on *zh*, *de* and *en*. The reason being that we extended the Valpal vocabulary of en, zh and de using English Framenet Baker et al. (1998), Chinese Framenet You and Liu (2005) and German Framenet Burchardt et al. (2009) by the process described in section 7.3.4. However the Italian Framenet is not publicly available.
We indeed performed experiments to analyze the impact of vocabulary extension on the performances. Table 7.8 outlines the results of these experiments. It can be observed in the table that extending the vocabulary of Valpal with the Framenet does lead to significant improvement in performance.

7.9 Conclusion

Valency Patterns Leipzig (ValPal) is a multilingual lexical database which provides the knowledge about the argument-patterns of various verb-forms in multiple languages including numerous low-resource languages. The database is originally created by the linguistic community to study the similarities and differences in the verb-patterns for various world's languages. In this work we utilised this database to improve the performance of the state-of-the-art cross-lingual model for SRL task.

We proposed and evaluated a novel framework to integrate the entire Valpal knowledge about any low-resource target-language into a state-of-the-art cross-lingual LSTM based model for SRL task. Our proposed framework only requires an unannotated target language corpus for the knowledge integration. Our results showed that VALPAL database knowledge injection does lead to significant improvement in performance. Furthermore, we extended the vocabulary of ValPal database with FrameNet database, to improve the performance even more.

As far as we are aware, this is the first work that aimed to integrate the semantic typology knowledge available in an external database into a neural-network model. We evaluated the impact of this knowledge injection on the performance of the model under various cross-lingual settings.

Chapter 8

Conclusion

State-of-the-art neural-network based approaches to most NLP tasks have achieved near human performance. However, these are supervised approaches that require large annotated datasets to be trained on. This limits their utility to only a few high-resource languages for which such dataset is available. Manually building such dataset is tedious, difficult and very expensive. Furthermore, the large scope of languages and the tasks makes complete coverage infeasible.

Cross-lingual Transfer-learning (CLT) based approaches are distinct class of approaches proposed by the researchers to address this issue of data-sparsity. A typical CLT based approach involves training a model on a high-resource source-language and is applied on a low-resource target-language. The CLT based approach represents the input text using either delexicalized features or cross-lingual/multilingual features to make the cross-lingual transferring from source to target language possible.

State-of-the-art CLT based approaches to most NLP tasks significantly outperform unsupervised approaches. The CLT based approaches perform almost at par with the fully supervised approaches if the source and target languages are genealogically and typologically close to each other whereas the performance drops significantly if the source and target languages are genealogically and typologically apart. In this project we used linguistic typology knowledge to address this issue.

Linguistic typology is the branch of linguistics that aims to classify all human languages based on their phonological, syntactic and phonological properties. These properties are represented as the values of numerous linguistic typology features. Hence, there are numerous publicly available typology databases that provides sets of typology-features and their possible values, as well as respective feature values for various languages. These databases are created by linguistics over decades primarily to study the similarities and distinctions among world's languages. However, in this work we successfully utilised the knowledge available in such typology databases to

improve the cross-lingual transferring ability of the CLT based models, specifically in scenarios where source and target languages are genealogically and typologically apart.
Our research project is a wide-scope project. In this project we experimented with numerous typology databases, and four key NLP intermediate NLP tasks namely *Constituency Parsing*, *Dependency Parsing*, *Enhanced Dependency Parsing* and *Semantic Role Labelling*.
In section 8.1 we provide a chapter-wise summary of our research work. In section 8.2 we outline the overall inferences about the cross-lingual transfer-learning with linguistic typology knowledge that we observed across various tasks. In section 8.3 we outline the key limitations of linguistic typology knowledge with cross-lingual NLP and in 8.4 we provide directions for future research.

8.1 Chapter-wise Summary

This section provides chapter-wise summary of key contributions made in the field of cross-lingual NLP through our entire research-project.

Chapter 2: This chapter provided a very detailed literature review of the field of various significant unsupervised and cross-lingual approaches to numerous NLP tasks, applicable within the low-resource scenarios. The chapter also reviewed linguistic-typology field including various open-source linguistic-typology databases. Finally, the chapter described previously published CLT based approaches that indeed used linguistic typology knowledge to improve their performance.

Chapter 3: This chapter provided a brief overview of the Constituency Parsing task including the Constituency Grammar schema. Subsequently, the chapter proposed the *Universal Recurrant Neural Network Grammar* (UniRNNG) model which is a cross-lingual neural network based constituency parser which utilises the linguistic typology knowledge available in WALS database to improve cross-lingual transferring. We evaluated the proposed model on a selected set of source-target pairs in both *few-shot* and *zero-shot* learning scenarios. We also evaluated the effect of polyglot training on the performance of our proposed UniRNNG model. Results obtained showed that feeding the linguistic typology knowledge available in WALS database into a cross-lingual RNNG parser does led to improvement in performance for language pairs comprising of source and target languages that are typologically and

geographically apart.

Chapter 4: This chapter proposed a multitasking model to predict numerous WALS typology features including phonological, semantic and features for various languages, as a solution to the *SigTyp 2020 Shared Task*. The proposed model was trained and evaluated on the train-test dataset provided as part of the shared task. The model was evaluated using the evaluation script provided by SigTyp 2020 indeed. The results showed that the proposed model performed at par with other complex solutions. This model inspired our proposed multitasking end-to-end dependency parsers and enhanced dependency parser described in chapters 5 and 6 respectively.

Chapter 5: This chapter described the dependency parsing framework and proposed and evaluated an *End-to-end BERT based model* for cross-lingual Dependency Parsing task. The proposed model injects linguistic typology knowledge in URIEL database by predicting the typology feature values for the target language being parsed as an auxiliary task within multitasking settings. Similar to the CP task, we evaluated the proposed model in both few-shot and zero-shot settings. Results indeed showed significant improvement in performance of due to the auxiliary task.
In the same chapter we also successfully improved the performance of the state-of-the-art *UDify* model for cross-lingual DP by adding the same auxiliary task of URIEL feature prediction.

Chapter 6: This chapter described the Enhanced Dependency parsing framework in detailed. Subsequently, inspired by the End-to-end BERT based dependency parser and UDify model described in Chapter 4, we proposed a *Multitasking End-to-end BERT based model for Enhanced Dependency* parsing. We proposed this model as part of *SigParse 2021 Shared Task*. Our proposed model performed at par with complex state-of-the-art models while being much simpler in design. Furthermore, in this chapter we also proved that injecting linguistic typology knowledge improves the performance of the proposed model for EDP task for most target languages.

Chapter 7: This chapter described the Semantic Role Labelling task including a review on monolingual and cross-lingual approaches to SRL. The chapter described a state-of-the-art BiLSTM based cross-lingual model for the SRL task. Subsequently, we provided a framework for induction of the semantic typology knowledge available in *Valency Patterns Leipzig (ValPal)* database. The proposed framework is built

on the Deep Probabilistic Logic framework for injecting first-order-logic rules into a neural network model. The results showed that such knowledge induction led to significant improvement in performance of the model.

8.2 Overall trends

As already described, our work is a wide-scope PhD where we experimented with four key distinct NLP tasks Constituency Parsing, Dependency Parsing, Enhanced Dependency Parsing and Semantic Role labelling. For each task, we evaluated and compared the performances of a state-of-the-art cross-lingual transfer-learning model of it, under numerous cross-lingual training and testing scenarios, such as *Mixed Polyglot vs single Source* training scenarios, *Few-shot vs Zero-shot* learning scenarios, Scenarios where Source and Target languages belong to same vs distant linguistic families etc.
Furthermore for each task, we proposed a framework to inject linguistic typology knowledge into a state-of-the-art model, and evaluated the impact of such typology knowledge injection within all the cross-lingual training and testing scenarios described previously.
In this section, we describe the key trends about cross-lingual transfer-learning and the impact of typology knowledge injection, that we observed across all four tasks.

- **Cross-lingual transfer learning performs better when source and target languages are typologically similar as compared to when they are typologically apart.** Specifically, we observed that models show strong performance when source and target languages belong to same linguistic family. For example, for all tasks we observed that a cross-lingual model trained on a single source-language *English* shows strong performance on the target languages *Danish* and *German*, but shows poor performance of *Chinese*.
  This observed trend is inline with the trends observed by most of the other researchers 2.1.1.1. However, some modern works Bommasani et al. (2021) claim that in the current large language model based approaches, the impact of the typology distance between source and target languages is comparatively less on the performances of the respective models. For example, in XTREME Hu et al. (2020) and IGLUE Bugliarello et al. (2022) benchmarks, cross-lingual transfer learning from Spanish to Chinese yields higher performance than from Spanish to Friulan.

- **Mixed polyglot training performs better that Monolingual training, specifically when the target language is apart from all the source languages.** If there are multiple source languages and a target language which is distinct from all the source languages, we observed for all the tasks that the mixed polyglot training on all the source languages leads to better performance than monolingual training on each of the source languages.

- **Typology knowledge induction leads to better performance within *Few-shot* learning settings as compared to *Zero-shot* learning.** For all the tasks, we observed that the performance of cross-lingual models improves significantly even when a handful of target languages are added to the training corpus along with most of source languages. This is observed in both *Mixed Training* and *Single Source Training* scenarios.

- **Typology knowledge induction leads to larger improvement in performance within Mixed Polyglot training scenario as compared to Monolingual training scenario.** We observed that same trend for all tasks. Key reason behind this would be that *Mixed polyglot* corpus training would lead to better generalization of models over a varied range of typology feature-values.

- **Typology knowledge induction leads to larger improvement in performance when the target languages are typologically distinct from source languages.** We observed this for all syntactic tasks.

8.3 Drawbacks of Typology knowledge Induction

In our work we aimed to improve the performances of cross-lingual transfer-learning models for various NLP tasks with typology knowledge induction. Although, we did achieve significant improvement in performances for all the tasks under various scenarios, yet there are several reasons that we observed in our work, that could limit the use of linguistic typology knowledge for a wide range of tasks and languages in the future work. In this section, we list these issues.

1. **Granularity:** Most typology databases assign a single value for each typology feature for a specific language. However, in most languages a single typology feature takes multiple feature-values depending upon the context. Thus injecting the linguistic typology knowledge with a single fixed value for each typology feature, confuses the model thereby dropping the performance.

We observed that for syntactic tasks, the granularity issue is very prominently in free word-order languages. For example for constituency parsing task, the typology knowledge injection leads to drop in performance for target language Russian.

For dependency parsing task, we aimed to reduce this issue by injecting the typology knowledge with multiple weighted feature-values for each typology feature (instead of a single fixed feature value). We assigned weight to a feature-value based on the percentage of times it is observed within training corpus for the respective typology feature.

If for any target language, there are defined set of rules that defines what value a typology feature can take within a context, such knowledge could be injected into the prior of the model, instead of injecting the fixed typology feature-values.

2. **Missing Typology:** In most typology databases, the values of numerous typology features for many languages are missing. This issue is more prominent for low-resource and less-documented languages but is also observed in well-documented languages as well if no dominant value is observed for a specific typology-feature. This limits the number of typology features or the number of languages for which the typology knowledge injection can be utilized with cross-lingual NLP model.

   For dependency Parsing and Enhanced Dependency Parsing, we aimed to address this issue by injecting the typology knowledge into the respective cross-lingual parser through multitasking. Multitasking allows the injection of typology feature by predicting these feature-values as an auxiliary task instead of feeding them directly along with word-embedding, thus even the typology feature with missing value can be used in this framework.

3. **Lack of Coverage:** For most of the syntactic and phonological typology database, we observed that the number of typology-features and the number of languages covered within these databases are very limited just making them less useful. For example, as described in chapter 7 we utilized the semantic typology knowledge available in ValPal database along with a state-of-the-art cross-lingual semantic role labelling model to improve its performance.

   However, as described in chapter 7 the ValPal database covers only 35 languages and that too with very limited vocabulary for most of these languages. We addressed this issue of limited by utilizing the FrameNet databases.

   If we encountered any missing verb-form within the training/test dataset, we

find a replacement-verb that belong to the same frame as that of the missing verb, and is available in ValPal. Subsequently, we assumed the feature-values of replacement verb as the feature-value for the missing verb. Our results showed that such approach worked, but this approaches is also limited by the availability of comprehensive FrameNet databases.

4. **Redundancy:** Most typology databases comprises of redundant features. For example WALS database consists of a feature titled *Order of Subject, Verb and Object*. The database also comprises of features *Order of Subject and Verb* and *Order of Verb and Object*. We observed that such redundant features may confuse the model thus leading to a drop in performance. We addressed this issue by manually filtering out the redundant features. Our experiments showed that such manual filtering does lead to marginal improvement in performance.

8.4 Future Research

The research-work described in this thesis can be extended in numerous directions. We discuss some of these directions in the following subsections.

8.4.1 Exploring new typology-features and new tasks

As already explained, our research-project is a wide-scope project, in which we attempted to inject typology knowledge available in numerous typology databases into cross-lingual models for a wide range of intermediary NLP tasks. However there are still a large number unexplored typology features and databases to be experimented with. Furthermore, there are numerous unexplored NLP tasks that can be aided with the linguistic typology knowledge within cross-lingual settings.
For example, there are numerous lexical databases such as the World Loanword Database (WOLD), the Intercontinental Dictionary Series (IDS), and the Automated Similarity Judgment Program (ASJP) and others (listed in chapter 2). The knowledge within these lexical databases can be utilised for tasks such as word-sense disambiguation or improving various multilingual/cross-lingual word-representations. Furthermore, morphological features is another set of typology features which are ignored in this project. These typology features can be used by researchers for various numerous morphological and lexical NLP tasks. Similarly, future researchers can also explore phonology typology features to aid speech-processing tasks such speech-recognition, speech-synthesis and speech-translation within cross-lingual settings.

In fact, the typology knowledge injection framework proposed/used by us in this project can also be indeed applied for these other tasks with these other typology features as well.

8.4.2 Exploring new typology knowledge injection frameworks

In this project, we explored numerous frameworks to inject linguistic typology knowledge into the neural-network models. These include *directly feeding-in*, *attention*, *deep probabilistic logic*, *multitasking* etc. Chapter 2 lists these frameworks in detail. There are numerous other frameworks proposed by researchers to inject external database knowledge into a machine-learning/deep-learning models. The popular frameworks for external knowledge injection into machine-learning models include knowledge induction through Posterior Regularization Ganchev et al. (2010), Generalized-Expectation based knowledge injection Mann and McCallum (2010), constraint-driven learning (CODL) Chang et al. (2007), injection with dual decomposition (DD) Komodakis et al. (2010), injecting typology knowledge by modelling the prior of bayesian model Cohen (2016) etc.

Similarly there are numerous opportunities for external knowledge injection into the deep-learning models, other than the ones explored by us in our project. For example, apart from *Deep probabilistic Logic* framework, there are other frameworks such as Hu et al. (2016b,a) to inject external knowledge as the first-order-logic constraints. Finally there are frameworks such as Vulić et al. (2017); Mrkšić et al. (2017); Ponti et al. (2018b) to inject external knowledge into the language-models for learning the multilingual text-representations (rather than injecting into the models for the main tasks).

In fact, building the frameworks for external knowledge injection in a neural-network model is an active research-area within the field of deep learning. Thus, new frameworks are being proposed every year. Researchers can modify and utilise these frameworks to improve typology knowledge injection.

8.4.3 Improvement of Multilingual Large Language Models with Typology

In section 2.1.1.4 we described the Transformer based Large Language Models (LLM). These LLM can be utilised to convert an input sentence (as word/token sequence) into a representation matrix. Such representation vector encode lexical, syntactic and semantic properties of the input sentence. These LLMs are trained on a large

raw-text corpus.
A topmost layer can be added to these LLMs depending upon the required downstream task. Recently these LLMs have demonstrated extraordinary performances on many downstream tasks. Furthermore, there are numerous multilingual LLMs that show extraordinary performance on cross-lingual tasks.
In future work, researchers can explore these multilingual LLM to investigate how much and which typology knowledge (about all the languages on which these are trained) is encoded within them. Researchers can adopt approaches similar to the typology feature-value prediction approach adopted by Malaviya et al. (2017) with features from LLMs, for such exploration. Finally the researchers can propose methods to inject the missing typology knowledge within these LLMs either during pre-training or through fine-tuning. Such knowledge injection should improve the cross-lingual transferability of these Multilingual LLMs even further.

8.4.4 Using Cross-lingual NLP for Typology

In this thesis, we aimed to inject the linguistic typology knowledge into various state-of-the-art approaches to numerous intermediate NLP tasks. However as described in section 8.3., one of the key limitations of using typology with cross-lingual NLP is the limited coverage as well as missing feature-values within the typology databases. In future work, the research can explore the reverse path as they can use cross-lingual NLP to predict the missing typology knowledge within the database.
For example, as described in chapter 7, the utilisation of ValPal database knowledge with cross-lingual SRL models was limited to a handful of languages due to low coverage of ValPal database. The ValPal database can be extended to a new language by autonomously creating a labelled dataset in that language using state-of-the-art cross-lingual SRL models, and then probabilistically extracting required argument patterns from it. Such research would be highly beneficial for the linguistics community.

8.4.5 Building new typology-databases

As explained in chapter 2, the manually created typology databases have several shortcomings including missing feature-values, inconsistencies in structure, redundancy issues etc. The NLP researchers aim to limit these shortcomings by applying various data-prepossessing techniques, however with limited effectiveness. All publicly available typology-databases were created by linguistics with no deep-learning or IT background, primarily to compare and study various linguistic structures in the

world. The databases were not created to be integrated with deep-learning models. Hence, there will always exists limitations with such integration.
Thus, future researchers can indeed explore bottoms-up approaches, where they could aim to build linguistic-typology knowledge and typology databases which are specifically built to be integrated into the deep-learning and machine learning models for various NLP tasks.

Appendix A

Results of End-to-end Model for Typology Feature Prediction

This appendix outlines all the results achieved for our End-to-end model for linguistic typology prediction described in chapter 4.

A.1 Results in Zero-shot learning

This section outlines the results in *Zero-shot* learning scenarios, obtained by the baseline Graph-based mBERT dependency parser Wu and Dredze (2019) as well as our proposed *Base End-to-end BERT parser* and *Multitasking End-to-end BERT parser* for all 90 target languages on which these models were evaluated, as Table A.1.

All *test* and *dev* corpora are downloaded from Universal Dependencies website. If Universal Dependencies website consists of more than one *test* (or *dev*) corpora for any target languages, all these test (or dev) corpora are concatenated into single *test* (or *dev*) corpora.

We summarise and describe the key inferences drawn from these results in details, in sections **??**.

Table A.1: Results achieved by various mBERT based Depndency Parsers evaluated on all 90 target languages under *Zero-shot* learning scenario

Begin of Table								
ZERO-SHOT								
	CL-Single				**CL-Poly**			
	mBERT	**Base E2E**	**Multi E2E**	**Aux task***	**mBERT**	**Base E2E**	**Multi E2E**	**Aux task***

aii	62.41	62.12	61.08	0.03	69.77	69.47	68.43	0.23
ar	48.53	48.19	47.2	0.03	73.2	72.9	71.84	0.24
hy	91.09	90.66	89.38	0.05	73.14	72.8	71.88	0.21
grc	90.82	90.54	89.38	0.05	71.79	71.51	70.58	0.22
af	90.54	90.24	88.98	0.06	74.23	73.81	72.76	0.21
am	62.29	61.93	60.76	0.04	68.63	68.29	67.15	0.15
akk	62.29	61.95	60.8	0.06	72.09	71.78	70.69	0.21
eu	40.53	40.1	38.83	0.1	57.9	57.62	56.41	0.27
zh	43.24	42.84	41.75	0.1	59.48	59.14	58.22	0.21
bxr	41.0	40.61	39.41	0.07	56.17	55.83	54.84	0.17
br	90.51	90.13	88.86	0.02	72.02	71.75	70.53	0.22
ca	79.92	79.55	78.57	0.06	69.55	69.27	68.34	0.18
bho	91.03	90.61	89.38	0.07	73.88	73.54	72.58	0.19
bg	85.93	85.5	84.45	0.04	69.18	68.84	67.85	0.26
bm	39.87	39.5	38.44	0.03	57.01	56.64	55.53	0.27
yue	41.68	41.37	40.36	0.07	53.23	52.83	51.9	0.16
be	90.83	90.47	89.2	0.09	73.89	73.54	72.45	0.24
cop	62.38	61.98	60.73	0.1	68.74	68.46	67.26	0.23
cs	73.11	72.73	71.73	0.05	71.31	71.02	69.81	0.2
lzh	41.27	40.97	39.97	0.07	58.29	57.97	56.76	0.16
hr	72.29	71.92	70.76	0.09	69.06	68.76	67.78	0.22
nl	76.07	75.8	74.65	0.01	72.77	72.39	71.3	0.22
en	92.65	92.38	91.27	0.09	69.36	68.97	68.02	0.21
da	81.82	81.53	80.55	0.1	69.25	68.82	67.57	0.2
fi	70.97	70.55	69.61	0.1	71.1	70.77	69.85	0.16
fr	84.62	84.21	83.28	0.03	73.7	73.33	72.38	0.25
et	70.77	70.48	69.48	0.07	68.54	68.16	67.17	0.15
myv	69.0	68.64	67.61	0.03	73.12	72.84	71.69	0.27
fo	91.02	90.64	89.53	0.06	69.48	69.05	68.11	0.14
de	77.97	77.68	76.69	0.1	73.83	73.49	72.55	0.16
gl	90.86	90.53	89.56	0.01	72.61	72.32	71.09	0.27
he	63.98	63.64	62.55	0.09	72.88	72.61	71.5	0.26
hi	44.56	44.25	43.24	0.02	74.01	73.72	72.58	0.17
id	56.18	55.79	54.53	0.05	51.92	51.65	50.53	0.25
hu	69.49	69.13	67.89	0.04	69.56	69.27	68.35	0.23
el	91.09	90.7	89.67	0.08	70.23	69.95	68.99	0.25
got	90.51	90.18	89.19	0.07	69.4	69.0	68.06	0.26
qhe	39.92	39.52	38.46	0.02	52.43	52.03	50.91	0.16
it	86.31	86.0	84.79	0.05	70.26	69.84	68.62	0.17
ja	35.89	35.46	34.39	0.07	71.4	71.08	69.92	0.2
ga	91.03	90.63	89.52	0.01	73.32	72.92	71.83	0.19
krl	69.19	68.87	67.73	0.09	71.93	71.64	70.42	0.21

ko	37.4	37.01	35.8	0.03	59.64	59.28	58.22	0.25
kpv	69.67	69.26	68.25	0.05	73.0	72.73	71.76	0.25
koi	69.68	69.27	68.16	0.04	72.49	72.19	71.2	0.16
kk	41.12	40.69	39.41	0.1	73.3	72.98	71.73	0.18
mr	90.71	90.32	89.35	0.01	73.35	72.92	71.72	0.21
lt	90.56	90.2	89.15	0.04	73.56	73.18	71.99	0.15
olo	69.41	69.08	67.9	0.08	74.11	73.8	72.54	0.22
la	50.99	50.71	49.47	0.06	69.53	69.14	67.86	0.21
lv	75.07	74.71	73.78	0.02	70.97	70.56	69.29	0.15
kmr	90.71	90.42	89.14	0.04	68.9	68.58	67.32	0.16
mt	62.16	61.81	60.77	0.04	69.25	68.97	67.98	0.24
gun	41.01	40.7	39.63	0.11	54.19	53.78	52.54	0.29
mdf	68.97	68.58	67.64	0.1	68.95	68.57	67.47	0.23
pcm	90.73	90.46	89.47	0.06	74.22	73.8	72.62	0.23
no	85.14	84.87	83.71	0.01	69.72	69.32	68.25	0.28
fro	90.72	90.39	89.19	0.1	70.28	69.88	68.64	0.24
sme	69.06	68.75	67.8	0.08	69.25	68.97	67.8	0.27
cu	90.98	90.64	89.72	0.04	68.63	68.26	67.14	0.25
orv	91.22	90.86	89.87	0.08	71.11	70.72	69.77	0.14
fa	91.32	91.0	89.95	0.01	70.35	69.92	68.9	0.16
pl	81.0	80.68	79.45	0.08	73.58	73.26	72.18	0.17
pt	82.32	81.92	80.7	0.06	71.5	71.1	69.9	0.26
ro	74.41	74.01	72.8	0.03	70.87	70.44	69.42	0.2
gd	91.2	90.79	89.7	0.07	69.79	69.51	68.46	0.23
ru	71.45	71.18	70.03	0.1	72.41	72.12	71.14	0.24
sa	90.55	90.19	89.23	0.09	74.46	74.17	72.92	0.14
sr	90.95	90.6	89.56	0.08	71.38	71.1	69.85	0.29
sms	69.45	69.03	67.82	0.03	71.63	71.3	70.08	0.19
sv	85.13	84.79	83.68	0.08	74.21	73.88	72.62	0.14
es	80.0	79.57	78.32	0.02	70.83	70.46	69.29	0.26
sl	74.3	74.02	72.87	0.09	71.82	71.4	70.18	0.15
sk	75.4	75.08	73.97	0.11	73.46	73.1	72.17	0.28
swl	40.03	39.61	38.58	0.07	54.82	54.53	53.6	0.24
th	40.65	40.27	39.26	0.03	55.32	54.92	53.68	0.21
gsw	90.86	90.52	89.4	0.11	73.68	73.25	72.11	0.22
uk	69.44	69.15	67.91	0.08	73.65	73.3	72.32	0.17
hsb	91.2	90.87	89.84	0.09	72.48	72.17	71.04	0.26
tr	41.06	40.66	39.57	0.05	70.68	70.33	69.37	0.21
te	38.51	38.15	36.94	0.09	71.61	71.23	70.07	0.15
tl	54.92	54.55	53.52	0.02	55.69	55.33	54.28	0.26
ta	38.62	38.19	37.15	0.02	71.71	71.31	70.13	0.27
ur	91.17	90.77	89.55	0.09	72.56	72.25	71.32	0.22

ug	42.06	41.73	40.61	0.05	69.56	69.29	68.3	0.27
vi	40.66	40.34	39.18	0.01	59.8	59.52	58.33	0.16
wo	42.08	41.79	40.62	0.08	53.65	53.37	52.27	0.22
yo	39.79	39.51	38.37	0.07	59.55	59.24	58.02	0.23
cy	91.1	90.83	89.65	0.04	71.75	71.36	70.36	0.29
wbp	39.93	39.55	38.48	0.06	52.29	51.99	50.83	0.19
End of Table								

A.2 Results in Few-shot learning

This section outlines the results obtained under *Few-shot* learning, by the baseline Graph-based mBERT dependency parser Wu and Dredze (2019) as well as our proposed *Base End-to-end BERT parser* and *Multitasking End-to-end BERT parser* for all 90 target languages on which these models were evaluated, as Table A.2.
Similar to the zero-hot learning setting, all *test* and *dev* corpora are downloaded from Universal Dependencies website. If Universal Dependencies website consists of more than one *test* (or *dev*) corpora for any target languages, all these test (or dev) corpora are concatenated into single *test* (or *dev*) corpora.

Table A.2: Results achieved by various mBERT based Depndency Parsers evaluated on all 90 target languages under *Few-shot* learning scenario

Begin of Table								
FEW-SHOT								
	CL-Single				**CL-Poly**			
	mBERT	**Base E2E**	**Multi E2E**	**Aux task***	**mBERT**	**Base E2E**	**Multi E2E**	**Aux task***
aii	63.32	62.89	63.45	0.57	70.58	70.15	71.02	0.58
ar	49.26	48.83	49.4	0.55	74.0	73.71	74.6	0.61
hy	91.73	91.31	91.76	0.5	74.02	73.67	74.49	0.6
grc	91.5	91.12	91.61	0.58	72.48	72.14	72.94	0.53
af	91.29	90.88	91.4	0.5	74.99	74.69	75.46	0.51
am	63.11	62.79	63.39	0.6	69.52	69.14	70.07	0.55
akk	62.99	62.67	63.16	0.51	72.96	72.64	73.59	0.62
eu	41.3	40.91	41.46	0.51	58.77	58.34	59.24	0.5
zh	44.1	43.74	44.19	0.53	60.15	59.72	60.59	0.61
bxr	41.83	41.46	41.92	0.54	56.95	56.61	57.54	0.63
br	91.44	91.01	91.49	0.51	72.68	72.32	73.25	0.57

ca	80.72	80.45	81.02	0.46	70.27	69.92	70.68	0.64
bho	91.77	91.38	91.9	0.59	74.81	74.49	75.31	0.61
bg	86.8	86.43	86.98	0.56	70.02	69.7	70.43	0.67
bm	40.73	40.46	40.92	0.51	57.84	57.47	58.36	0.66
yue	42.43	42.15	42.64	0.45	53.96	53.66	54.55	0.63
be	91.54	91.21	91.79	0.53	74.57	74.25	75.07	0.56
cop	63.06	62.76	63.32	0.6	69.62	69.19	69.99	0.61
cs	73.87	73.54	74.15	0.46	72.02	71.59	72.34	0.56
lzh	41.97	41.67	42.25	0.6	59.0	58.59	59.38	0.6
hr	73.17	72.76	73.35	0.52	69.73	69.39	70.27	0.67
nl	76.97	76.59	77.06	0.49	73.6	73.21	74.03	0.54
en	93.57	93.2	93.81	0.55	70.29	69.96	70.66	0.64
da	82.56	82.14	82.64	0.47	70.07	69.8	70.56	0.68
fi	71.63	71.34	71.91	0.52	71.95	71.64	72.57	0.59
fr	85.53	85.1	85.61	0.49	74.6	74.3	75.14	0.58
et	71.41	70.98	71.49	0.54	69.33	69.03	69.77	0.56
myv	69.91	69.49	70.08	0.47	73.81	73.54	74.36	0.66
fo	91.81	91.48	91.94	0.5	70.4	70.05	70.81	0.66
de	78.61	78.27	78.84	0.53	74.67	74.31	75.2	0.57
gl	91.56	91.29	91.83	0.51	73.38	72.95	73.73	0.52
he	64.75	64.36	64.81	0.47	73.7	73.28	74.08	0.5
hi	45.24	44.86	45.34	0.58	74.64	74.31	75.04	0.62
id	56.97	56.59	57.17	0.53	52.84	52.47	53.24	0.52
hu	70.4	69.98	70.46	0.6	70.47	70.04	70.91	0.66
el	91.79	91.52	92.05	0.56	70.93	70.63	71.5	0.58
got	91.23	90.86	91.38	0.58	70.05	69.65	70.46	0.65
qhe	40.57	40.25	40.8	0.56	53.3	52.87	53.59	0.55
it	86.97	86.62	87.19	0.51	70.97	70.65	71.62	0.5
ja	36.7	36.38	36.98	0.55	72.15	71.83	72.64	0.51
ga	91.94	91.6	92.17	0.59	74.01	73.68	74.39	0.67
krl	69.93	69.6	70.1	0.49	72.66	72.26	72.97	0.54
ko	38.15	37.82	38.36	0.49	60.41	60.11	60.95	0.53
kpv	70.42	70.13	70.73	0.52	73.64	73.21	74.07	0.6
koi	70.38	70.01	70.59	0.51	73.4	72.97	73.75	0.59
kk	42.04	41.72	42.25	0.58	74.16	73.88	74.81	0.57
mr	91.63	91.28	91.88	0.58	74.03	73.71	74.5	0.58
lt	91.24	90.9	91.39	0.54	74.23	73.84	74.68	0.51
olo	70.28	69.98	70.52	0.47	75.01	74.74	75.45	0.63
la	51.71	51.29	51.79	0.59	70.32	69.89	70.67	0.67
lv	75.82	75.39	75.9	0.48	71.87	71.55	72.34	0.54
kmr	91.53	91.21	91.81	0.52	69.74	69.36	70.19	0.55
mt	62.85	62.43	62.9	0.56	70.1	69.68	70.65	0.57

gun	41.84	41.46	42.01	0.53	55.03	54.65	55.45	0.62
mdf	69.81	69.52	70.13	0.47	69.77	69.42	70.27	0.66
pcm	91.43	91.08	91.66	0.47	75.13	74.74	75.66	0.56
no	86.01	85.69	86.23	0.47	70.37	69.98	70.84	0.59
fro	91.62	91.19	91.67	0.49	71.06	70.78	71.5	0.64
sme	69.85	69.46	69.97	0.6	70.11	69.82	70.67	0.53
cu	91.66	91.34	91.91	0.53	69.38	69.1	69.95	0.51
orv	92.11	91.79	92.4	0.54	71.92	71.62	72.36	0.53
fa	92.24	91.95	92.51	0.56	71.17	70.86	71.61	0.65
pl	81.84	81.41	81.96	0.53	74.28	73.91	74.73	0.53
pt	83.02	82.61	83.08	0.56	72.38	71.97	72.91	0.66
ro	75.31	75.02	75.56	0.52	71.63	71.26	72.14	0.61
gd	91.94	91.67	92.19	0.5	70.65	70.32	71.27	0.62
ru	72.2	71.82	72.41	0.53	73.06	72.67	73.42	0.65
sa	91.19	90.77	91.32	0.59	75.36	75.0	75.7	0.67
sr	91.7	91.27	91.73	0.46	72.29	71.91	72.61	0.6
sms	70.12	69.82	70.42	0.5	72.53	72.16	72.89	0.53
sv	85.83	85.52	86.02	0.46	74.84	74.48	75.29	0.67
es	80.64	80.3	80.87	0.48	71.64	71.32	72.06	0.53
sl	75.23	74.95	75.44	0.51	72.58	72.25	73.1	0.64
sk	76.19	75.88	76.4	0.6	74.33	74.03	74.95	0.5
swl	40.95	40.63	41.24	0.52	55.6	55.17	55.89	0.52
th	41.3	40.93	41.44	0.58	56.07	55.7	56.46	0.53
gsw	91.51	91.1	91.64	0.54	74.49	74.1	74.99	0.61
uk	70.24	69.94	70.45	0.58	74.41	74.07	74.81	0.64
hsb	91.93	91.63	92.14	0.51	73.24	72.84	73.66	0.59
tr	41.76	41.36	41.83	0.59	71.35	71.08	71.87	0.58
te	39.22	38.85	39.41	0.47	72.46	72.07	72.78	0.57
tl	55.83	55.52	56.04	0.48	56.52	56.15	56.94	0.61
ta	39.44	39.15	39.69	0.56	72.46	72.05	72.87	0.66
ur	91.83	91.44	92.05	0.58	73.32	72.94	73.87	0.53
ug	42.99	42.72	43.28	0.6	70.28	69.87	70.78	0.56
vi	41.32	40.97	41.45	0.57	60.71	60.28	61.17	0.6
wo	42.95	42.65	43.17	0.5	54.35	53.94	54.76	0.61
yo	40.53	40.25	40.75	0.55	60.48	60.07	60.88	0.56
cy	91.82	91.46	91.93	0.56	72.66	72.38	73.22	0.55
wbp	40.8	40.52	41.03	0.56	53.21	52.83	53.79	0.5
End of Table								

Appendix B

Results of UDify with Typology model

This appendix outlines the results obtained by the three variants of our proposed models namely *UDify-w-Syntax* (predicts only syntactic typology features), *UDify-w-Syntactic+Semantic* (predicts syntactic and semantic typology-features) and *UDify-w-All* (predicts all the URIEL typology-features), as well as the baselines described in Section 5.5.1 as table B.1.
We summarise and describe all the key inferences drawn from these results in details, in sections 5.5.6 of chapter 5.

Table B.1: Results achieved on all 80 test tree-banks

Begin of Table							
Corpus	**Model**	**UPOS**	**UFeats**	**Lemma**	**UAS**	**LAS**	**Typo F1**
Afrikaans-AfriBooms (size: 1315)	UDPipe	98.25	97.66	97.46	91.26	88.46	–
	UDify	95.31	91.34	94.5	88.79	85.17	–
	Multi-w-Lang_id	96.61	92.64	94.84	90.15	87.87	–
	Multi-w-Syntax	96.73	93.51	95.04	94.36	89.96	82.27
	Multi-w-Syntax+Semantic	94.8	90.51	88.31	83.91	90.63	74.82
	Multi-w-All	93.73	88.8	86.48	81.5	85.96	64.94
Arabic-PADT (size: 21864)	UDPipe	96.83	94.11	95.28	88.29	83.69	–
	UDify	95.35	99.35	99.97	88.6	84.42	–
	Multi-w-Lang_id	96.64	99.33	99.66	89.92	87.13	–
	Multi-w-Syntax	96.76	99.31	99.59	93.78	89.24	81.75

	Multi-w-Syntax+Semantic	96.41	92.77	94.39	90.83	84.12	74.5
	Multi-w-All	96.16	89.76	90.53	86.93	81.51	69.26
Armenian-ArmTDP (size: 1975)	UDPipe	93.49	82.85	92.86	79.65	72.3	–
	UDify	94.42	76.9	85.63	87.01	79.99	–
	Multi-w-Lang_id	96.02	80.58	86.62	89.06	84.3	–
	Multi-w-Syntax	96.15	83.06	87.19	93.43	86.23	83.55
	Multi-w-Syntax+Semantic	92.3	82.87	86.93	85.35	84.2	71.52
	Multi-w-All	91.5	81.24	85.22	79.53	81.21	60.95
Basque-BDT (size: 5396)	UDPipe	96.11	92.48	96.29	86.8	83.55	–
	UDify	95.45	86.8	90.53	85.47	81.5	–
	Multi-w-Lang_id	96.71	88.85	91.16	88.02	85.28	–
	Multi-w-Syntax	95.45	94.95	98.46	92.96	87.4	88.3
	Multi-w-Syntax+Semantic	95.58	87.18	82.9	88.37	81.11	79.26
	Multi-w-All	93.56	84.17	80.34	84.61	79.0	73.01
Belarusian-HSE (size: 319)	UDPipe	93.63	73.3	87.34	80.44	74.58	–
	UDify	96.12	88.36	93.97	91.08	88.59	–
	Multi-w-Lang_id	97.01	95.77	96.72	93.69	89.9	–
	Multi-w-Syntax	97.13	96.22	96.83	95.59	92.26	83.94
	Multi-w-Syntax+Semantic	96.64	92.73	89.73	91.63	92.88	73.98
	Multi-w-All	95.56	90.76	88.24	86.3	88.83	66.82
Bulgarian-BTB (size: 8907)	UDPipe	98.98	97.82	97.94	95.21	92.18	–
	UDify	96.7	96.57	95.1	95.7	92.58	–
	Multi-w-Lang_id	97.54	97.01	95.4	95.05	92.26	–
	Multi-w-Syntax	97.64	97.3	95.57	96.61	93.46	82.07
	Multi-w-Syntax+Semantic	95.06	93.48	87.06	98.7	94.63	74.77
	Multi-w-All	93.42	92.38	84.14	93.69	92.16	69.3
Buryat-BDT (size: 19)	UDPipe	40.34	32.4	58.17	34.07	20.3	–
	UDify	61.73	47.45	61.03	49.61	27.46	–
	Multi-w-Lang_id	73.25	47.9	61.09	56.98	41.08	–
	Multi-w-Syntax	73.73	54.74	62.8	74.42	58.5	82.69

	Multi-w-Syntax+Semantic	72.56	53.23	59.49	77.17	47.95	71.76
	Multi-w-All	70.76	49.35	56.68	73.23	44.38	63.12
Catalan-AnCora (size: 13123)	UDPipe	98.88	98.37	99.07	95.12	92.96	–
	UDify	98.89	98.34	98.14	95.61	93.69	–
	Multi-w-Lang_id	99.0	98.49	98.22	95.9	93.96	–
	Multi-w-Syntax	99.08	98.58	98.26	96.97	93.55	81.77
	Multi-w-Syntax+Semantic	97.47	95.05	95.29	91.97	90.02	71.06
	Multi-w-All	96.12	93.31	92.87	86.69	87.62	60.62
Chinese-GSD (size: 7994)	UDPipe	94.88	99.22	99.99	85.84	81.7	–
	UDify	93.48	99.31	100.0	92.98	84.66	–
	Multi-w-Lang_id	97.46	93.0	75.43	90.79	86.76	–
	Multi-w-Syntax	97.57	93.83	76.49	94.61	89.19	60.82
	Multi-w-Syntax+Semantic	95.17	93.62	70.49	86.34	91.08	49.85
	Multi-w-All	94.17	91.89	67.93	81.95	88.31	38.65
Coptic-Scriptorium (size: 792)	UDPipe	94.7	96.35	95.49	87.4	82.79	–
	UDify	27.17	52.85	55.71	28.29	11.53	–
	Multi-w-Lang_id	51.24	60.49	58.89	51.29	32.13	–
	Multi-w-Syntax	52.06	65.65	60.7	71.54	53.73	84.55
	Multi-w-Syntax+Semantic	50.62	59.7	59.62	71.52	49.73	75.59
	Multi-w-All	48.95	56.29	57.39	68.19	43.67	64.17
Croatian-SET (size: 6914)	UDPipe	98.13	92.25	97.27	92.45	88.13	–
	UDify	97.89	88.97	97.15	92.98	90.5	–
	Multi-w-Lang_id	98.33	90.66	97.3	94.29	92.07	–
	Multi-w-Syntax	98.42	91.8	97.38	95.65	92.08	81.92
	Multi-w-Syntax+Semantic	97.08	86.49	92.89	95.4	81.97	72.68
	Multi-w-All	96.44	83.45	90.91	90.35	75.32	61.19
Czech-CAC (size: 102993)	UDPipe	99.37	96.34	98.57	93.48	91.2	–
	UDify	98.14	96.55	97.18	94.74	92.77	–
	Multi-w-Lang_id	98.5	96.99	97.33	93.9	92.84	–

	Multi-w-Syntax	98.59	97.29	97.41	96.04	93.82	82.61
	Multi-w-Syntax+Semantic	96.63	94.17	94.72	97.75	87.74	76.35
	Multi-w-All	96.29	90.57	92.35	91.46	85.34	65.28
Czech-CLTT (size: 102993)	UDPipe	98.88	91.59	98.25	87.86	84.99	–
	UDify	99.17	93.66	98.86	93.7	91.97	–
	Multi-w-Lang_id	99.18	94.58	98.88	94.14	91.71	–
	Multi-w-Syntax	99.26	95.19	98.9	95.13	93.7	82.49
	Multi-w-Syntax+Semantic	98.64	91.41	91.24	86.51	89.79	74.57
	Multi-w-All	96.66	88.17	87.53	80.78	86.25	66.55
Czech-FicTree (size: 102993)	UDPipe	98.55	95.87	98.63	93.32	90.16	–
	UDify	98.18	96.36	97.33	95.77	93.98	–
	Multi-w-Lang_id	98.52	96.83	97.47	95.9	93.27	–
	Multi-w-Syntax	98.61	97.15	97.54	95.3	93.52	82.45
	Multi-w-Syntax+Semantic	97.04	95.16	88.41	89.48	94.11	75.34
	Multi-w-All	95.34	92.47	84.58	83.23	87.94	64.9
Czech-PDT (size: 102993)	UDPipe	99.18	97.23	99.02	94.94	92.92	–
	UDify	98.21	98.38	97.55	96.27	93.99	–
	Multi-w-Lang_id	98.54	98.52	97.67	96.08	93.1	–
	Multi-w-Syntax	98.63	98.61	97.74	95.69	94.8	81.59
	Multi-w-Syntax+Semantic	95.22	96.9	94.12	86.46	96.18	71.77
	Multi-w-All	93.41	93.3	91.73	83.01	92.49	65.57
Danish-DDT (size: 4383)	UDPipe	97.78	97.33	97.52	88.25	85.68	–
	UDify	96.02	89.78	91.0	89.76	85.52	–
	Multi-w-Lang_id	97.09	91.34	91.6	92.53	87.77	–
	Multi-w-Syntax	97.2	92.39	91.94	93.76	89.87	82.18
	Multi-w-Syntax+Semantic	96.56	90.73	85.33	93.11	83.86	73.52
	Multi-w-All	95.36	89.03	83.13	87.91	80.49	64.36
Dutch-Alpino (size: 18051)	UDPipe	96.83	96.33	97.09	93.13	90.14	–
	UDify	97.12	92.59	98.23	95.82	92.15	–

	Multi-w-Lang_id	97.82	93.69	98.3	96.25	92.69	–
	Multi-w-Syntax	97.92	94.42	98.34	96.59	93.22	82.11
	Multi-w-Syntax+Semantic	97.58	92.81	90.05	87.22	91.38	74.53
	Multi-w-All	96.59	91.16	88.05	82.43	87.64	64.24
Dutch-LassySmall (size: 18051)	UDPipe	96.5	96.42	97.41	91.82	88.01	–
	UDify	98.89	96.18	93.49	95.73	92.59	–
	Multi-w-Lang_id	99.0	96.68	93.91	96.14	93.95	–
	Multi-w-Syntax	99.08	97.02	94.14	96.05	94.27	82.29
	Multi-w-Syntax+Semantic	96.1	90.53	86.95	85.32	85.91	71.28
	Multi-w-All	94.1	89.36	84.08	82.57	80.57	60.34
English-EWT (size: 25377)	UDPipe	96.29	97.1	98.25	91.21	88.55	–
	UDify	97.73	96.12	95.84	94.64	90.04	–
	Multi-w-Lang_id	98.22	96.63	96.09	93.9	90.07	–
	Multi-w-Syntax	98.32	96.97	96.22	94.76	91.65	81.84
	Multi-w-Syntax+Semantic	95.0	96.22	92.23	88.32	81.7	72.3
	Multi-w-All	94.52	95.07	90.36	81.53	75.82	66.63
English-GUM (size: 25377)	UDPipe	96.02	96.82	96.85	88.4	85.25	–
	UDify	95.44	94.12	93.15	91.01	87.6	–
	Multi-w-Lang_id	96.7	94.96	93.59	92.8	89.74	–
	Multi-w-Syntax	96.82	95.53	93.84	93.3	91.27	82.38
	Multi-w-Syntax+Semantic	96.33	88.65	85.01	91.19	88.07	71.87
	Multi-w-All	95.15	87.52	83.36	88.24	85.02	66.05
English-LinES (size: 25377)	UDPipe	96.91	96.31	96.45	84.79	80.35	–
	UDify	94.55	90.43	94.42	89.56	85.34	–
	Multi-w-Lang_id	96.11	91.88	94.77	91.71	88.13	–
	Multi-w-Syntax	96.23	92.86	94.97	93.51	89.89	82.15
	Multi-w-Syntax+Semantic	92.8	88.52	86.98	83.74	87.1	72.22
	Multi-w-All	92.35	86.44	84.13	79.28	82.34	63.23
English-ParTUT (size: 25377)	UDPipe	96.1	95.51	97.74	91.53	88.51	–
	UDify	96.16	92.61	96.45	94.72	92.02	–

	Multi-w-Lang_id	97.18	93.7	96.65	94.19	92.97	–
	Multi-w-Syntax	97.29	94.43	96.76	94.66	93.07	82.21
	Multi-w-Syntax+Semantic	95.24	87.42	89.92	93.54	87.18	74.59
	Multi-w-All	94.78	84.1	86.6	90.09	82.66	65.91
Estonian-EDT (size: 25749)	UDPipe	97.64	96.23	95.3	88.52	85.7	–
	UDify	96.91	87.45	77.73	91.65	86.97	–
	Multi-w-Lang_id	97.68	89.39	79.3	92.51	87.78	–
	Multi-w-Syntax	98.16	97.34	95.68	93.24	89.49	94.0
	Multi-w-Syntax+Semantic	95.95	88.72	74.83	90.1	86.44	84.33
	Multi-w-All	93.85	85.02	72.23	84.23	84.05	76.18
Finnish-FTB (size: 27198)	UDPipe	96.65	96.62	95.49	90.89	88.1	–
	UDify	94.37	82.8	96.68	88.8	83.21	–
	Multi-w-Lang_id	95.99	85.51	96.86	90.97	85.49	–
	Multi-w-Syntax	96.12	87.33	96.97	94.4	89.35	82.17
	Multi-w-Syntax+Semantic	94.63	85.13	95.54	83.78	86.72	72.11
	Multi-w-All	94.1	83.61	93.65	79.29	82.58	66.5
Finnish-TDT (size: 27198)	UDPipe	97.45	95.43	91.45	90.67	88.25	–
	UDify	94.43	90.48	82.89	86.8	82.41	–
	Multi-w-Lang_id	96.03	91.92	84.08	89.67	85.5	–
	Multi-w-Syntax	96.16	92.89	84.76	92.58	89.15	82.76
	Multi-w-Syntax+Semantic	94.19	87.24	77.24	90.98	88.75	71.47
	Multi-w-All	93.01	83.9	75.86	87.38	83.01	65.01
French-GSD (size: 33399)	UDPipe	97.63	97.13	98.35	91.77	89.18	–
	UDify	99.14	95.42	98.32	94.77	92.85	–
	Multi-w-Lang_id	99.16	96.05	98.38	94.18	92.07	–
	Multi-w-Syntax	99.24	96.47	98.42	95.57	93.09	82.08
	Multi-w-Syntax+Semantic	98.52	93.7	97.28	91.49	92.14	71.13
	Multi-w-All	97.28	91.12	93.64	84.77	87.26	64.84
French-ParTUT (size: 33399)	UDPipe	96.93	94.43	95.7	93.97	91.43	–
	UDify	95.91	95.08	96.52	92.24	88.65	–

	Multi-w-Lang_id	97.8	91.27	92.16	93.67	90.63	–
	Multi-w-Syntax	97.91	92.33	92.47	94.05	91.75	78.07
	Multi-w-Syntax+Semantic	96.92	90.9	88.81	89.36	88.05	66.92
	Multi-w-All	95.29	89.88	84.99	85.83	86.02	58.57
French-Sequoia (size: 33399)	UDPipe	98.79	98.09	98.57	93.84	92.2	–
	UDify	98.11	95.92	95.5	93.15	90.27	–
	Multi-w-Lang_id	98.48	96.47	95.77	93.74	90.82	–
	Multi-w-Syntax	98.57	96.83	95.92	94.37	91.27	82.08
	Multi-w-Syntax+Semantic	95.2	90.06	86.83	88.32	86.32	75.4
	Multi-w-All	93.9	86.61	84.06	82.46	84.06	68.37
French-Spoken (size: 33399)	UDPipe	95.91	100.0	96.92	83.08	77.71	–
	UDify	96.23	98.67	96.59	86.42	81.19	–
	Multi-w-Lang_id	97.23	98.76	96.78	90.28	84.23	–
	Multi-w-Syntax	97.34	98.82	96.89	93.39	88.49	81.8
	Multi-w-Syntax+Semantic	94.08	94.21	87.61	82.96	81.89	75.73
	Multi-w-All	93.18	92.38	83.82	78.53	76.85	69.6
Galician-CTG (size: 2872)	UDPipe	97.84	99.83	98.58	86.66	84.04	–
	UDify	96.51	97.1	97.08	84.88	81.02	–
	Multi-w-Lang_id	97.41	97.45	97.23	88.55	83.49	–
	Multi-w-Syntax	97.52	97.68	97.32	92.77	88.59	82.22
	Multi-w-Syntax+Semantic	96.75	93.95	89.53	91.93	89.52	75.0
	Multi-w-All	95.31	92.51	85.92	87.76	83.59	69.39
Galician-TreeGal (size: 2872)	UDPipe	95.82	93.96	97.06	83.26	78.23	–
	UDify	94.59	80.67	94.93	85.52	78.21	–
	Multi-w-Lang_id	96.13	83.73	95.24	88.31	81.67	–
	Multi-w-Syntax	96.26	85.79	95.42	92.63	85.98	81.8
	Multi-w-Syntax+Semantic	94.16	84.84	88.28	89.43	84.42	69.56
	Multi-w-All	93.12	82.96	85.17	82.64	81.41	58.73
German-GSD (size: 166849)	UDPipe	94.48	90.68	96.8	87.17	82.71	–
	UDify	97.48	96.63	95.23	88.64	85.15	–

	Multi-w-Lang_id	98.06	97.06	95.52	91.49	86.49	–
	Multi-w-Syntax	97.78	90.7	80.19	93.92	89.53	69.14
	Multi-w-Syntax+Semantic	97.85	91.07	94.13	91.44	83.6	59.88
	Multi-w-All	96.76	89.17	90.6	87.22	80.87	54.63
Gothic-PROIEL (size: 3387)	UDPipe	96.61	90.73	94.75	86.61	80.93	–
	UDify	95.55	85.97	80.57	86.37	80.13	–
	Multi-w-Lang_id	96.77	88.16	81.93	88.24	84.09	–
	Multi-w-Syntax	97.7	92.18	92.64	91.62	87.82	90.23
	Multi-w-Syntax+Semantic	95.55	85.73	75.99	90.14	80.92	82.58
	Multi-w-All	93.8	82.53	71.99	84.73	78.59	71.71
Greek-GDT (size: 1662)	UDPipe	97.98	94.96	95.82	92.9	90.59	–
	UDify	97.08	99.97	98.8	95.91	93.62	–
	Multi-w-Lang_id	97.79	99.78	98.83	94.66	93.51	–
	Multi-w-Syntax	97.89	99.87	98.84	96.56	94.05	81.96
	Multi-w-Syntax+Semantic	97.69	92.93	97.88	96.65	87.99	70.44
	Multi-w-All	95.88	89.61	96.23	91.83	85.26	63.61
Hebrew-HTB (size: 5241)	UDPipe	97.02	95.87	97.12	90.4	87.56	–
	UDify	96.21	96.02	97.28	92.14	89.68	–
	Multi-w-Lang_id	97.21	96.55	97.42	93.99	91.55	–
	Multi-w-Syntax	97.32	96.9	97.5	94.2	92.59	82.29
	Multi-w-Syntax+Semantic	97.32	95.77	94.17	92.57	82.66	72.15
	Multi-w-All	96.81	92.77	91.51	87.58	76.24	66.3
Hindi-HDTB (size: 13304)	UDPipe	97.52	94.15	98.67	94.95	91.93	–
	UDify	98.3	92.22	95.86	95.93	92.2	–
	Multi-w-Lang_id	98.6	93.38	96.1	95.47	93.32	–
	Multi-w-Syntax	98.69	94.15	96.24	95.72	92.11	82.27
	Multi-w-Syntax+Semantic	98.5	91.25	87.06	85.1	94.9	76.16
	Multi-w-All	97.47	89.99	84.47	81.16	89.88	67.11
Hungarian-Szeged (size: 910)	UDPipe	95.76	91.75	95.05	84.17	79.86	–
	UDify	96.36	86.16	90.19	91.01	86.21	–

	Multi-w-Lang_id	97.31	88.31	90.85	91.8	88.59	–
	Multi-w-Syntax	97.42	89.76	91.22	94.65	90.96	82.35
	Multi-w-Syntax+Semantic	94.01	84.9	87.63	94.74	81.06	71.86
	Multi-w-All	93.46	81.98	86.48	91.43	74.44	65.56
Indonesian-GSD (size: 4477)	UDPipe	93.69	95.58	99.64	86.54	80.22	–
	UDify	93.36	93.32	98.37	87.75	81.4	–
	Multi-w-Lang_id	95.31	94.29	98.43	90.96	84.62	–
	Multi-w-Syntax	96.82	90.23	91.52	92.83	87.4	76.11
	Multi-w-Syntax+Semantic	92.65	87.9	93.12	88.35	85.84	64.79
	Multi-w-All	91.91	86.23	89.17	85.59	83.36	55.57
Irish-IDT (size: 858)	UDPipe	92.72	82.43	90.48	81.77	73.72	–
	UDify	90.96	82.09	81.08	79.38	70.65	–
	Multi-w-Lang_id	93.72	84.91	82.4	84.27	76.57	–
	Multi-w-Syntax	93.88	86.82	83.16	90.08	82.83	83.97
	Multi-w-Syntax+Semantic	90.39	83.34	81.23	82.63	73.14	76.57
	Multi-w-All	89.53	80.31	78.39	76.84	68.89	71.55
Italian-ISDT (size: 29685)	UDPipe	98.39	98.11	98.66	95.24	93.29	–
	UDify	98.51	98.01	97.72	96.15	94.3	–
	Multi-w-Lang_id	98.74	98.21	97.83	96.57	93.16	–
	Multi-w-Syntax	98.83	98.34	97.89	96.53	94.18	82.27
	Multi-w-Syntax+Semantic	95.14	93.4	94.91	94.87	92.56	71.28
	Multi-w-All	94.13	90.05	90.97	91.84	88.32	62.14
Italian-ParTUT (size: 29685)	UDPipe	98.38	97.77	98.16	93.62	91.45	–
	UDify	99.18	96.69	98.52	95.9	94.05	–
	Multi-w-Lang_id	99.19	97.11	98.57	95.38	94.85	–
	Multi-w-Syntax	99.27	97.39	98.6	96.97	94.76	81.61
	Multi-w-Syntax+Semantic	96.52	90.68	93.6	98.24	89.89	74.21
	Multi-w-All	94.83	88.46	91.05	94.28	87.47	65.06
Japanese-GSD (size: 47926)	UDPipe	98.13	99.98	99.52	95.99	94.66	–
	UDify	98.73	93.44	96.5	95.1	93.43	–

	Multi-w-Lang_id	98.22	94.27	90.14	94.89	93.71	–
	Multi-w-Syntax	98.31	94.93	90.55	95.15	94.23	76.72
	Multi-w-Syntax+Semantic	96.63	91.07	90.11	97.63	91.41	70.03
	Multi-w-All	95.98	90.07	87.13	95.22	88.14	63.07
Kazakh-KTB (size: 31)	UDPipe	55.84	40.4	63.96	55.12	35.2	–
	UDify	91.29	99.58	99.21	74.74	66.63	–
	Multi-w-Lang_id	93.94	99.52	99.21	81.92	72.97	–
	Multi-w-Syntax	94.1	99.48	99.21	88.58	80.55	82.36
	Multi-w-Syntax+Semantic	91.45	96.21	96.02	81.59	80.85	73.43
	Multi-w-All	90.23	93.96	93.1	75.99	74.65	62.45
Korean-GSD (size: 27410)	UDPipe	96.29	99.77	93.4	88.84	85.38	–
	UDify	91.98	99.89	100.0	83.24	75.73	–
	Multi-w-Lang_id	94.4	99.56	99.75	87.17	81.12	–
	Multi-w-Syntax	94.55	99.63	99.59	91.16	84.3	81.2
	Multi-w-Syntax+Semantic	92.58	94.34	92.28	90.01	85.69	68.82
	Multi-w-All	92.18	92.19	89.53	87.41	79.76	63.39
Korean-Kaist (size: 27410)	UDPipe	95.59	100.0	94.3	88.62	86.68	–
	UDify	94.67	99.98	85.89	87.9	84.85	–
	Multi-w-Lang_id	96.19	99.64	86.86	90.58	87.0	–
	Multi-w-Syntax	96.31	99.9	87.42	92.87	89.07	83.56
	Multi-w-Syntax+Semantic	94.16	97.98	81.44	94.07	80.1	75.91
	Multi-w-All	93.34	95.61	77.72	88.69	74.55	68.58
Kurmanji-MG (size: 20)	UDPipe	53.36	41.54	69.58	46.16	35.25	–
	UDify	60.23	37.78	58.08	36.98	21.52	–
	Multi-w-Lang_id	74.25	55.98	63.82	64.24	44.36	–
	Multi-w-Syntax	74.72	61.74	65.41	78.91	60.28	85.67
	Multi-w-Syntax+Semantic	72.44	60.99	58.85	72.01	61.32	73.28
	Multi-w-All	71.18	60.04	55.01	66.12	58.02	66.53
Latin-ITTB (size: 34060)	UDPipe	98.34	96.97	98.99	92.35	90.09	–
	UDify	97.71	88.63	94.0	93.22	90.69	–
	Multi-w-Lang_id	98.21	90.38	94.38	94.19	91.25	–

	Multi-w-Syntax	97.8	95.01	94.73	95.24	91.82	82.42
	Multi-w-Syntax+Semantic	97.18	87.84	85.95	88.24	92.54	75.68
	Multi-w-All	96.76	85.14	82.36	84.01	87.23	64.79
Latin-Perseus (size: 34060)	UDPipe	88.4	79.1	81.45	72.86	62.94	–
	UDify	91.5	83.21	80.84	80.24	72.19	–
	Multi-w-Lang_id	94.08	85.85	82.18	84.84	78.38	–
	Multi-w-Syntax	94.24	87.63	82.95	90.66	82.9	83.56
	Multi-w-Syntax+Semantic	90.67	84.65	78.08	91.71	84.56	73.65
	Multi-w-All	90.08	81.38	74.86	86.87	77.64	64.8
Latin-PROIEL (size: 34060)	UDPipe	97.01	91.53	96.32	84.97	80.29	–
	UDify	96.79	89.49	91.79	85.89	81.56	–
	Multi-w-Lang_id	97.6	91.1	92.33	87.94	85.54	–
	Multi-w-Syntax	96.89	89.63	82.71	93.14	88.1	74.65
	Multi-w-Syntax+Semantic	94.25	87.77	89.66	87.03	80.45	65.26
	Multi-w-All	93.56	84.81	86.89	83.26	77.58	56.93
Latvian-LVTB (size: 10156)	UDPipe	96.11	93.01	95.46	87.6	83.75	–
	UDify	97.5	95.41	94.6	88.94	85.68	–
	Multi-w-Lang_id	98.07	96.04	94.94	91.26	87.31	–
	Multi-w-Syntax	97.74	90.55	93.44	93.79	90.15	81.29
	Multi-w-Syntax+Semantic	97.47	91.74	86.86	87.64	79.88	71.0
	Multi-w-All	95.4	88.6	84.17	84.92	74.62	65.49
Lithuanian-HSE (size: 2494)	UDPipe	81.7	60.47	76.89	53.52	43.71	–
	UDify	90.49	71.84	81.27	81.15	70.38	–
	Multi-w-Lang_id	93.39	74.81	69.51	85.29	76.17	–
	Multi-w-Syntax	93.56	78.07	70.84	90.47	81.32	74.84
	Multi-w-Syntax+Semantic	91.27	77.81	65.57	83.26	80.25	64.85
	Multi-w-All	90.6	76.85	62.99	77.68	76.15	54.86
Maltese-MUDT (size: 1123)	UDPipe	95.99	100.0	100.0	86.18	81.24	–
	UDify	90.56	99.63	82.84	84.65	76.17	–

	Multi-w-Lang_id	93.45	99.48	84.04	88.21	80.22	–
	Multi-w-Syntax	93.62	99.4	84.72	92.02	85.5	82.93
	Multi-w-Syntax+Semantic	92.66	93.39	80.36	83.99	81.66	75.95
	Multi-w-All	92.39	90.17	77.56	77.68	78.32	65.6
Marathi-UFAL (size: 373)	UDPipe	80.1	67.23	81.31	71.59	62.37	–
	UDify	94.29	84.49	87.71	76.46	69.34	–
	Multi-w-Lang_id	95.93	86.92	88.55	82.65	76.35	–
	Multi-w-Syntax	96.06	88.56	89.03	88.99	82.35	82.7
	Multi-w-Syntax+Semantic	94.22	86.29	82.01	83.57	76.35	73.14
	Multi-w-All	93.6	82.85	80.63	77.5	73.46	66.84
Norwegian-Bokmaal (size: 33282)	UDPipe	98.31	97.14	98.64	93.07	91.17	–
	UDify	98.34	91.82	98.13	96.37	93.95	–
	Multi-w-Lang_id	98.63	93.04	98.21	95.16	93.86	–
	Multi-w-Syntax	98.72	93.86	98.25	95.93	92.85	82.54
	Multi-w-Syntax+Semantic	98.67	88.01	92.42	92.62	88.51	76.51
	Multi-w-All	97.17	86.85	89.65	87.88	83.47	66.37
Norwegian-Nynorsk (size: 33282)	UDPipe	98.14	97.02	98.18	93.71	91.63	–
	UDify	97.83	96.17	97.34	95.08	92.93	–
	Multi-w-Lang_id	98.29	96.68	97.48	94.66	92.82	–
	Multi-w-Syntax	98.38	97.01	97.55	96.47	93.01	82.47
	Multi-w-Syntax+Semantic	98.11	96.32	90.59	87.94	85.1	71.03
	Multi-w-All	96.65	95.16	88.84	85.59	79.94	64.62
Norwegian-NynorskLIA (size: 33282)	UDPipe	89.59	86.13	93.93	69.27	61.26	–
	UDify	95.01	93.36	96.13	75.8	70.0	–
	Multi-w-Lang_id	96.41	94.33	96.35	82.11	76.21	–
	Multi-w-Syntax	96.54	94.98	96.48	89.43	82.32	82.45
	Multi-w-Syntax+Semantic	96.47	91.97	93.68	84.95	71.57	75.39
	Multi-w-All	94.4	88.52	92.04	81.15	65.31	66.05
Persian-Seraji (size: 4798)	UDPipe	97.75	97.78	97.44	91.68	88.29	–
	UDify	96.22	94.73	92.55	91.21	87.46	–

	Multi-w-Lang_id	97.22	95.47	93.04	93.1	89.74	–
	Multi-w-Syntax	98.19	92.08	87.04	95.54	91.46	76.91
	Multi-w-Syntax+Semantic	95.33	89.81	89.63	85.58	88.76	65.78
	Multi-w-All	95.08	88.41	86.93	79.6	84.56	56.29
Polish-LFG (size: 31496)	UDPipe	98.8	95.49	97.54	96.77	94.95	–
	UDify	98.97	96.29	94.47	96.82	95.12	–
	Multi-w-Lang_id	99.05	96.78	94.82	96.24	94.76	–
	Multi-w-Syntax	99.13	97.1	95.01	96.58	94.42	82.34
	Multi-w-Syntax+Semantic	95.77	95.71	90.4	97.59	94.27	75.21
	Multi-w-All	95.55	92.96	88.51	92.51	90.71	64.28
Portuguese-Bosque (size: 17992)	UDPipe	97.07	96.4	98.46	91.48	89.16	–
	UDify	97.54	89.36	85.46	93.38	88.75	–
	Multi-w-Lang_id	98.1	90.99	86.46	92.93	90.17	–
	Multi-w-Syntax	97.33	95.97	93.31	93.61	91.43	88.42
	Multi-w-Syntax+Semantic	97.25	86.0	86.6	85.0	87.72	78.71
	Multi-w-All	96.86	83.66	85.4	82.64	85.66	72.05
Portuguese-GSD (size: 17992)	UDPipe	98.31	99.92	99.3	94.28	92.9	–
	UDify	98.04	95.75	98.95	96.21	94.53	–
	Multi-w-Lang_id	98.43	96.32	98.97	95.64	95.03	–
	Multi-w-Syntax	98.52	96.71	98.98	96.08	94.67	82.17
	Multi-w-Syntax+Semantic	94.99	90.1	91.69	89.47	84.48	74.26
	Multi-w-All	93.54	88.1	88.37	85.86	79.15	65.18
Romanian-Nonstandard (size: 21782)	UDPipe	96.68	90.88	94.78	90.07	85.15	–
	UDify	96.85	87.24	92.7	89.73	86.45	–
	Multi-w-Lang_id	97.64	89.22	93.17	92.32	89.02	–
	Multi-w-Syntax	98.17	96.46	95.13	94.02	90.33	84.2
	Multi-w-Syntax+Semantic	95.03	86.69	87.79	94.45	92.69	76.75
	Multi-w-All	94.77	84.06	84.24	89.51	86.1	67.56
Romanian-RRT (size: 21782)	UDPipe	97.96	97.53	98.41	92.72	88.15	–
	UDify	96.94	93.41	94.15	93.43	89.91	–

	Multi-w-Lang_id	97.7	94.37	94.52	93.95	91.22	–
	Multi-w-Syntax	98.31	91.55	94.59	94.39	91.86	82.46
	Multi-w-Syntax+Semantic	97.79	88.58	92.66	96.92	89.67	74.87
	Multi-w-All	96.92	85.79	91.08	91.81	85.74	63.73
Russian-GSD (size: 54099)	UDPipe	97.1	92.66	97.37	89.47	85.69	–
	UDify	97.44	95.13	86.56	89.8	86.94	–
	Multi-w-Lang_id	98.03	95.81	87.48	91.44	88.86	–
	Multi-w-Syntax	98.13	96.26	88.01	92.79	90.68	83.44
	Multi-w-Syntax+Semantic	97.55	90.07	87.88	94.22	88.19	73.55
	Multi-w-All	96.62	86.43	84.78	91.21	83.72	64.12
Russian-SynTagRus (size: 54099)	UDPipe	99.12	97.57	98.53	95.22	93.74	–
	UDify	97.46	89.3	93.8	97.35	95.3	–
	Multi-w-Lang_id	98.04	90.94	94.19	96.42	94.06	–
	Multi-w-Syntax	98.14	92.04	94.42	96.6	95.49	82.8
	Multi-w-Syntax+Semantic	96.47	91.56	85.24	97.58	95.59	73.68
	Multi-w-All	95.69	88.9	82.75	94.32	92.34	65.16
Russian-Taiga (size: 54099)	UDPipe	93.18	82.87	89.99	76.81	70.47	–
	UDify	95.39	88.47	90.19	85.05	78.83	–
	Multi-w-Lang_id	96.67	90.24	90.85	87.91	83.11	–
	Multi-w-Syntax	96.79	91.44	91.22	92.4	86.68	82.88
	Multi-w-Syntax+Semantic	94.62	84.64	81.78	93.23	76.18	74.81
	Multi-w-All	92.94	82.19	79.46	89.74	71.12	68.8
Serbian-SET (size: 3328)	UDPipe	98.33	94.35	97.36	93.68	90.25	–
	UDify	97.67	97.66	95.44	95.19	92.17	–
	Multi-w-Lang_id	98.18	97.92	95.71	94.34	93.51	–
	Multi-w-Syntax	98.28	98.09	95.87	95.15	92.72	81.84
	Multi-w-Syntax+Semantic	96.82	92.0	94.62	97.62	95.09	69.71
	Multi-w-All	96.12	90.24	91.3	93.18	88.4	62.96
Slovak-SNK (size: 8483)	UDPipe	96.83	90.82	96.4	90.77	87.85	–
	UDify	98.8	87.71	94.04	97.1	95.01	–

	Multi-w-Lang_id	98.94	89.61	94.42	96.52	94.48	–
	Multi-w-Syntax	99.02	90.89	94.63	97.19	94.18	82.57
	Multi-w-Syntax+Semantic	98.24	89.32	86.29	93.11	97.1	72.51
	Multi-w-All	96.77	86.76	82.84	87.17	90.83	61.05
Slovenian-SSJ (size: 8556)	UDPipe	98.61	95.92	98.25	93.75	91.95	–
	UDify	97.72	93.29	89.43	95.75	93.57	–
	Multi-w-Lang_id	98.89	94.4	96.7	94.78	93.36	–
	Multi-w-Syntax	98.97	95.03	96.81	96.14	93.33	88.09
	Multi-w-Syntax+Semantic	98.01	94.46	96.56	99.07	95.06	75.63
	Multi-w-All	96.0	92.62	92.86	93.82	92.85	66.03
Slovenian-SST (size: 8556)	UDPipe	93.79	86.28	95.17	74.89	68.89	–
	UDify	95.4	89.81	95.15	80.89	75.55	–
	Multi-w-Lang_id	96.67	91.36	95.45	86.08	79.37	–
	Multi-w-Syntax	96.79	92.41	95.61	90.85	84.94	82.51
	Multi-w-Syntax+Semantic	94.89	91.45	89.54	93.06	77.71	71.56
	Multi-w-All	93.25	89.92	87.1	89.35	72.92	66.05
Spanish-AnCora (size: 28492)	UDPipe	98.91	98.49	99.17	92.85	90.77	–
	UDify	98.53	97.89	98.07	94.72	92.23	–
	Multi-w-Lang_id	98.76	98.11	98.15	94.35	92.69	–
	Multi-w-Syntax	98.84	98.26	98.2	95.05	92.43	82.47
	Multi-w-Syntax+Semantic	98.49	94.01	90.46	86.57	82.26	74.49
	Multi-w-All	97.12	91.6	88.9	82.45	80.2	68.69
Spanish-GSD (size: 28492)	UDPipe	96.85	97.09	98.97	92.14	89.46	–
	UDify	97.1	89.7	91.6	92.22	88.69	–
	Multi-w-Lang_id	97.15	90.15	94.35	92.4	89.44	–
	Multi-w-Syntax	97.26	91.36	94.57	95.03	92.11	84.76
	Multi-w-Syntax+Semantic	96.27	88.24	86.95	88.19	90.98	77.8
	Multi-w-All	95.32	87.05	83.41	84.96	87.81	67.84
Swedish-LinES (size: 7479)	UDPipe	96.78	89.43	97.03	86.97	82.76	–
	UDify	96.83	88.89	89.33	91.31	86.21	–

	Multi-w-Lang_id	97.63	90.59	90.05	93.03	88.63	–
	Multi-w-Syntax	97.73	91.74	90.46	94.14	89.37	82.74
	Multi-w-Syntax+Semantic	96.32	85.52	88.97	95.04	81.05	72.38
	Multi-w-All	94.5	84.43	85.93	88.81	75.54	64.43
Swedish-Talbanken (size: 7479)	UDPipe	97.94	96.86	98.01	90.73	87.71	–
	UDify	98.48	95.81	98.08	92.92	90.61	–
	Multi-w-Lang_id	98.72	96.37	98.16	93.6	91.35	–
	Multi-w-Syntax	98.81	96.75	98.21	93.97	91.65	81.81
	Multi-w-Syntax+Semantic	97.59	91.2	95.15	85.2	88.69	72.94
	Multi-w-All	96.31	88.6	92.99	79.93	86.24	65.96
Tamil-TTB (size: 400)	UDPipe	91.05	87.28	93.92	74.37	66.63	–
	UDify	90.47	70.0	67.17	80.1	70.38	–
	Multi-w-Lang_id	93.4	76.35	82.58	86.07	75.59	–
	Multi-w-Syntax	93.57	79.4	83.33	89.93	83.32	91.56
	Multi-w-Syntax+Semantic	89.6	76.93	82.69	82.85	81.06	81.72
	Multi-w-All	89.4	75.18	81.16	78.94	77.48	71.06
Telugu-MTG (size: 1051)	UDPipe	93.07	99.03	100.0	92.74	86.5	–
	UDify	96.58	91.77	73.55	89.46	84.62	–
	Multi-w-Lang_id	95.39	99.3	99.72	94.52	87.82	–
	Multi-w-Syntax	95.53	99.28	99.93	95.94	90.11	98.97
	Multi-w-Syntax+Semantic	92.4	96.82	94.55	85.87	92.79	88.27
	Multi-w-All	91.8	93.97	90.45	80.01	86.46	81.44
Turkish-IMST (size: 3664)	UDPipe	96.01	92.55	96.01	75.11	68.48	–
	UDify	88.59	59.22	72.82	80.85	69.2	–
	Multi-w-Lang_id	92.14	65.81	74.75	84.77	76.11	–
	Multi-w-Syntax	92.33	70.26	75.85	89.8	81.3	84.15
	Multi-w-Syntax+Semantic	89.85	65.69	69.47	91.52	74.33	77.58
	Multi-w-All	88.44	64.48	65.52	89.05	71.95	71.7
Ukrainian-IU (size: 5496)	UDPipe	97.59	92.66	97.23	90.2	87.16	–
	UDify	98.02	89.67	95.34	95.3	91.01	–

	Multi-w-Lang_id	98.42	91.25	95.62	95.87	91.56	–
	Multi-w-Syntax	98.51	92.31	95.78	95.26	92.1	81.89
	Multi-w-Syntax+Semantic	96.75	91.34	94.79	96.5	81.13	75.05
	Multi-w-All	95.12	87.64	93.36	92.39	78.1	63.56
Urdu-UDTB (size: 4043)	UDPipe	93.66	81.92	97.4	89.41	83.53	–
	UDify	93.8	90.38	88.8	88.3	83.33	–
	Multi-w-Lang_id	95.61	91.84	89.56	89.56	86.99	–
	Multi-w-Syntax	95.74	92.82	89.99	93.24	89.31	83.37
	Multi-w-Syntax+Semantic	94.04	92.7	81.71	84.07	86.3	72.71
	Multi-w-All	93.4	90.32	79.35	80.98	83.0	61.73
Uyghur-UDT (size: 1656)	UDPipe	89.87	88.3	95.31	79.97	68.6	–
	UDify	75.88	70.8	79.7	67.78	50.69	–
	Multi-w-Lang_id	83.67	75.48	81.13	76.91	61.43	–
	Multi-w-Syntax	83.99	78.65	81.94	85.84	73.26	83.4
	Multi-w-Syntax+Semantic	83.15	72.27	79.1	76.14	74.65	77.17
	Multi-w-All	82.6	69.28	76.9	73.88	72.57	67.9
Vietnamese-VTB (size: 1400)	UDPipe	89.68	99.72	99.55	72.2	64.38	–
	UDify	85.59	65.49	77.18	75.29	64.18	–
	Multi-w-Lang_id	90.14	71.05	78.79	81.82	72.35	–
	Multi-w-Syntax	90.36	74.8	79.71	89.12	78.29	83.13
	Multi-w-Syntax+Semantic	88.68	68.43	71.27	91.0	70.24	73.17
	Multi-w-All	86.7	67.48	69.45	87.18	67.23	68.02
Corpus	**Model**	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**	**Typo F1**

Appendix C

Results of proposed End-to-end EDP model

This appendix compares the results achieved by our proposed *ED-parser* described in chapter 6 with the results achieved by the other participants of *IWPT 2021 Shared tasks* as table C.1. We summarised the results and provided key inferences drawn from these in sections

Table C.1: Results of all participants of *IWPT 2021 Shared Task*

Begin of Table							
Language	**Models**	**UPOS**	**UFeats**	**Lemma**	**UAS**	**LAS**	**ELAS**
Bulgarian	combo	98.72	97.23	97.25	92.98	89.52	86.67
	dcu-epfl	98.89	97.57	97.30	93.25	90.19	92.44
	fastparse	99.15	97.95	97.97	87.85	83.39	78.73
	grew	99.15	97.95	97.97	94.36	91.62	88.83
	robertnlp	99.13	98.31	0.01	96.30	94.15	93.16
	shanghaitech	0.00	35.92	0.01	5.80	1.54	92.52
	tgif	0.00	35.98	0.01	10.58	1.13	93.63
	unipi	98.81	97.57	97.40	95.29	92.71	90.84
	Base E2E	**98.81**	**35.97**	**97.4**	**93.37**	**90.03**	**78.45**
	Base E2E-w-Aux	**98.21**	**35.67**	**97.2**	**93.17**	**90.33**	**80.85**
	Base E2E-w-Typ	**99.01**	**35.97**	**98.1**	**93.87**	**90.63**	**81.85**
English	combo	95.74	93.54	95.26	89.61	87.22	84.09
	dcu-epfl	94.96	93.53	95.66	86.45	83.64	85.70
	fastparse	95.85	94.16	96.04	82.36	77.99	73.00
	grew	95.85	94.16	96.04	89.22	86.83	85.49
	robertnlp	96.24	94.44	0.00	90.79	88.48	87.88
	shanghaitech	0.28	32.80	0.00	3.71	1.24	87.27

	tgif	0.28	32.76	0.00	7.86	1.08	88.19
	unipi	95.17	93.70	95.76	90.64	88.47	87.11
	Base E2E	**95.17**	**32.77**	**95.76**	**87.07**	**84.46**	**75.4**
	Base E2E-w-Aux	**94.67**	**33.07**	**96.46**	**86.77**	**84.96**	**78.1**
	Base E2E-w-Typ	**95.37**	**33.47**	**96.76**	**87.57**	**85.46**	**78.8**
Estonian	combo	97.42	96.57	86.09	90.00	87.53	84.02
	dcu-epfl	96.46	95.30	95.58	85.31	82.35	84.35
	fastparse	96.89	95.78	94.90	71.70	64.50	60.05
	grew	96.89	95.78	94.90	86.62	83.85	78.19
	robertnlp	97.09	96.46	0.00	90.02	87.59	86.55
	shanghaitech	0.12	34.99	0.00	3.67	1.16	86.66
	tgif	0.12	35.08	0.01	11.86	0.82	88.38
	unipi	96.49	95.33	95.55	87.11	84.14	81.27
	Base E2E	**96.49**	**35.04**	**95.55**	**85.41**	**82.46**	**74.03**
	Base E2E-w-Aux	**96.59**	**35.34**	**96.25**	**85.71**	**83.06**	**76.33**
	Base E2E-w-Typ	**96.89**	**35.74**	**96.55**	**86.21**	**83.36**	**76.63**
Latvian	combo	97.35	94.97	96.53	92.91	90.25	84.57
	dcu-epfl	95.95	93.59	95.34	88.47	85.10	86.96
	fastparse	96.28	93.79	95.81	78.37	72.03	66.43
	grew	96.28	93.79	95.81	88.32	85.27	77.45
	robertnlp	97.61	95.18	0.03	93.62	91.25	88.82
	shanghaitech	0.58	35.57	0.03	4.22	1.42	89.17
	tgif	0.56	35.62	0.03	10.37	0.97	90.23
	unipi	96.12	93.45	95.45	89.90	86.63	83.01
	Base E2E	**96.12**	**35.61**	**95.45**	**88.51**	**85.19**	**76.67**
	Base E2E-w-Aux	**96.12**	**35.11**	**96.05**	**88.71**	**85.29**	**78.57**
	Base E2E-w-Typ	**96.62**	**35.91**	**96.55**	**89.51**	**85.89**	**78.97**
Lithuanian	combo	97.26	95.05	93.76	88.03	84.75	79.75
	dcu-epfl	93.47	87.74	92.71	78.36	73.25	78.04
	fastparse	95.97	91.07	93.61	61.39	53.55	48.27
	grew	95.97	91.07	93.61	82.54	78.65	74.62
	robertnlp	97.42	93.20	0.00	90.49	83.27	80.76
	shanghaitech	1.51	30.12	0.00	5.12	1.77	80.87
	tgif	1.51	30.20	0.00	10.89	1.24	86.06
	unipi	93.40	87.14	92.66	82.75	78.31	71.31
	Base E2E	**93.4**	**30.09**	**92.66**	**78.25**	**73.52**	**73.52**
	Base E2E-w-Aux	**93.5**	**29.99**	**93.26**	**78.25**	**73.72**	**76.92**

	Base E2E-w-Typ	**93.8**	**30.59**	**93.66**	**79.05**	**74.42**	**77.22**
Russian	combo	98.94	98.04	98.16	95.37	94.29	90.73
	dcu-epfl	98.19	87.67	97.39	92.61	90.97	92.83
	fastparse	98.86	88.97	98.33	87.09	83.23	78.56
	grew	98.86	88.97	98.33	94.22	92.97	90.56
	robertnlp	99.06	89.51	0.00	95.65	94.64	92.64
	shanghaitech	0.02	36.35	0.00	3.35	0.73	93.59
	tgif	0.02	36.37	0.00	13.81	0.51	94.01
	unipi	98.25	87.52	97.49	94.51	93.32	90.90
	Base E2E	**98.25**	**36.32**	**97.49**	**92.67**	**91.01**	**76.33**
	Base E2E-w-Aux	**97.75**	**36.02**	**97.89**	**92.57**	**91.61**	**78.83**
	Base E2E-w-Typ	**98.45**	**36.92**	**98.49**	**93.27**	**92.01**	**79.53**
Slovak	combo	97.88	95.03	95.61	93.19	91.72	87.04
	dcu-epfl	96.55	91.15	94.72	89.27	86.60	89.59
	fastparse	97.67	93.42	96.47	78.23	71.71	64.28
	grew	97.67	93.42	96.47	92.27	90.45	86.92
	robertnlp	98.28	95.54	0.00	96.16	93.88	89.66
	shanghaitech	1.19	22.69	0.00	6.06	1.96	90.25
	tgif	1.17	22.69	0.00	13.67	1.60	94.96
	unipi	96.62	91.44	94.61	93.32	91.75	86.05
	Base E2E	**96.62**	**22.68**	**94.61**	**90.09**	**87.49**	**77.45**
	Base E2E-w-Aux	**96.32**	**22.68**	**94.81**	**90.29**	**87.39**	**80.85**
	Base E2E-w-Typ	**96.92**	**23.48**	**95.71**	**90.89**	**88.19**	**81.15**
Swedish	combo	97.67	89.19	92.45	90.31	87.82	83.20
	dcu-epfl	96.12	87.92	92.47	85.83	82.30	85.20
	fastparse	97.25	88.82	93.60	78.88	73.11	67.26
	grew	97.25	88.82	93.60	89.26	86.59	81.54
	robertnlp	98.30	89.87	0.00	92.15	89.92	88.03
	shanghaitech	0.00	33.79	0.00	1.55	0.34	86.62
	tgif	0.00	33.79	0.00	8.42	0.20	89.90
	unipi	96.07	87.83	92.47	90.86	88.53	84.91
	Base E2E	**96.05**	**33.56**	**92.46**	**85.64**	**82.18**	**73.12**
	Base E2E-w-Aux	**95.75**	**33.06**	**92.66**	**86.04**	**82.08**	**75.52**
	Base E2E-w-Typ	**96.45**	**34.06**	**93.06**	**86.54**	**82.78**	**76.02**
Language	**Models**	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**	**ELAS**

Bibliography

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Anne Abeillé, Lionel Clément, and François Toussenel. Building a treebank for french. In *Treebanks*, pages 165–187. Springer, 2003.

Salim Abu-Rabia and Ekaterina Sanitsky. Advantages of bilinguals over monolinguals in learning a third language. *Bilingual Research Journal*, 33(2):173–199, 2010.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 206–211, 2013.

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, 2017.

Željko Agić, Jörg Tiedemann, Danijela Merkler, Simon Krek, Kaja Dobrovoljc, and Sara Može. Cross-lingual dependency parsing of related languages with rich morphosyntactic tagsets. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pages 13–24, 2014.

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 *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, 2015.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. Generating high quality proposition banks for

multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, 2015.

Aho Alfred and Jeffrey Ullman. The theory of parsing, translation and compiling. 1972.

Sture Allén. *Text Processing: Text Analysis and Generation, Text Typology and Attribution: Proceedings of Nobel Symposium 51.* Number 16. Coronet Books, 1982.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. Transferring semantic roles using translation and syntactic information. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2017.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. Cross-lingual transfer of semantic roles: From raw text to semantic roles. *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, 2019.

Waleed Ammar. *Towards a Universal Analyzer of Natural Languages.* PhD thesis, Ph. D. thesis, Google Research, 2016.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444, 2016.

Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Translation artifacts in cross-lingual transfer learning. *arXiv preprint arXiv:2004.04721*, 2020.

Ehsaneddin Asgari and Hinrich Schütze. Past, present, future: A computational investigation of the typology of tense in 1000 languages. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. Linear neural parsing and hybrid enhancement for enhanced universal dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 206–214, 2020.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

James K Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132, 1979.

Mark C Baker. *The atoms of language: The mind's hidden rules of grammar*. Basic books, 2008.

Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. Multilingual subjectivity analysis using machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 127–135, 2008.

ICSI Barkley. English framenet. `https://framenet.icsi.berkeley.edu/fndrupal/`.

James Barry, Joachim Wagner, and Jennifer Foster. The adapt enhanced dependency parser at the iwpt 2020 shared task. *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies2*, 2020.

Regina Barzilay and Yuan Zhang. Hierarchical low-rank tensors for multilingual transfer parsing. Association for Computational Linguistics, 2015.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. Textual inference and meaning representation in human robot interaction. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 65–69, 2013.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Emily M Bender. Linguistic typology in natural language processing. *Linguistic Typology*, 20(3):645–660, 2016.

Emily M Bender, Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In *Proceedings of the 7th workshop on language technology for cultural heritage, social sciences, and humanities*, pages 74–83, 2013.

Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.

Yevgeni Berzak, Roi Reichart, and Boris Katz. Contrastive analysis with predictive power: Typology driven estimation of grammatical error distributions in esl. *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 2016.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. The hindi/urdu treebank project. In *Handbook of Linguistic Annotation*, pages 659–697. Springer, 2017.

Eckhard Bick. Arboretum, a hybrid treebank for danish. In *Proceedings of TLT 2003 (2nd Workshop on Treebanks and Linguistic Theory, Växjö*, pages 9–20, 2003.

Eckhard Bick, Heli Uibo, and Kadri Muischnek. Preliminary experiments for a cg-based syntactic tree corpus of estonian. `https://corp.hum.sdu.dk/tgrepeye_est.html`.

Balthasar Bickel. Typology in the 21st century: Major current developments. *Linguistic Typology*, 11(1):239–251, 2007.

Balthasar Bickel, Johanna Nichols, Taras Zakharko, Alena Witzlack-Makarevich, Kristine Hildebrandt, Michael Rießler, Lennart Bierkandt, Fernando Zúñiga, and John B Lowe. The autotyp typological databases. *Version 0.1. 0. Available online at: https://github. com/autotyp/autotyp-data/tree/0.1. 0*, 2017.

Philip Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.

Johannes Bjerva and Isabelle Augenstein. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Johannes Bjerva, Elizabeth Salesky, Sabrina Mielke, Aditi Chaudhary, Giuseppe G. A. Celano, Edoardo M. Ponti, Ekaterina Vylomova, Ryan Cotterell, and Isabelle Augenstein. SIGTYP 2020 Shared Task: Prediction of Typological Features. In *Proceedings of the Second Workshop on Computational Research in Linguistic Typology*. Association for Computational Linguistics, 2020.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Taylor L Booth. Probabilistic representation of formal languages. In *10th annual symposium on switching and automata theory (swat 1969)*, pages 74–81. IEEE, 1969.

Gosse Bouma, Djamé Seddah, and Daniel Zeman. Overview of the iwpt 2020 shared task on parsing into enhanced universal dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 151–161, 2020.

Melissa Bowerman, Stephen C Levinson, and Stephen Levinson. *Language acquisition and conceptual development*. Number 3. Cambridge University Press, 2001.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. Iglue: A benchmark for transfer learning across modalities, tasks, and languages. In *International Conference on Machine Learning*, pages 2370–2392. PMLR, 2022.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. Framenet for the semantic analysis of german: Annotation, representation and automation. *Multilingual FrameNets in Computational Lexicography: methods and applications*, 200:209–244, 2009.

Jan Buys and Phil Blunsom. Robust incremental neural semantic graph parsing. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

Rui Cai and Mirella Lapata. Alignment-free cross-lingual semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3883–3894, 2020.

R Caruana. *Multitask Learning. Autonomous Agents and Multi-Agent Systems.* PhD thesis, Carnegie Mellon University, 1998.

Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. *Advances in neural information processing systems*, 27, 2014.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 280–287, 2007.

Eugene Charniak et al. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, 2016.

Ciprian Chelba and Frederick Jelinek. Structured language modeling. *Computer Speech & Language*, 14(4):283–332, 2000.

Wenliang Chen, Min Zhang, Yoshimasa Tsuruoka, Yujie Zhang, Yiou Wang, Kentaro Torisawa, Haizhou Li, et al. Smt helps bitext dependency parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 73–83, 2011.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. Multi-source cross-lingual model transfer: Learning what to share. *ACL*, 2018.

Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

Noam Chomsky. Tool module, chomskys universal grammar (retrieved 2010-10-07). `https://thebrain.mcgill.ca/flash/capsules/outil_rouge06.html`, 1960.

Noam Chomsky. *Lectures on government and binding: The Pisa lectures.* Number 9. Walter de Gruyter, 1993.

Janara Christensen, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120, 2011.

Yoeng-Jin Chu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014.

Guglielmo Cinque and Luigi Rizzi. The cartography of syntactic structures. *Oxford Handbook of linguistic analysis*, 2010.

Manuel R Ciosici, Leon Derczynski, and Ira Assent. Quantifying the morphosyntactic content of brown clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1541–1550, 2019.

Shay Cohen. Bayesian analysis in natural language processing. *Synthesis Lectures on Human Language Technologies*, 9(2):1–274, 2016.

Shay B Cohen, Dipanjan Das, and Noah A Smith. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, 2011.

Reed Coke, Ben King, and Dragomir Radev. Classifying syntactic regularities for hundreds of languages. *URL:https://dblp.org/db/journals/corr/corr1603.html#CokeKR16*, 2016.

Chris Collins and Richard Kayne. Syntactic structures of the world's languages. *New York: New York University*, 2009.

Bernard Comrie. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press, 1989.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *Proceedings of the 6th International Conference on Learning Representations*, 2018a.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2018b.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2019.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332, 2005.

Ryan Cotterell and Jason Eisner. Probabilistic typology: Deep generative models of vowel inventories. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

Ryan Cotterell and Jason Eisner. A deep generative model of vowel formant typology. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Michael A Covington. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th annual ACM southeast conference*, volume 1. Citeseer, 2001.

William Croft. *Typology and universals*. Cambridge University Press, 2002.

James Cross and Liang Huang. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176, 2016.

Roy G d'Andrade. *The development of cognitive anthropology*. Cambridge University Press, 1995.

Zeman Daniel, Popel Martin, Straka Milan, Hajic Jan, Nivre Joakim, Ginter Filip, Luotolahti Juhani, Pyysalo Sampo, Petrov Slav, Potthast Martin, et al. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, volume 1, pages 1–19. Association for Computational Linguistics, 2017.

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

Hal Daumé III and Lyle Campbell. A bayesian model for discovering typological implications. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2009.

Angel Daza and Anette Frank. Translate and label! an encoder-decoder approach for cross-lingual semantic role labeling. *Translate and Label! An Encoder-Decoder Approach for Cross-lingual Semantic Role Labeling*, 2019.

Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pages 1–8, 2008.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Lrec*, volume 6, pages 449–454, 2006.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, 2014.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. Universal dependencies. *Computational linguistics*, 47(2):255–308, 2021.

Mathieu Dehouck, Mark Anderson, and Carlos Gómez-Rodríguez. Efficient eud parsing. *Proceedings of the 16th International Conference on Parsing Technologies and*

the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Robert MW Dixon. Where have all the adjectives gone? *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 1 (1):19–80, 1977.

Robert MW Dixon and Robert MW Dixon. *Ergativity*. Cambridge University Press, 1994.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016.

Matthew S Dryer and Martin Haspelmath. Wals online. max planck institute for evolutionary anthropology. `https://wals.info/`, 2013.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pages 845–850, 2015a.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. A neural network model for low-resource universal dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 339–348, 2015b.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning cross-lingual word embeddings without bilingual corpora. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.

Greg Durrett, Adam Pauls, and Dan Klein. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, 2012.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

Adam Ek and Jean-Philippe Bernardy. How much of enhanced ud is contained in ud? In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 221–226, 2020.

Mohab Elkaref and Bernd Bohnet. A simple lstm model for transition-based dependency parsing. *arXiv preprint arXiv:1708.08959*, 2017.

Agnieszka Falenska and Özlem Çetinoğlu. Lexicalized vs. delexicalized parsing in low-resource scenarios. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 18–24, 2017.

Meng Fang and Trevor Cohn. Model transfer for tagging low-resource languages using a bilingual dictionary. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017.

Hao Fei, Meishan Zhang, and Donghong Ji. Cross-lingual semantic role labeling with high-quality translated training corpus. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Janet Dean Fodor. Setting syntactic parameters. 2001.

Janet Dean Fodor and William Gregory Sakas. Evaluating models of parameter setting. In *Proceedings of the 28th annual boston university conference on language development*, volume 1, pages 1–27. Cascadilla Press Somerville, MA, 2004.

G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

W Nelson Francis and Henry Kucera. Brown corpus manual. *Letters to the Editor*, 5 (2):7, 1979.

David M Gaddy, Yuan Zhang, Regina Barzilay, and Tommi S Jaakkola. Ten pairs to tag-multilingual pos tagging via coarse mapping between embeddings. Association for Computational Linguistics, 2016.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049, 2010.

Ryan Georgi, Fei Xia, and William Lewis. Comparing language similarity across genetic and typologically-based groupings. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 385–393, 2010.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

Rob Goedemans, Jeffrey Heinz, and Harry Van der Hulst. Stresstyp2. *University of Connecticut, University of Delaware, Leiden University, and the US National Science Foundation*, 2014.

Raúl Gómez. Understanding categorical cross-entropy loss, binary cross-entropy loss, softmax loss, logistic loss, focal loss and all those confusing names. *URL: https://gombru. github. io/2018/05/23/cross_ entropy_loss/(visited on 29/03/2019)*, 2018.

Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. In *HLT-NAACL*, pages 1386–1390, 2015.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *International Conference on Machine Learning*, pages 748–756. PMLR, 2015.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Harvey J Greenberg. Greedy algorithms for minimum spanning tree. *University of Colorado at Denver*, 1998.

Joseph H Greenberg. Language universals: A research frontier: Empirical limits of logically possible types provide a basic method for linguistic generalization. *Science*, 166(3904):473–478, 1969.

Joseph H Greenberg et al. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113, 1963.

Stefan Grünewald, Prisca Piccirilli, and Annemarie Friedrich. Coordinate constructions in english enhanced universal dependencies: Analysis and computational modeling. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, 2015.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. A universal framework for inductive transfer parsing across multi-typed treebanks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 12–22, 2016a.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. A representation learning framework for multi-source transfer parsing. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016b.

Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *Proceedings of the 13th International Conference on Spoken Language Translation*, 2016.

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, and Maria Antonia Martı. Lluıs marquez, adam meyers, joakim nivre, sebastian padó, jan štepánek, pavel stranák, mihai surdeanu, nianwen xue, and yi zhang. 2009. the conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, 2009.

Harald Hammarström, Robert Forkel, and Martin Haspelmath. Glottolog 3.0. *Max Planck Institute for the Science of Human History*, 2017.

Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

Aaron Li-Feng Han, Derek F Wong, Lidia S Chao, Yi Lu, Liangye He, and Liang Tian. A universal phrase tagset for multilingual treebanks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 247–258. Springer, 2014.

Chunghye Han, Narae Han, Eonsuk Ko, and Martha Palmer. Korean treebank: Development and evaluation. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, 2002.

Iren Hartmann, Martin Haspelmath, and Bradley Taylor. Valency patterns leipzig (database). 2013.

Martin Haspelmath. *The typological database of the World Atlas of Language Structures*. Berlin: Walter de Gruyter, 2009.

Martin Haspelmath and Uri Tadmor. Wold. max planck institute for evolutionary anthropology. `https://wold.clld.org/`, 2009.

Han He and Jinho D Choi. Adaptation of multilingual transformer encoder for robust enhanced universal dependency parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 181–191, 2020.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, 2017.

Johannes Heinecke. Hybrid enhanced universal dependencies parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 174–180, 2020.

James Henderson. Discriminative training of a neural network statistical parser. In *ACL'04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 95–102. Association for Computational Linguistics, 2004.

Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *Proceedings of the 2nd International Conference on Learning Representations*, 2013.

Daniel Hershcovich, Miryam de Lhoneux, Artur Kulmizev, Elham Pejhan, and Joakim Nivre. Kopsala: Transition-based graph parsing via efficient training and effective encoding. *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016a.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679, 2016b.

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*, pages 1222–1231, 2009.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–325, 2005.

Wenbin Jiang and Qun Liu. Dependency parsing and projection based on word-pair classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 12–20, 2010.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.

Audrey Joseph Aoun Yen-Hui and Samuel Jay Keyser. *Principles and parameters in comparative grammar*, volume 20. MIT Press, 1991.

Backus JW. c the syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In *Proceedings of the International Conference of Information Processing UNESCO Paris June*, 1959.

Aishwarya Kamath and Rajarshi Das. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*, 2018.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. Turku enhanced parser pipeline: From raw text to enhanced graphs in the iwpt 2020 shared task. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 162–173, 2020.

Ronald M Kaplan. A general syntactic processor. *Natural language processing*, (8): 194–241, 1973.

Yasuhiro Kawata and Julia Bartels. Stylebook for the japanese treebank in verbmobil. In *Verbmobil-Report 240, Seminar für Sprachwissenschaft, Universität Tübingen*, 2000.

Richard Kayne. Some notes on comparative syntax, with special reference to english and french. In *The Oxford handbook of comparative syntax*. Oxford University Press, 2012.

Mary Ritchie Key and Bernard Comrie. Ids. leipzig: Max planck institute for evolutionary anthropology. http://www.ids-leipzig.de/, 2015.

Atif Khan, Naomie Salim, and Yogan Jaya Kumar. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, 30:737–747, 2015.

Mitesh M Khapra, Salil Joshi, Arindam Chatterjee, and Pushpak Bhattacharyya. Together we can: Bilingual bootstrapping for wsd. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 561–569, 2011.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Paul Kingsbury and Martha Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer, 2003.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.

Ömer Kırnap, Erenay Dayanık, and Deniz Yuret. Tree-stack lstm in transition based dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 124–132, 2018.

Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2018.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, 2012.

Philipp Koehn et al. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):531–552, 2010.

Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing universal dependencies universally. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019a.

Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing universal dependencies universally. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019b.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

Mikhail Kozhevnikov and Ivan Titov. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200, 2013.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. What do recurrent neural network grammars learn about syntax? *arXiv preprint arXiv:1611.05774*, 2016.

George Lakoff et al. Frame semantic control of the coordinate structure constraint. 1986.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *URL:https://dblp.org/rec/conf/iclr/LanCGGSS20.html*, 2019.

Stanislas Lauly, Alex Boulanger, and Hugo Larochelle. Learning multilingual word representations using a bag-of-words autoencoder. *arXiv preprint arXiv:1401.1803*, 2014.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2019.

William Lewis and Fei Xia. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008.

Zhenghua Li, Min Zhang, and Wenliang Chen. Soft cross-lingual syntax projection for dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 783–793, 2014.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, 2018.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*, 2021.

Patrick Littel, David R Mortensen, and Lori Levin. Uriel typological database. *Pittsburgh: CMU*, 2016.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, 2017.

Haitao Liu. Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6):1567–1578, 2010.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. 2015.

Xia Lu. Exploring word order universals: A probabilistic graphical model approach. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 150–157, 2013.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.

Anatole Lyovin et al. *An Introduction to the Languages of the World*. Oxford University Press, USA, 1997.

Gary F. Simons M. Paul Lewis and Charles D. Fennig. Ethnologue: Languages of the World, Eighteenth edition, 2015.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo, 2004.

Ian Maddieson, Sébastien Flavier, Egidio Marsico, Christophe Coupé, and François Pellegrino. Lapsyd: Lyon-albuquerque phonological systems database. *INTERSPEECH-2013*, 2013.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*, 2020.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. Learning language representations for typology prediction. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

Glenn K Manacher. An improved version of the cocke-younger-kasami algorithm. *Computer Languages*, 3(2):127–133, 1978.

Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(2), 2010.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

Daniel Jurafsky James H. Martin. Constituency parsing. In *Speech and Language Processing*, chapter 13. 2021a.

Daniel Jurafsky James H. Martin. Dependency parsing. In *Speech and Language Processing*, chapter 14. 2021b.

Ryan McDonald and Joakim Nivre. Yvonne irmbach-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.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, 2005.

Ryan McDonald, Slav Petrov, and Keith B Hall. Multi-source transfer of delexicalized dependency parsers. 2011.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, 2013.

Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

Susanne Maria Michaelis, Philippe Maurer, Martin Haspelmath, and Magnus Huber. *The atlas of pidgin and creole language structures*. Oxford University Press, 2013.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Alireza Mohammadshahi and James Henderson. Syntax-aware graph-to-graph transformer for semantic role labelling. *Proceedings of the 2019 Conference on Empirical*

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2021.

Mehryar Mohri and Fernando CN Pereira. Dynamic compilation of weighted context-free grammars. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 891–897, 1998.

Simonetta Montemagni, Francesco Barsotti, Marco Battista, Nicoletta Calzolari, Ornella Corazzari, Antonio Zampolli, Francesca Fanciulli, Maria Massetani, Remo Raffaelli, Roberto Basili, et al. The italian syntactic-semantic treebank: Architecture, annotation, tools and evaluation. 2003.

Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

Steven Moran, Daniel McCloy, and Richard Wright. Phoible online. 2014.

Antonio Moreno, Susana López, and Manuel Alcántara. Spanish tree bank: Specifications, version 5. *Technical paper*, 1999.

Nikola Mrkšić, Ivan Vulić, Diarmuid O Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the association for Computational Linguistics*, 5:309–324, 2017.

Phoebe Mulcaire, Jungo Kasai, and Noah A Smith. Low-resource parsing with crosslingual contextualized representations. *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019.

Yugo Murawaki. Diachrony-aware induction of binary latent representations from typological features. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 451–461, 2017.

Allan H Murphy. The finley affair: A signal event in the history of forecast verification. *Weather and forecasting*, 11(1):3–20, 1996.

Tahira Naseem, Regina Barzilay, and Amir Globerson. Selective sharing for multilingual dependency parsing. The Association for Computational Linguistics, 2012.

Phuong Thai Nguyen, Xuan Luong Vu, Thi Minh Huyen Nguyen, Hong Phuong Le, et al. Building a large syntactically-annotated corpus of vietnamese. 2009.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. Wider context by using bilingual language models in machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206, 2011.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Svetoslav Marinov. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 221–225, 2006a.

Joakim Nivre, Jens Nilsson, and Johan Hall. Talbanken05: A swedish treebank with phrase structure and dependency annotation. In *LREC*, pages 1392–1395, 2006b.

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 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, 2007.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

Mohammad Norouzi, David J Fleet, and Russ R Salakhutdinov. Hamming distance metric learning. *Advances in neural information processing systems*, 25, 2012.

Robert Östling. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, 2015.

Robert Östling and Jörg Tiedemann. Continuous multilinguality with language vectors. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2016.

Sebastian Padó and Mirella Lapata. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340, 2009.

Nikolaos Pappas and Andrei Popescu-Belis. Multilingual hierarchical attention networks for document classification. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

Douglas B Paul and Janet Baker. The design for the wall street journal-based csr corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Asya Pereltsvaig. *Languages of the World*. Cambridge University Press, 2020.

Jacob Persson, Richard Johansson, and Pierre Nugues. Text categorization using predicate-argument structures. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 142–149, 2009.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2011.

Steven Pinker. *The language instinct: The new science of language and mind*, volume 7529. Penguin UK, 1995.

Frans Plank and Elena Filimonova. The universals archive: A brief introduction for prospective users. *STUF-Language Typology and Universals*, 53(1):109–123, 2000.

Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

Edoardo Ponti, Roi Reichart, Anna-Leena Korhonen, and Ivan Vulic. Isomorphic transfer of syntactic structures in cross-lingual nlp. Association for Computational Linguistics, 2018a.

Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. Adversarial propagation and zero-shot cross-lingual transfer of word vector specialization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018b.

Edoardo Maria Ponti, Helen O'horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601, 2019.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Mohammad Sadegh Rasooli and Michael Collins. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293, 2017.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 147–155, 2009.

Ian Roberts and Anders Holmberg. On the role of parameters in universal grammar: A reply to newmeyer. *Organizing Grammar: Linguistic Studies in Honor of Henk van Riemsdijk. Berlin: Mouton de Gruyter*, pages 538–553, 2005.

Rudolf Rosa and Zdeněk Žabokrtskỳ. Klcpos3-a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, 2015.

Denis Rothman. *Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more*. Packt Publishing Ltd, 2021.

Guy Rotman, Ivan Vulić, and Roi Reichart. Bridging languages through images with deep partial canonical correlation analysis. 2018.

Rishiraj Saha Roy, Rahul Katare, Niloy Ganguly, and Monojit Choudhury. Automatic discovery of adposition typology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1037–1046, 2014.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. Unsupervised cross-lingual representation learning. In *Proceedings of ACL 2019, Tutorial Abstracts*, pages 31–38, 2019a.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019b.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *The 46th International Symposium on Computer Architecture*, 2019.

Patrick Schone and Daniel Jurafsky. Language-independent induction of part of speech class labels using only language universals. *Machine Learning: Beyond Supervision*, 2001.

Sebastian Schuster and Christopher D Manning. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, 2016.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.

Ehsan Shareghi, Yingzhen Li, Yi Zhu, Roi Reichart, and Anna Korhonen. Bayesian learning for neural dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3509–3519, 2019.

Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 12–21, 2007.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Tianze Shi and Lillian Lee. Tgif: Tree-graph integrated-format parser for enhanced ud with two-stage generic-to individual-language finetuning. *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, 2021.

Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

Ur Shlonsky. The cartographic enterprise in syntax. *Language and linguistics compass*, 4(6):417–429, 2010.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. Building a tree-bank of modern hebrew text. *Traitement Automatique des Langues*, 42(2): 247–380, 2001.

Kiril Simov and Petya Osenova. Btb-tr04: Bultreebank morphosyntactic annotation of bulgarian texts. Technical report, Technical Report BTB-TR04, Bulgarian Academy of Sciences, 2004.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. Xlda: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471*, 2019.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. *Proceedings of the fifth conference on Applied natural language processing*, 1997.

David A Smith and Jason Eisner. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 822–831, 2009.

Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of acl-08: hlt*, pages 737–745, 2008.

Anders Søgaard and Julie Wulff. An empirical etudy of non-lexical extensions to delexicalized transfer. In *Proceedings of COLING 2012: Posters*, pages 1181–1190, 2012.

Milan Straka. Udpipe 2.0 prototype at conll 2018 ud shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, 2018.

Milan Straka, Jana Straková, and Jan Hajič. Evaluating contextualized embeddings on 54 languages in pos tagging, lemmatization and dependency parsing. *Proceedings of 1st Workshop on Language Technologies for Historical and Ancient Languages*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A Smith. Greedy, joint syntactic-semantic parsing with stack lstms. *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. In *The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012)*, 2012.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, 2013.

Hiroya Takamura, Ryo Nagata, and Yoshifumi Kawasaki. Discriminative analysis of linguistic features for typological study. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 69–76, 2016.

Leonard Talmy. Path to realization: A typology of event conflation. In *Annual Meeting of the Berkeley Linguistics Society*, volume 17, pages 480–519, 1991.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *Lrec*, 2008.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. The penn treebank: an overview. *Treebanks*, pages 5–22, 2003.

Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

Yee Teh, Hal Daume III, and Daniel M Roy. Bayesian agglomerative clustering with coalescents. *Advances in neural information processing systems*, 20, 2007.

Ivan Titov and Alexandre Klementiev. Crosslingual induction of semantic roles. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 647–656, 2012.

Michael Tomasello. Beyond formalities: The case of language acquisition. *The Linguistic Review*, 22(2-4):183–197, 2005.

Hungarian Szeged Treebank. Szeged treebank 2.0: A hungarian natural language database with detailed syntactic analysis. *Hungarian linguistics at the University of Szeged*.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *Proceedings of the Eighth International Conference on Learning Representations*, 2019.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

David Vilares, Carlos Gómez-Rodríguez, and Miguel A Alonso. One model, two languages: training bilingual parsers with harmonized treebanks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2015.

Ivan Vulic and Marie-Francine Moens. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, volume 2, pages 719–725. ACL; East Stroudsburg, PA, 2015.

Ivan Vulić, Wim De Smet, and Marie Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 479–484, 2011.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

Bernhard Wälchli and Michael Cysouw. Lexical typology through similarity semantics: Toward a semantic map of motion verbs. *Linguistics*, 50(3):671–710, 2012.

James A Walker. Sali a. tagliamonte, roots of english: Exploring the history of dialects. cambridge and new york: Cambridge university press, 2013. pp. xv+ 153. isbn 978-0-521-68189-6. *English Language & Linguistics*, 18(3):568–572, 2014.

Dingquan Wang and Jason Eisner. The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, 4:491–505, 2016.

Dingquan Wang and Jason Eisner. Fine-grained prediction of syntactic typology: Discovering latent structure with supervised learning. *Transactions of the Association for Computational Linguistics*, 5:147–161, 2017.

Hai Wang and Hoifung Poon. Deep probabilistic logic: A unifying framework for indirect supervision. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Mengqiu Wang and Christopher D Manning. Cross-lingual projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics*, 2:55–66, 2014.

Xinyu Wang, Yong Jiang, and Kewei Tu. Enhanced universal dependency parsing with second-order inference and mixture of training data. *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, 2020.

Yaqing Wang and Quanming Yao. Few-shot learning: A survey. 2019.

Søren Wichmann. Genealogical classification in historical linguistics. In *Oxford Research Encyclopedia of Linguistics*. 2017.

Søren Wichmann, André Müller, Viveka Velupillai, Cecil H Brown, Eric W Holman, Pamela Brown, Sebastian Sauppe, Oleg Belyaev, Matthias Urban, Zarina Molochieva, et al. The asjp database (version 16). *Leipzig: Max Planck Institute for Evolutionary Anthropology*, 2013.

Andrew Wilson, Paul Rayson, and AM McEnery. Corpus linguistics by the lune: a festschrift for geoffrey leech. 2003.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1779–1785, 2014.

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Wilhelm Max Wundt. *Völkerpsychologie: Entwicklungsgesetze von Sprache, Mythus und Sitte.* 1900.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

Min Xiao and Yuhong Guo. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129, 2014.

Min Xiao and Yuhong Guo. Annotation projection-based representation learning for cross-lingual dependency parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 73–82, 2015.

Bright Xu. Nlp chinese corpus: Large scale chinese corpus for nlp, 2019.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2): 207–238, 2005.

Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of the eighth international conference on parsing technologies*, pages 195–206, 2003.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*, 2016.

David Yarowsky and Grace Ngai. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

Liping You and Kaiying Liu. Building chinese framenet database. In *2005 international conference on natural language processing and knowledge engineering*, pages 301–306. IEEE, 2005.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *LREC*, volume 2008, pages 28–30, 2008.

Daniel Zeman and Jan Hajic. Fgd at mrp 2020: prague tectogrammatical graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 33–39, 2020.

Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, 2008.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21, 2018.

Yuan Zhang, Roi Reichart, Regina Barzilay, and Amir Globerson. Learning to map into a universal pos tagset. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1368–1378, 2012.

Yue Zhang, Rui Wang, and Luo Si. Syntax-enhanced self-attention-based semantic role labeling. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer, 2014.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. A closer look at few-shot crosslingual transfer: The choice of shots matters. 2021.

Guangyou Zhou, Tingting He, Jun Zhao, and Wensheng Wu. A subspace learning framework for cross-lingual sentiment classification with partial parallel data. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.