From treebanks to partial syntactic structure induction

Marine Courtin

Université Sorbonne Nouvelle

École doctorale 622 - Sciences du langage

Laboratoire de Phonétique et Phonologie

Thèse de doctorat en Sciences du Langague

Marine Courtin

From treebanks to partial syntactic structure induction

Sous la direction de Martine Adda-Decker et Kim Gerdes

Soutenue le 31/01/2024

Comité de jury:

Marie CANDITO	MCF HDR (LLF, U Paris Cité)	Rapporteur
Ramon FERRER-I-CANCHO	PR (U Politècnica de Catalunya, Barcelone)	Rapporteur
Delphine BERNHARD	MCF (LiLPa, CNRS U de Strasbourg)	Examinateur
Sylvain KAHANE	PR (Modyco, CNRS U Paris Nanterre)	Examinateur
Martine ADDA-DECKER	DR (LPP, CNRS U Sorbonne Nouvelle)	Dir de thèse
Kim GERDES	PR (LISN, CNRS U Paris Saclay)	Dir de thèse

Titre Des corpus arborés à l'induction de structures syntaxiques partielles.

Résumé
Nos travaux portent sur les *treebanks*, ces corpus de textes dotés d'annotations de structures syntaxiques. Ils sont très utiles dans de nombreux domaines, de la linguistique au traitement automatique de la langue. Après une introduction portant sur leur rôle dans des domaines variés, nous plongeons dans l'histoire de leur création, depuis les pratiques d'annotation manuelle de textes vers les treebanks modernes avec l'avènement des technologiques. Le chapitre 3 montre les méthodes de création de ces treebanks. Le chapitre 4 discute des problématiques liées à la constitution des guides d'annotation, et mets en évidences certaines de ces problématiques au travers de deux études, la première portant sur traitement des expressions multi-mots, la seconde sur la constitution d'un treebank dans une langue peu pourvue en ressources, le Naija langue parlée au Nigéria étudiée dans le cadre du projet ANR NaijaSynCor. Le chapitre 5 présente l'outil Arborator-Grew, conçu pour faciliter l'annotation collaborative des treebanks. Le chapitre 6 étudie comment des lois linguistiques fondamentales comme la loi de *Menzerath-Altmann* et le *Heavy Constituent Shift* interagissent. Il propose également plusieurs procédures pour générer des arbres artificiels, permettant de contraster leurs propriété avec celles des arbres syntaxiques. Enfin, le chapitre 7 vise à utiliser des techniques statistiques pour découvrir la structure sous-jacente des phrases dans un texte. En résumé, ce travail montre l'importance des treebanks dans notre compréhension des langues, et leur rôle dans le développement des technologies linguistiques en soulignant l'innovation continue dans ce domaine.

Mots-clés: Traitement Automatique de la Langue, corpus arborés, syntaxe de dépendance, annotation syntaxique.

Title From treebanks to partial syntactic structure induction.

Abstract
This document focuses on *treebanks*, textual corpora with syntactic annotations. These treebanks are invaluable in numerous fields, ranging from linguistic studies to natural language processing. First, we explore how these treebanks aid researchers in multiple domains. Next, we dive into the history of treebank development. Before the computer age, researchers began to manually create collections of annotated texts, which evolved with the advent of computers into modern treebanks. Chapter 3 focuses on the challenges and methods of creating these treebanks. Chapter 4 addresses challenges relating to developing annotation guidelines. These discussions bring us to two case studies, the first relating to how to best handle complex multi-word expressions, the second retracing the development of a treebank for a low-resource language, the Naija pidgin-creole spoken spoken in Nigeria and its analysis as part of the ANR NaijaSynCor project. Chapter 5 introduces a new tool, Arborator-Grew, designed to facilitate collaborative annotation of treebanks. Chapter 6 studies how linguistic laws such as the *Menzerath-Altmann* law and the *Heavy Constituent Shift* interact. It also introduces several tree generation algorithms, which we use to contrast the properties of syntactic and artificial trees. Finally, Chapter 7 aims to use statistical techniques to latent structure of sentences in a text. In summary, this work highlights the importance of treebanks in our understanding of languages and their significant role in the development of language technologies. It also emphasizes continuous innovation in this field, opening new avenues for the study and analysis of languages.

Key-words: Natural Language Processing, treebanks, dependency syntax, syntactic annotation.

Acknowledgments

Thank you to the members of my jury for having accepted to evaluate my work, and for making my defense an enjoyable moment.

I'm very greatful for my supervisors Kim Gerdes and Martine Adda-Decker. Thank you for your time, patience and understanding while I wrote this thesis. Martine, thank you for learning so much about dependency syntax for my sake, you often provided useful feedback when I missed the big picture. Kim, I will miss our time in the office rue des Bernardins where we had so many interesting discussions, usually resulting in a whiteboard full of illegible squibbles.

This thesis would have never happened without Sylvain and his continued support throughout the years. Thank you for always believing in me, and for the great advice along the way.

I'm very greatful to the people I met in my first research lab and who made me feel welcome: thank you Elise, Danilo, Luigi and Chunxiao for showing me around. Thank you also for everyone at the LPP for providing a great working environment. I feel like I should have learned more about phonetics by now, but that one is on me. Thank you also to everyone at the Almanach team for welcoming me, I learned a lot while I worked there.

I'm very greatful to Bernard Caron for welcoming me into the NaijaSynCor project, I learned so much because of it and it has renewed my passion for learning and analyzing languages.

Thank you to Mirek, Radek and Xinying for their warm welcome in Ostrava, I have fond memories of our workshop together. Ján, I'm glad we got to meet at the Syntaxfest, and sad that I couldn't find the time to come work with you in Bratislava. Xinying, thank you for adding me to your pokemon collection, I hope we get to travel together again.

To my parents, thank you, I couldn't have done this without you. I'm very greatful that you've always supported me in pursuing my interests.

I've been gifted with two excellent brothers: Cyril, now that I'll have more time

let's go on some more adventures; Edouard, you're not quite right in the head for doing a second thesis, but I can't say I'm surprised, I hope we'll see more of each other now. Love you both lots.

I'm greatful beyond words for the friendship of Ilaine and Iris. Ilaine thank you for adopting me and cheering me on throughout the years. You've both been an immense source of support these past years, thank you for turning this strange, foreign land into a home.

I'm very thankful for the two musketeers Arthur and Gabriele. Thank you for your quiet and unwavering support and for commiserating with me, I'm glad we were brought together by the friendship fairy.

I'm greatful that I had many friends to share the daily lab life with, and tag along through the ups and downs of research: Jade, Gabriele, Angélina, Alexis, Gaël, Kirian, Yixuan, Pedro, Morgan, Mathilde and Yoann. In particular, Jade thank you for always being there to discuss our uncertain futures around sushi, you're one of the only people I trust to give me music recommendations.

Pierre, thank you for the sciency and not-so-sciency discussions, though I'm still not quite sure if I should thank you for inspiring the initial idea behind this thesis. I hope we get to work together one day !

To all of my climbing friends, thank you for the fun and carelessness, and for keeping me (somewhat) sane. I'll see you around, let's climb some more together !

Contents

List of Figures **xiii**

List of Tables **xix**

I Theoretical review **1**

1 Why should we develop treebanks ? **3**

1.1 Introduction 4

1.2 Linguistic research 6

1.2.1 Data-based revision of syntactic theory 6

1.2.2 Contributions to other linguistic disciplines 14

1.3 Teaching foreign languages and syntax 16

1.4 Language technologies 20

1.4.1 Training automatic morpho-syntactic processing systems 21

1.4.2 Creating or enriching linguistic resources 22

1.4.3 Downstream applications 24

1.4.4 Conclusion 26

2 History of treebank development **29**

2.1 The Genesis of Systematic Analyses 32

2.1.1 Pioneers and Their Contributions 33

2.2 Linguistically annotated corpora 35

2.2.1 Before computers 35

2.2.2 Computerized corpora 37
2.3 The first Treebanks 39
2.3.1 Swedish pioneers 39
2.3.2 The first democratized and readily available treebanks 42
2.3.2.1 The Penn Treebank (PTB) 43
2.3.2.2 Combining constituency and dependency 46
2.3.2.3 Prague Dependency Treebank (PDT) 47
2.3.2.4 A paradigm shift toward dependency 48
2.3.2.5 Towards standardisation of dependency annotation . . 49
2.4 Is there still a need for treebanks? 52

3 Treebank Development Practices 55
3.1 Priors of Syntactic Annotation 57
3.1.1 Transcription 58
3.1.2 Tokenization 60
3.1.2.1 Tokenization in UD 62
3.1.2.2 How to represent tokenized text? 62
3.2 Subtasks of Treebank Development 63
3.2.1 Lemmatization 63
3.2.2 Morphological Features 64
3.2.3 Part-of-Speech Tagging in Treebank Development 65
3.3 Resource Scenarios in Linguistic Annotation 67
3.3.1 What can we consider low-resource ? 67
3.3.2 Rich Resource Languages 69
3.3.3 Low-Resource Scenarios 71
3.3.3.1 Exploring Neighboring Languages: 71
3.3.3.2 Utilizing Parallel Corpora: 72
3.3.3.3 Model Transfer 73
3.3.3.4 Annotation Transfer 74
3.3.3.5 Unsupervised Methods 76

3.4 Conclusion 77

II Contributions **79**

4 Annotation guidelines and grammars **81**

4.1 Concerns when developing annotation guidelines 85

4.1.1 Annotation guidelines : content and structure 85

4.1.2 Problems faced by annotators 90

4.1.3 Linguistic criteria and external factors 92

4.1.3.1 Learnability 93

4.1.3.2 Error vs linguistically debatable analysis 95

4.2 Case Study n°1 : Multi-word Expressions 96

4.2.1 Idioms and syntactic irregularity 97

4.2.1.1 MWE and tokenisation in UD 101

4.2.2 Propositions for the encoding of MWEs in UD 104

4.2.3 Conclusion 108

4.3 Case Study n°2 : Naija 109

4.3.1 Linguistic context 110

4.3.2 Treebank development 111

4.3.2.1 Corpus Metadata 111

4.3.2.2 Transcription 112

4.3.2.3 Morphosyntactic analysis 113

4.3.2.4 Polycategoriality and polyfunctionality 115

4.3.2.5 Surface-Syntactic UD annotation 116

4.3.2.6 Evaluation 120

4.3.3 Some idiosyncratic syntactic constructions of Naija 123

4.3.3.1 Clefts in Naija 123

4.3.3.2 Interrogatives 125

4.3.3.3 Serial Verb Constructions 126

4.3.4 Conclusion on the NaijaSyncor treebank 128

4.4 Conclusion and perspectives for the chapter 128

5 Collaborative treebank development **131**

5.1 Introduction 134

5.2 Comparison to related tools 137

5.3 Architecture 140

5.4 New Features provided by Arborator-Grew 142

5.4.1 Class Sourcing 143

5.4.2 Error Mining 145

5.5 Distribution 149

5.6 Conclusion and perspectives 150

6 Global properties of treebanks **153**

6.1 Menzerath-Altmann & Heavy Constituent Shift 157

6.1.1 Related works on Menzerath-Altmann Law 158

6.1.2 Heavy Constituent Shift 158

6.1.3 The Co-effect Hypothesis: Combining Heavy Constituent Shift and Menzerath-Altmann Law 161

6.1.4 Methodology 162

6.1.5 Results 164

6.1.6 Conclusion 165

6.2 Adapting MAL to more syntactically defined units 166

6.2.1 Linear dependency segment 169

6.2.2 Experimental results 171

6.2.3 Conclusion 174

6.3 Syntactically grounded vs artificial trees 175

6.3.1 Looking into the properties of syntactic dependency trees 177

6.3.1.1 Features 177

6.3.1.2 Hypotheses 178

6.3.2 Random tree generation with constraints 179

6.3.3 Results and discussion 183

6.3.3.1 Correlation between properties 183
6.3.3.2 Distribution of configurations 184
6.3.4 Conclusion 187
6.4 Conclusions of the chapter 188

7 Structure Induction from Raw Corpora: Mining Syntactic Fragments 193
7.1 Objectives 195
7.2 Autonomy and Syntactic Units 196
7.2.1 Autonomy Measure 196
7.2.2 Syntactic Units 198
7.3 Data 198
7.4 Methodology 201
7.5 Results and Discussion 203
7.5.1 Size of the Training Corpus 203
7.5.2 Influence of the Annotation Scheme 204
7.5.3 Evolution of Scores Depending on the n Best 205
7.5.4 Comparison with the baseline 207
7.6 Conclusion and Perspectives 208

8 Conclusion and perspectives 211

A Appendix 215
A.1 Grew patterns 215
A.1.1 Filtering query results based on a subpattern 215
A.1.2 Complement of auxiliaries 215
A.1.3 Heavy constituent shift extraction 216
A.1.4 Co-effect of Menzerat-Altmann and Heavy Constituent Shift . . 216
A.2 Tables of experimental results 217
A.2.1 Co-effect of MAL and HCS : number of selected clauses per language 217
A.3 Diagrams illustrating specific tree configurations 222

A.3.1 Trigrams configurations organized by type 222
A.3.2 Disjoint dependencies 222

Bibliography **223**

List of Figures

1.1 Dependency tree for the sentence `A cat walks into a bar` 5

1.2 Constituency analysis for the sentence `A cat walks into a bar` . . . 5

1.3 Dislocated subject in French (SUD annotation scheme) for the sentence 'mais lui il a commencé à marcher bien plus tôt que moi' translated as "But him he started walking much earlier than I did". Sentence_id : ParisStories_2022_10_frèreHyperDifférent___38 12

1.4 Top: screenshot illustrating a query for a VSO pattern using the Grew-Match tool [Guillaume, 2021]. Below: The only two sentences (in red) found in the French Rhapsodie Treebank for VSO with their corresponding dependency trees. Both examples are of interrogative form, known for allowing an SV order swap. S,V and O are highlighted in green. . . 13

1.5 Process flowchart of an example of syntactic similarity research in English in [Wang, 2017]. 18

1.6 Illustration of a parallel analysis between French and English in ParTUT (UD annotation) 20

2.1 Historic Development Towards Treebanks. The illustration delineates projects along two dimensions: (x-axis) the degree to which the underlying text is a corpus, spanning from isolated written examples to representative electronic/digital corpora; and (y-axis) annotation types, evolving from simpler grammatical explanations to complex formal analyses. 32

2.2 Extract of the Hebrew root words associated with the word *light* in the King James' Bible, as shown on the Blue Letter Bible website. 37
2.3 Example of a sentence annotated in the MAMBA format in the Talbanken corpus. 41
2.4 Two sentences tagged in the Penn Treebank format 44
2.5 Penn Treebank annotation for *What did Casey throw ?*, a Wh-question with a trace subject. 46

3.1 Language families with more than 1 million speakers covered by multilingual transformer models from [Hedderich et al., 2021] 68

4.1 Excerpt of the POS Tagging Guidelines of the Penn Treebank explaining how to distinguish between adjective and gerund. 85
4.2 Decision tree for the identification of verbal multi-word expressions (VMWE) in the PARSEME guidelines [Savary et al., 2017]. 87
4.3 Table of dependency relation labels in the UD guidelines 89
4.4 UD guidelines for the `acl` relation in Naija (Nigerian Pidgin). 89
4.5 Analyses with UD *fixed* relations in UD_English-ParTUT and UD_French-GSD 102
4.6 Analyses with UD *fixed* relations in UD_English-PartTUT and UD_French-GSD 102
4.7 Measures of MWEs of English and French UD v2 illustrating important % differences in the usage of *compound*, *fixed* and *flat* across different treebanks. 104
4.8 Left: UD analysis of the adjective `top of the range` - case `a)`. Right: Proposed encoding of Functional MWEs in the CoNLLU format. 106
4.9 UD analysis of sentences 2a and 2b. 107
4.10 Analysis A for *a lot (of)* and *in front (of)* 107
4.11 Analysis B for *a lot (of)* and *in front (of)* 107
4.12 Analyses A and B for *in front of a lot of houses* 108
4.13 Workflow of the NAIJASYNCOR project. 109

4.14 Map of the 11 survey locations. 111

4.15 Analysis for a predicating adjective in 'You will be strong' [PRT_05_Ghetto-life_P_24] 116

4.16 Analysis for an adjective modifying a noun in 'then they did strong magic' [IBA_04_Alaska-Pepe_P_95] 116

4.17 Analysis of *one* as a numeral in 'he then cut one of its ears.' [IBA_04_Alaska-Pepe_P_5] 116

4.18 Analysis of *one* as a determiner in `so I go tell you one story` 'so I will tell you a story.' [IBA_04_Alaska-Pepe_P_5] 117

4.19 Analysis of *one* as a pronoun in `dat one dey too much` 'that one is too much.' [ABJ_INF_08_Impatience_106] 117

4.20 Analysis for the sentence `you know di level now base on who you be` 'You have an idea of the level, based on who you are.' [WAZK_08_Fuel-Price-Increase_MG___23] 118

4.21 Analysis for the sentence `sey na my mama younger sister` 'I said she's my mom's younger sister.' [IBA_01_Fola-Lifestory_MG___61] 118

4.22 UD and SUD analyses for the sentence `dem go seize am` 'They will seize it.' [DEU_C01_D_6] 119

4.23 The three structures of Naija clefts 124

4.24 Analysis of the wey-cleft `na nineteen eighty four wey de born me` 'It's in nineteen eighty-four that I was born.' [KAD_09_Kabir-Gymnasium_P_6]] 125

4.25 Analysis of the copular predication in *na* in `na di ting wey Buhari meet` 'This is the thing that Buhari found.' [IBA_25_Buying-Indomi_159] 125

4.26 Comparative analysis of interrogatives with and without the copula *na* 126

4.27 Serial verb construction annotation in Naija (SUD) for the sentence `di man just carry everyting put underground` 'The man just buried everything.' 127

5.1 A screenshot of the user interface that gives access to the different annotations of a sentence, with one dependency tree per user. The sentence is drawn from the treebank annotation project of spoken Naija (Nigerian Pidgin-Créole), discussed in section 4.3. The sentence can roughly be translated by *I really like maize dumplings.* 136

5.2 Software architecture of Arborator-Grew. 141

5.3 Interface for the relation selection for a SUD project configuration. . . . 142

5.4 Pattern to look for potential errors on verbs without subjects 146

5.5 Search result querying a *comp:aux* relation, with pink highlighting of the governor and the dependent. The sentence from the Old French text *La Chanson de Roland* 'The Song of Roland' (1040 - 1115) can be translated by *The rich duke Gaifier has arrived."* 146

5.6 Grew-match integration component, with example queries on the right and automatic idiosyncratic highlighting of the Grew query language. . 147

5.7 Relation table showing a count of all occurrences of *vocative* relations between a governor (category to the left) and its dependent (category on top) based on their respective parts-of-speech. Here the table query searches in the annotator's own trees; alternatively, the most recent accessible trees can be taken into account. The annotator can directly click to visualize, for example, the two pronouns that have a proper noun as a vocative dependent, in order to verify the two corresponding trees. If a tree contains an error, it can directly be corrected. 148

5.8 Rewriting rule to update the label of nominal subjects. This could be used to transfer from one annotation scheme into another. 149

6.1 Dependency tree of the sentence *I'll give some to my good friend from Akron'.* 162

6.2 The average size c of C constituents (y-axis) is bigger than the average size a of A constituents (x-axis) across the 80 languages of SUD 2.7 where we have at least 20 occurrences of corresponding structures. . . . 165

6.3 Dependency tree of sentence *"This black book on the table costs twenty euros, which is too much for me"* 170
6.4 The MAL in Czech dependency treebanks (SL - sentence length in clauses, f, rf - frequencies and relative frequencies of sentence lengths, MCL – the mean clause length in LDSs). 172
6.5 The MAL modelled by function $y(x) = ax^b$ as the relation between sentence length and the mean clause length 173
6.6 Unordered tree 179
6.7 Random tree generation 181
6.8 Non-linearized trigram configurations distribution for French 185
6.9 Trigram configurations distributions for French. 186

7.1 Dependency tree for the sentence *This tall girl likes climbing.* 198
7.2 Example transformation to merge an amalgamation (à+le → au) 199
7.3 Example annotation for the 4 schemes (from left to right and top to bottom: UD, SUD, SUD+, SUD++) 201
7.4 Influence of training corpus size on the precision of extracted fragments (scheme: SUD, n=1) 204
7.5 Influence of the annotation scheme on the precision of extracted fragments (size: 1 million tokens, n=1) 205
7.6 Precision and recall on extracted fragments from the n best segmentations (SUD schema, 1M tokens) 206
7.7 Precision on random fragments and fragment candidates based on the size of the fragment. Candidate fragments are from the 10 best segmentations (SUD schema, 1M tokens) 207

A.1 Example of a Balanced configuration 222
A.2 Example of a Chain configuration 222
A.3 Example of a Zigzag configuration 222
A.4 Example of a Bouquet configuration 222
A.5 Example of a disjoint set of dependencies 222

List of Tables

2.1 Timeline of some major works on linguistically annotated corpora and treebanks. 35
2.2 Timeline of major advances in treebank development 39

4.1 Classifying MWEs according to two dimensions: degree of compositionality (columns) and degree of syntactic regularity (rows). 100
4.2 Inter-annotator agreement scores and the effect of pre-parsing 121
4.3 Parsing results with and without macro-syntactic annotation - LAS (Labeled Attachment Score) and UAS (Unlabeled Attachment Score). . 122

5.1 Permissions according to user role in Arborator-Grew 143

6.1 Tree-based metrics 177

7.1 Proportion of sequences of length 2 to 4 that are catenas in the different annotation schemes of the annotated corpora. 201

A.1 Values of a, b, c and the numbers of selected clauses in each language. . 221

Part I

Theoretical review

Chapter 1

Why should we develop treebanks ?

Chapter contents

1.1 Introduction 4

1.2 Linguistic research 6

1.2.1 Data-based revision of syntactic theory 6

1.2.2 Contributions to other linguistic disciplines 14

1.3 Teaching foreign languages and syntax 16

1.4 Language technologies 20

1.4.1 Training automatic morpho-syntactic processing systems . . 21

1.4.2 Creating or enriching linguistic resources 22

1.4.3 Downstream applications 24

1.4.4 Conclusion 26

This thesis is concerned with all aspects of treebanks, from the design of their guidelines, to the development of tools to facilitate their development, their development using manual annotation and less supervised methods of structure induction, to the exploitation of the resulting resource to describe linguistic phenomena.

These are the aspects we will cover in the second part of the thesis which is dedicated to our main contributions. The first section will cover some background information necessary for the understanding of the various problems posed by these questions. In the present chapter we will describe what treebanks are and look at their various uses for linguistic research, teaching linguistics and learning languages, and the language technologies that benefit from them.

1.1 Introduction

[Jurafsky and Martin, 2008] define a treebank as a syntactically annotated corpus, where each sentence is associated to a parse tree.

The parse tree often contains part-of-speech tags for each word in the sentence, as well as the syntactic structure of the sentence which can be expressed in the form of relations between words (in a dependency framework, as in Figure 1.1) or groupings of words (phrases in a constituency approach as in Figure 1.2). Other grammatical frameworks can be used to make a treebank such as the Combinatory Categorial Grammar for example [Hockenmaier and Steedman, 2007].

Annotated corpora that only contain information on a word per word basis such as a corpora tagged with part-of-speeches is usually not considered as a treebank [Nivre, 2008a].

Constituency grammar is concerned with explaining the structure of a sentence through describing the various units that make it up, and how the smaller units can be grouped into bigger units until one big unit encompasses the whole sentence. These units are called constituents, and typically given a category based on their composition.

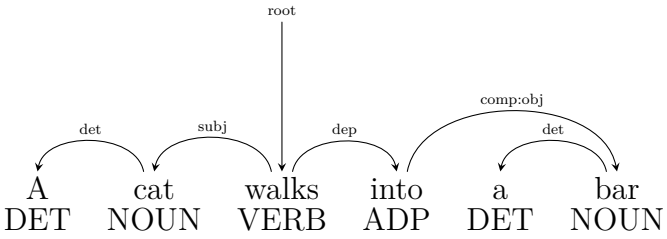


Figure 1.1: Dependency tree for the sentence `A cat walks into a bar`

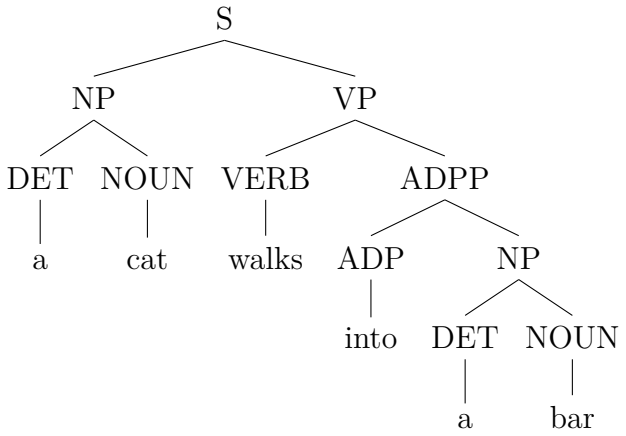


Figure 1.2: Constituency analysis for the sentence `A cat walks into a bar`

In Figure 1.2, the words `a` and `bar` make up a constituent which is called a noun phrase (labelled as NP on the figure).

Dependency grammar, on the other hand, doesn't introduce intermediate units in the form of constituents. Instead in this framework the structure is expressed in the form of directed relations (called dependencies) between the smaller units. These dependencies involve two participants : a governor and a dependant, with the dependant fulfilling a certain function for its governor. For example in Figure 1.1 `a` (the dependant) is the determiner for `bar` (the governor)[1].

The term treebank has also been used to describe corpora annotated with semantic [Bos et al., 2017] or discourse-level information [Carlson et al., 2002], but here we will reserve the term to strictly syntactically annotated corpora[2]. Textual corpora

[1]By convention, we will represent the dependencies as arrows going from the governor to the dependant

[2]Semantics commonly uses more complex structures than simple trees, and we often speak of "graph banks" if the relations are semantic. Yet, some of the standard defining features of treebanks such as open and closed POS classes, valency (eg the distinction between modifiers and complements)

annotated in this manner contain linguistic information that can be useful in various contexts. We will present and discuss the use of treebanks in linguistic research, linguistic and language teaching and language technologies.

1.2 Linguistic research

Treebanks are a subtype of annotated corpora that include syntactic information. The annotation can be leveraged to observe many linguistic phenomena, describe and analyse their nature on the basis of empirical data. While corpora, and particularly digitized corpora can already provide a variety of options to study linguistic phenomena on the basis of surface strings, treebanks extend those possibilities by giving access to syntactic annotations that can be queried to extract examples and provide quantitative analyses of the constructions contained within these corpora.

1.2.1 Data-based revision of syntactic theory

In many ways, treebanks can be used to challenge and revise particular syntactic theories on the basis of data. We see three areas in particular where treebanks can be useful : integrating more realistic linguistic examples that challenge introspective approaches to syntactic theory, introducing frequency as an important descriptor of construction use and covering a broad range of data.

Theoretical syntax In the literature many authors have provided ample criticism of theoretical syntax that is not based on data but rather on *introspection* and *acceptability judgements* by native speakers to figure out which structures are permissible in a language and which are not. The latter type of linguistics is often referred to as *armchair linguistics*, in opposition to *empirical linguistics* which is based on data. At the heart of this opposition there is the idea that there are two tenants to lan-

and the central distinction of Universal Dependencies between lexical words and function words, are sometimes considered to be questions of semantic type that go beyond distributional questions of syntax.

guage : *competence* and *performance* [Chomsky, 1965]. [3] Competence is the innate and unconscious ability of an ideal speaker-listener to understand and utter linguistic productions that respect the rules of their language while performance refers to how they actually use their language in everyday life.

> Linguistic theory is concerned primarily with an ideal speaker-listener, in a completely homogeneous speech-community, who knows its (the speech community's) language perfectly and is unaffected by such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying his knowledge of this language in actual performance. [Chomsky, 1965, p. 3]

There are some advantages to basing syntactic theory on introspective methods rather than corpora. According to [Corbin, 1980, p. 155] :

> [the interest of introspection is]the possibility of considering utterances, other than those attested. Introspection can thus be conceived as the privileged instrument for research on the farthest bounds of the possible that can be predicted from observable data.[4]

Thus the study of competence using introspective methods allows us to also deal with *non-attested constructions* and determine whether they would be permissible despite their absence from data (with the obvious caveat that sometimes native speakers aren't available to provide judgements, as is often the case when studying historical syntax, where the only available materials are grammars if ever they do exist, and corpora).

The problem with delimiting "core" constructions However, the choice of the structures which will be deemed relevant can include significant biases that limit

[3] As described in [Gerdes, 2018, p. 17] and references therein, this opposition can be found in other linguistic traditions outside of generative linguistics, using a different terminology such as *register* and *use*.

[4] Our translation.

the researcher's findings. As pointed out by [Gerdes, 2018], the distinction between the "core" constructions which exhibit regular properties and the "periphery" where exceptions to regularities abound is problematic :

> The limitation to the "core" of Language, to the regularities that are governed by parameter settings, has been identified as central to the development of structural linguistics. It is only at that "core" that exact regularities can be discovered [Hajičová, 2011]. The study of the "periphery", the exceptions, is put off to a later date. The problem is of methodological nature because the distinction between core and periphery (or between competence and performance) is defined on the fly, dynamically while describing a language. This allows identifying regularities nearly *ad libitum* because it allows excluding all contrary data points that one might encounter to the periphery.

[Sampson, 2003] also criticises the lack of comprehensiveness of syntactic analyses provided by theoretical linguists, who are free to leave out a certain number of constructions as they provide their own examples.

> For the theoretical linguists who set much of the tone of computational linguistics up till the 1980s, this kind of comprehensive explicitness was not a priority. Syntactic theorists commonly debated alternative analyses for a limited number of "core" constructions which were seen as having special theoretical importance, trying to establish which analysis of some construction is "psychologically real" for native speakers of the language in question. They saw no reason to take a view on the analysis of the many other constructions which happen not to be topics of theoretical controversy (and, because they invented their examples, they could leave most of those other constructions out of view). [Sampson, 2003, p. 27]

By contrast, in a treebank where the analysis is expected to cover every token of every sentence, there is much less leeway to leave out constructions. Sometimes the

provided analysis will be more arbitrary than syntactically motivated as we will discuss in Chapter 4 when looking more in details at the annotation guidelines design process, but this should be made explicit rather than "left as an exercise for the reader" as if the construction presented no interest or complexity of analysis. Syntacticians who have never attempted to annotate a corpus might not realise the amount of seemingly infrequent constructions that require description. To solidify this point, we can turn to Sampson recollecting his thoughts when he began working on the Lancaster Treebank :

> When I began drawing trees for Geoffrey Leech's project, he produced a 25-page typescript listing a set of grammatical category symbols which he suggested to use, with notes on how to apply them to debatable cases.
>
> I remember thinking that this seemed to cover every possible linguistic eventuality, so that all I needed to do was to apply Leech's guidelines more or less mechanically to a series of examples. I soon learned differently. Every second or third sentence seemed to present some new analytic problem, not covered in the existing body of guidelines. So I and the research team I was working with began making new decisions and cumulating the precedents we set into an ever-growing body of analytic rules. What grew out of a 25-page typescript was published in 1995 as a book of 500 large-format pages, *English for the Computer* (Sampson 1995). [Sampson, 2003, p. 26]

> The idea that a natural language is governed by a limited set of clear-cut grammatical rules does not survive the experience of structurally annotating real-life examples. [Sampson, 2003, p. 29]

Because treebanks are often developed with intended uses other than doing linguistic research on them, for example with the intent of developing natural language processing systems, they have to cover a large range of data in terms of genre and modality. Natural language processing systems are developed for a range of applications, and will sometimes have to be trained on data outside of what is traditionally studied in linguistic theory to be efficient in the domain where they will be applied

to. This data could cover a range of domains such as biomedical texts, user-generated content or law documents for example. The development of treebanks that account for such genres of texts can be expected to yield new insight into the variations of syntactic structures outside of typical grammatical description work.

In addition to expanding what constructions are studied and described, developing and analysing treebanks also forces us to take a good look at our syntactic theories, and revise them where necessary :

> Despite the low esteem in which theoretical linguists held taxonomic work, I soon found that even a small-scale English treebank yielded new scientific findings, sometimes findings that contradicted conventional linguistic wisdom. [Sampson, 2003]

He then provides an example wherein his study of the Lancaster Treebank challenged commonly held beliefs about the type of basic clause structure found in English :

> For instance, introductory textbooks of linguistics very commonly suggest that the two most basic English sentence types are the types "subject – transitive verb – object", and "subject – intransitive-verb". Here, for instance, are the examples quoted by Victoria Fromkin and Robert Rodman in *An Introduction to Language* to illustrate the two first and simplest structures diagrammed in their section on sentence structure (Fromkin & Rodman 1983: 207-9):
> `the child found the puppy.`
> `the lazy child slept.`
>
> Looking at statistics on clause structure in the treebank I developed at Lancaster, though, I found that this is misleading (Sampson 1987a: 90). "Subject - transitive verb – object" is a common sentence type, but sentences of the form "subject – intransitive-verb" are strikingly infrequent in English. If the sentence has no noun phrase object to follow the verb,

> it almost always includes some other constituent, for instance an adverbial element or a clause complement, in post-verb position. *The lazy child slept* may be acceptable in English, but it could be called a "basic" type of English sentence only in some very un-obvious sense of the word "basic". [Sampson, 2003]

According to him, the "subject - intransitive-verb" structure can hardly be called basic, as it is very infrequent to encounter it without any other constituent in the post-verb position. He makes the argument, that basicness has something to do with frequency of use. For Sampson, in order to aptly describe a language and its constructions, one should take into account the frequencies at which these constructions occur in the data.

I can recount one such realisation I had when I started annotating Orfeo [Debaisieux et al., 2016], a corpus of spoken French, during my masters. Prior to that, I had no idea that having a dislocated subject was so frequent in French. Figure 1.3 displays an example sentence with a dislocated subject extracted from the Paris Stories treebank : `mais lui il a commencé à marcher bien plus tôt que moi` ('But him he started walking much earlier than I did'). In fact I had never even heard of dislocations despite having been taught French Grammar in school. This construction was simply not part of the curriculum which is overtly based on how French is written rather than spoken. I can imagine that foreign learners of French also rarely encounter this construction in their studies. But being confronted with dozens of examples that included subject dislocation, it became part of what I consider "basic" constructions of spoken French.

The tools that we use to explore and query treebanks can also impact the kind of insight we might gather from them. Most treebank exploration tools will provide a querying system where the user can look for constructions that match a pattern they have in mind (typically by using a specialised query language with its own vocabulary and syntax). Figure 1.4 illustrates a query for a VSO[5] pattern in a French

[5]VSO stands for *Verb Subject Object*, describing a linear ordering for these three elements inside

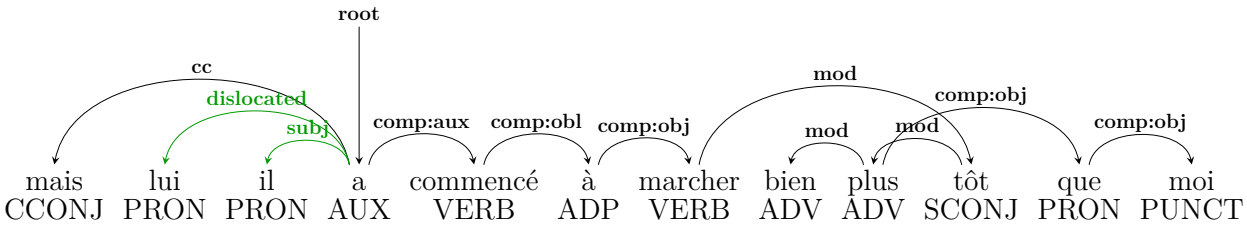


Figure 1.3: Dislocated subject in French (SUD annotation scheme) for the sentence 'mais lui il a commencé à marcher bien plus tôt que moi' translated as "But him he started walking much earlier than I did". Sentence_id : ParisStories_2022_10_frèreHyperDifférent___38

treebank[6], French being known for its SVO word order. Strictly following this SVO rule for French, one might thus expect no results for the VSO pattern. Two sentences are nonetheless retrieved from the Rhapsodie Treebank:

`mais connaissez-vous les sites, euh, sociaux spécialisés` (but do you know, uhm, specialized social sites)?

`certains membres de l'Académie Goncourt auraient dit 'mais pourquoi lui donnerait-on pas le prix Goncourt'` (some members of the Académie Goncourt would have said 'but why don't we give him/her the Prix Goncourt').

Both correspond to an interrogative form, illustrating that the inversion from SV to VS may appear in this case.

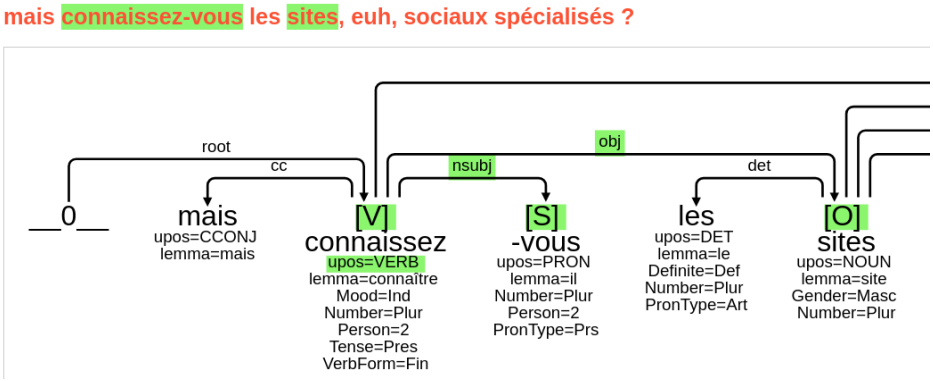
The most crucial part then, is to select the right patterns to see how well the examples match our current understanding of syntax. It will not escape the reader that designing and selecting these patterns of interests is just another way to look for answers to questions we already know. In this aspect, treebank querying is similar to the practice of "armchair linguistics".

It is also possible however, to provide ways of querying treebanks that make less assumptions about what constitutes an interesting query. One option is to design very generic patterns and look exhaustively through their instantiations. The collaborative annotation tool Grew-Match (which we will discuss in more details in Chapter 5) does

a sentence or clause. Other word orders are also referred to using similar abbreviations such as SVO or OVS.

[6]The query can be accessed here : https://universal.grew.fr/?custom=653a7b2dcf877

```
pattern {
V [upos="VERB"];
V -[nsubj]-> S;
V -[obj]-> O;
V << S ;
S << O;
}
```



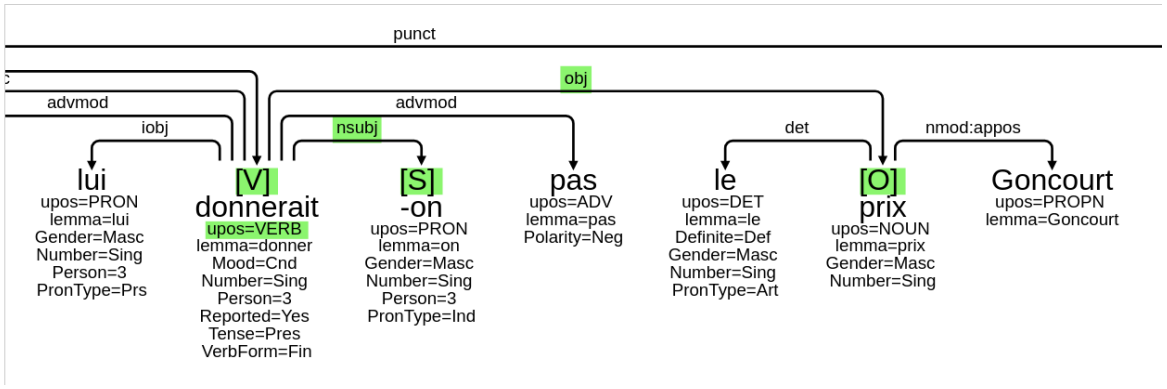


Figure 1.4: Top: screenshot illustrating a query for a VSO pattern using the Grew-Match tool [Guillaume, 2021]. Below: The only two sentences (in red) found in the French Rhapsodie Treebank for VSO with their corresponding dependency trees. Both examples are of interrogative form, known for allowing an SV order swap. S,V and O are highlighted in green.

this when it provides a *relation table* (see Figure 5.7).

The generic pattern used by Grew-Match corresponds to any pair of nodes linked by a dependency relation, and is instantiated by many different patterns that correspond to triplets (dependent-POS, governor-POS, relation-label). This allows for an exhaustive look at the triplets that appear in the treebank and those that do not, and is extremely useful in looking for unusual constructions that could be either errors or unexpected patterns that one might not have thought to look for and integrated in their grammatical description of the language.

1.2.2 Contributions to other linguistic disciplines

While we have so far focused on the implications of using treebanks for specifically syntactic theory, other disciplines in linguistics also benefit from having these resources available. As long as the phenomenon explored is linked to syntactic properties (or as long as the researchers involved want to investigate whether it is) then access to treebanks will facilitate the study. The following list is by no means meant to be exhaustive, but rather to showcase examples of work in various linguistic disciplines that relied on treebanks as a source of syntactic information :

- Typology : e.g. [Choi et al., 2021] looked at word order properties across a variety of languages and checked whether the dominant word-order in terms of usage corresponds to the documented dominant word order in the WALS database [Dryer and Haspelmath, 2013]. They also looked at intra-language consistency using several treebanks for the same language when they were available; [Liu, 2010] measured the distribution of head-initial and head-final dependencies in treebanks for 20 languages and discussed the results with respect to Tesnière's typological classification system [Tesnière, 1959].

- Prosody : A major contibution for spoken French was done in the framework of the ANR Rhapsodie project [Lacheret-Dujour et al., 2019]. The main objective was to define rich, explicit, and reproducible schemes for the annotation of

prosody and syntax in different genres to study the prosody/syntax/discourse interface. Koehn and colleagues [Koehn et al., 2000] made use of syntactic features extracted from a parse tree to predict prosodic breaks. Recent work on different languages including French [Ghaly, 2020] aimed at improving automatic syntactic parsing of spontaneous spoken sentences using prosody. Among the achieved results, the author underlined the potential of prosody to resolve ambiguity and improve parsing. Reversely, a recent study on Mandarin [Hong et al., 2023], relates a data-driven approach to constructing a prosodic grammar making use of large transcribed speech corpora with syntactic-tree parsing and automatic prosodic labeling.

- Semantics : Following the example of the Penn Treebank initiative which significantly contributed to encourage work on statistical parsing, the Abstract Meaning Representation (AMR) project [Banarescu et al., 2013] aims at a semantic representation language in which the meanings of English sentences are written down. The authors hope that this kind of semantic bank resources will spur new work in statistical natural language understanding and generation. Other researchers applied the AMR framework for different languages (Brazilian Portuguese, Latvian, Turkish...).

- Sociolinguistics : An innovative large-scale study [Johannsen et al., 2015] of syntactic variation among demographic groups (age and gender) across several languages makes use of parsed data annotated with universal dependencies. The results reveal several age and gender-specific variations across languages, for example that women use VP conjunctions more frequently than male speakers.

Treebanks and linguistic research The availability of treebanks covering a variety of languages, text genre, domain and modalities has revolutionised corpus linguistics, and many linguistic domains where syntactic information may be important to understand a phenomenon. With regards to linguistic theory treebanks are useful in that they provide a way to challenge and revise it on the basis of data. Taxonomies of

grammatical classes and syntactic functions are often challenged as the data reveals more and more "corner case" examples that fuzzy up the boundaries between them. In the same way, the existence of "core constructions" and what belongs in this group can be challenged, especially as frequency becomes an integral part of grammatical descriptions. This can have significant effects on typological descriptors which are often taken for granted such as the word order of a language, and its degree of freedom. Treebank annotation also shines in making apparent where syntactic description still lacks precision and where decisions are made arbitrarily rather than by relying on tests and criteria.

All of these aspects might suggest that the main contribution of treebanks to syntactic theory consists of muddying up rules by showing that regular patterns are actually not as regular as they seem to be, but treebanks can also provide new ways of looking at the data and extracting regularities that are less perceptible by the human brain.

1.3 Teaching foreign languages and syntax

While treebanks are an invaluable resource for answering research questions with regards to many linguistic disciplines, they can also be used in a pedagogical context to teach usage-based syntax and encourage students to form their own hypotheses based on authentic language materials.

In language learning contexts, it is most typical to use materials specifically designed for students as the primary resource for target language data. These materials take into account the assumed level of the interlocutor and provide appropriate language examples often centred around a scene of everyday life (ordering food at a cafe, introducing oneself..) with specific target vocabulary and grammatical notions in mind. There is an expected progression in the level of difficulty as the learner advances through the materials and learns the lexical, morphological and grammatical information needed. The role of these materials is not to provide authentic language materials, but rather to anticipate learners' needs and give saliency to specific features [Wang,

2017] (for example the differences in meaning between two similar constructions in a specific pragmatic context).

Corpora can be used as an alternative to these materials, and provide complementary information regarding frequency of use, genre-dependent variation and so forth.

> Firstly, using corpora offers an alternative to the reference works (e.g dictionaries or textbooks) that were used until now. The nature of corpora allows to move away from the exemplary approach which has long been used as the basis for developing an understanding of the target language's functioning in a language learning context. This approach places usage contexts and variation at the centre of the understanding of the target language. [Ciekanski, 2014][7]

There is a parallel to be drawn with our discussion of "core" and "peripheral" constructions in 1.2.1, the selection of *pedagogically interesting* constructions or grammatical notions to introduce is a highly contentious subject. There is a choice to be made between simple constructions which present less difficulty for the learner and frequent constructions which the learner will encounter more often. In a learning context that prioritises *autonomy* and where treebanks are presented as a language material, the learner will be able to formulate hypotheses on which constructions belong to which group, and select the ones that seem most relevant to his/her specific needs.

Enabling learners to fill gaps in their understanding When learners are confronted to a syntactic construction that they do not fully master, it can be difficult for them to figure out the next step. They might look up the construction in a textbook or grammar where they will find a definition and a few examples but it might not be sufficient to fully grasp when and how to use this construction. Corpora then, and in particular syntactically annotated corpora which make the structure of sentences explicit, can be used as supplementary materials to access other examples and allow the learner to observe when and where the construction occurs (situational and

[7] Our translation.

linguistic contexts) and gather clues to how they might apply it to their own communicative needs. The problem of accessing those relevant examples that have similar constructions without pre-required *linguistic knowledge* or familiarity with specific *query languages* used for annotation extraction is a complex one. [Wang, 2017] proposes a similarity-based syntactic query system where the user provides input in the form of a natural language example of the construction of interest. The input then goes through a Natural Language Processing pipeline to produce a morpho-syntactic analysis which will be compared to other syntactic analyses available in the corpus. Results are provided in the form of a list where examples are ranked based on their similarity with the original example. This provides an elegant way to ensure that non-specialists can further their understanding of the language.

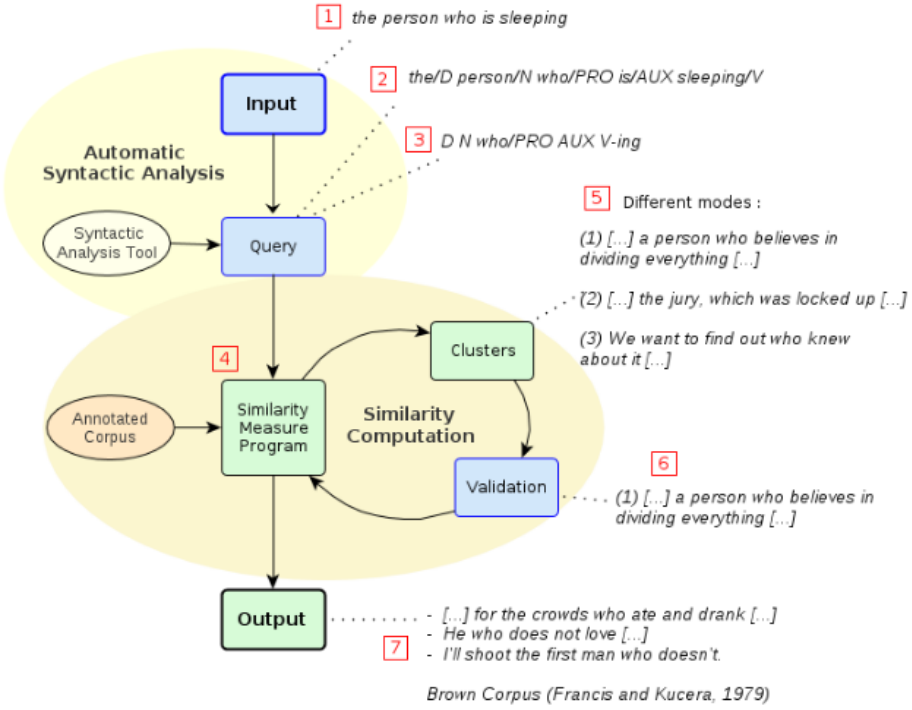


Figure 1.5: Process flowchart of an example of syntactic similarity research in English in [Wang, 2017].

Parallel treebanks Parallel treebanks are corpora in at least two languages that cover the same text in translation, where both texts have been syntactically annotated

(whether manually or automatically)[8]. The treebanks can be provided as is, or with alignments on paragraph, sentence or word level, which means that the corresponding paragraph (or sentence, or word) in the target and source language are matched to facilitate comparison.

Figure 1.6 shows an example from the ParTUT parallel treebanks [Bosco et al., 2012] in French and English, with a dependency analysis using the Universal Dependencies' annotation scheme. Examples like this can illustrate differences in word ordering : e.g the predicate "suffisamment précise" translated as "specific enough" is built following a similar construction, an adjective (ADJ) modified (advmod) by an adverb (ADV), but the word order is reversed, with the adverb coming first in French, while the adjective comes first in English in this specific construction. If the learner is repeatedly exposed to such examples, they might build a generalising rule for themselves (in English an adverb that modifies an adjective will come after the adjective, unlike in French), or if they have already learned such a rule, they might see this as a positive example that reinforces the rule or (as will probably be the case here) find that the example doesn't follow the regularity, and update it. Learners will also have ample opportunity to observe how grammar and lexicon may constrain each other, resulting in highly co-occurring sequences in the form of idioms, light-verb constructions, verb-particle constructions, compounds and complex function words, sometimes described under the umbrella term *multi-word expressions* [Constant et al., 2017]. Coming back to Figure 1.6, we can note that the main verb `s'assurer` in `Assurez-vous` is translated by a verb-particle construction `make sure` which is composed of a verb `make`, and a clausal complement `sure` in the form of an adjective. While `certain` is often an appropriate translation for `sure` (e.g `I'm quite certain` is a paraphrase for `I'm quite sure`), the learner might notice that the use of `certain` with `make` is very rare, and that the combination between `make` and `sure` is preferred.

We have outlined but a few ways in which treebanks, and in particular parallel treebank corpora offer innovative opportunities in the context of language teaching.

[8]Ideally, the annotation is done using a shared annotation schemes such that there is no need for a conversion.

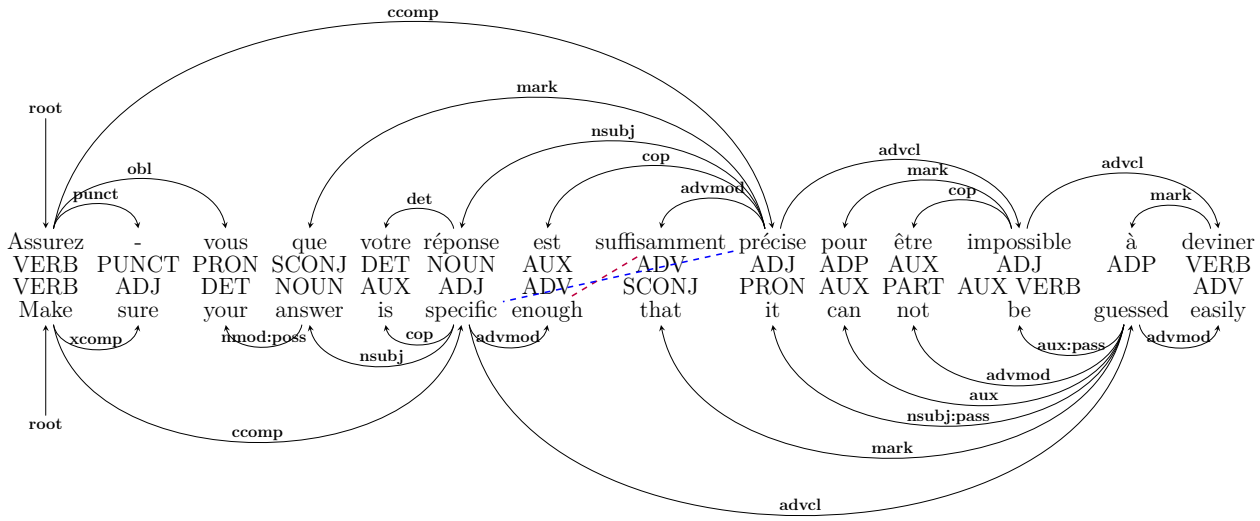


Figure 1.6: Illustration of a parallel analysis between French and English in ParTUT (UD annotation)

1.4 Language technologies

Spoken language understanding has been a major quest of Grail since the beginnings of artificial intelligence combining automatic language processing and computer science to enable human-machine communication. However, it became rapidly obvious that language understanding poses a wide range of challenges, including syntactic analysis and structuring.

From the early stages, morpho-syntactically tagged corpora and treebanks were developed with the idea of developing tools to automatically provide syntactic analyses (see [Francis and Kučera, 1979] describing the design of the Brown corpus and its tagged version). Because corpus annotation is such a lengthy, costly and difficult process, it makes sense that researchers would want to make the most out of the existing treebanks, rather than creating new ones when it can be avoided.

Treebanking costs In the literature, it is widely accepted that treebanking is a costly endeavour, there is not a lot of data that precisely addresses this topic. [Schneider, 2015] distinguishes two types of costs : *upfront costs* which will be the same

regardless of the quantity of data to annotate and covers documenting the annotation scheme, training annotators and building the annotation platform where that is necessary, and *unit cost* which depends on the volume of data to annotate, the number of annotation layers and the granularity of the annotation (introducing more distinctions will make the annotation more difficult and require more training for the annotators and more time to annotate). [Martínez Alonso et al., 2016] provides treebanking costs for 6 projects on building French treebanks undertaken by the Alpage team, and [Seddah et al., 2020] document treebanking costs for their treebank of North African Arabizi. Crucially in the latter work, they report a cost about five times higher than the cost reported for the French treebanks, which they partly attribute to the fact that there were no preexisting guidelines for the language. Thus the costs of annotation will also depend on the prior availability of other treebanks or morpho-syntactic processing tools for the language and domain, with lower resource languages requiring more resources than highly documented ones. It is also worth noting that highly documented languages are in general "written languages" which tend to be highly standardised. This standardisation results in less surface form variation which facilitates automatic processing that have to rely on those surface forms.

1.4.1 Training automatic morpho-syntactic processing systems

With the advent of machine learning these last decades, treebank corpora can be used beyond their original purpose to train automatic morpho-syntactic processing systems. These systems learn to make predictions based on the *statistical regularities* they observe in the training data, and to apply them to new, unseen data. In this way, morpho-syntactic taggers can learn to predict part-of-speech tags and parsers can learn to predict syntactic structures by relying on what they have learned during training. This is particularly useful when the training data and the new data share strong similarities in terms of language variety, text genre and domain, although methods have also been devised to extend this possibility to new data that deviates significantly from the training data [Denis and Sagot, 2009; Şahin and Steedman, 2018].

Evaluating morpho-syntactic processing systems When devising taggers and parsers (or any machine learning system) it is important to know how well we can expect them to perform in the future. For this purpose, there is usually a phase where the tool's performance is evaluated and its errors examined. In the NLP field, evaluation is a very crucial topic around which entire shared tasks [Oepen et al., 2017]; [Fares et al., 2018], journal issues [Paroubek et al., 2007] and workshops are centred. The field mainly distinguishes two types of evaluation : *intrinsic evaluation* where the quality of the system is judged based on its ability to generate predefined annotations (the reference annotation which are considered as a ground truth) and *extrinsic evaluation* or task-oriented evaluation, where the system is judged based on how well it functions "in relation to its setup's purpose" [Galliers and Jones, 1993, p. 22]. Treebanks are vital for the intrinsic evaluation of taggers and parsers, as they provide the reference annotation against which the predicted annotations are evaluated.[9]

1.4.2 Creating or enriching linguistic resources

Treebanks can be used to create or improve various types of linguistic resources including grammars and valency lexicons. While grammars aim at formulating languages' rules that prescribe how to properly combine their building blocks such as sounds and morphemes, words and phrases, valency lexicons [Grishman et al., 1994] more specifically describe obligatory and optional complements of words. A clear benefit of treebanks is to provide a means to confront and connect linguistic knowledge to practical language usage via annotated corpora. Rules can be weighted by their frequency of occurrence. Some rules might be identified as dispensable, whereas observed language productions could reveal incomplete or even missing rules.

[9]Note that while treebanks can be used to evaluate taggers and parsers, taggers and parsers can also be used in turn to provide feedback on annotation consistency. Looking at the type of errors produced by taggers and parsers on a treebank will likely result in finding that some of them are motivated by inconsistencies and errors. In addition, because these systems are sensitive to heterogeneity between training data and test data, they can be used to measure how similar texts are, which lends itself to numerous applications [Guibon et al., 2015].

Grammar Treebanks can be seen as the output of applying a grammar (i.e a collection of rules, which can be probabilistic or not) to a corpus. This grammar exists either in the minds of the linguists (documented by the annotation guidelines), or as a formal or statistical grammar learned by a parser.

Grammar extraction allows us to collect those rules by walking the trees from a treebank, assign them probabilities and possibly remove redundant rules to make the grammar more compact [Krotov et al., 1998]. The extracted grammar may follow a different formalism than that which the treebank was built upon (for example extracting an LTAG grammar from a dependency treebank). We may cite here an excerpt of [Howell, 2020]:

> Grammar extraction uses the syntactic information available in treebanks, collections of syntactic trees, to define grammars. Typically these grammars are produced by walking the trees in a treebank, collecting rules that could produce those structures and pruning to remove redundant rules, taking rule probability into account in order to make the grammar more efficient [Krotov et al., 1998].

The quantification of grammar rules may be of interest to the design of NLP-based technologies as well as in a context of foreign language teaching and learning.

Treebanks are important for dependency analysis and to train high accuracy stochastic analyzers [Candito et al., 2010]. Further attempts of creating linguistic resources have been reported, such as the learning of categorical grammars from a treebank [Alfared, 2012]

Lexicon Treebanks have been used to build or to increase valency lexicons. Valency lexicons become particularly useful in the case of free or flexible word order languages, such as Czech [Kettnerová and Lopatková, 2019]. Beyond monolingual settings, cross-language valency mapping (e.g. between Czech and English, [Uresova et al., 2015]) may contribute to foster linguistic comparison. It will be especially helpful in multilingual natural language processing applications such as machine translation or semantic role

labelling.

1.4.3 Downstream applications

As linguistic resources, treebanks (and by extension grammars and parsers which can be learned from them) provide information that is valuable to many downstream tasks in NLP. Their relative usefulness in relation to these tasks is measured through *extrinsic evaluation* as mentioned in 1.4.1.

Semantic Textual Similarity Task (STS) This classification task involves assigning a degree of similarity to pairs of short texts (often sentences). The reference similarity score is usually obtained by averaging the scores given by several human annotators, from no similarity to semantic equivalence. [Vo and Popescu, 2015] find that adding syntactic trees improves performance in a STS task, in comparison with the baseline which contains no syntactic information.

Semantic Role Labelling This classification task involves assigning semantic role labels (agent, instrument, goal...) to spans in a sentence. These semantic roles are helpful to generalise over different surface realisation of the same argument structures, for example "X gave Y a slice of cake" and "Y was given a slice of cake by X", and to infer answers to queries, such as "Who gave a slice of cake to Y" ?

Systems usually sub-divide the task in two : identifying the correct arguments for a given predicate (*argument identification*) and labelling them (*argument classification*). Typically, the features used for this task will rely on access to a form of syntactic structure as in [Gildea and Jurafsky, 2002] who rely on phrase type, governing category and parse tree path among other features.

Readability Assessment This classification task consists in assigning a predefined readability class to a given document, paragraph or sentence. Readability assessments can be used to improve the accessibility of texts and reduce barriers to information to all members of society. [Venturi et al., 2015] look at Italian health-related documents

(specifically informed consent forms) using READ-IT, a readability-assessment tool for Italian which takes dependency-parsed texts as input. Their features include lexical, morpho-syntactic and syntactic information (e.g percentage of verbal roots with an explicit subject, average clause length..).

Machine comprehension task / Question Answering Question-answering (QA) systems are a type of machine comprehension task designed to provide precise answers to specific questions posed by users, either drawing from a predefined set of data or dynamically getting information from various sources. In the paper proposed by [Liu et al., 2017], the authors explore methods to embed syntactic information into the deep neural models to improve the accuracy of their machine comprehension task.

Sentiment Analysis Sentiment analysis is the computational process of determining and extracting subjective information, such as emotions or opinions, from textual data to identify the sentiment of the content as positive, negative, or neutral. In [Socher et al., 2013] and [Tai et al., 2015], the authors used recursive networks relying on constituency parsing trees. The results suggest that tree structures contribute to an improved modelling of the syntactical property of natural sentences.

Machine Translation Before the advent of neuronal approaches to machine translation (MT), treebanks used to play a pivotal role in enhancing MT systems particularly for language pairs with different word orders. Treebanks provide detailed syntactic information and hierarchical structures of sentences, enabling MT models to grasp the intricate relationships between words, phrases, and clauses in source and target languages. For example, in [Tinsley et al., 2009], the authors describe their work on large parallel Turkish-English treebank to improve the translation quality of phrase-based MT systems.

Coreference resolution Coreference resolution consists in detecting *mentions* inside a document and linking those that refer to the same entity (also called a *coreference chain*). Because mentions are often realized using nouns, pronouns and noun phrases

it seems obvious that using syntactic features would help in detecting them. As a matter of fact [Recasens and Hovy, 2009, p. 52] found that head match, i.e whether the string of the head of a mention matches that of another mention's head, was the most relevant feature commonly used in coreference resolution, which means that accurately finding heads is of crucial importance. This reliance on syntactic features explains why, as noted by [Grobol, 2020], corpora with coreference annotation often have some sort of syntactic analysis available, sometimes often predating the coreference annotation itself.

Looking for syntax in deep representations With the development of *representation learning* methods using artificial neural networks, many NLP tasks such as coreference resolution, or semantic role labelling have moved on from using rich, explicit linguistic features (which often included morpho-syntactic and syntactic information, and thus relied on treebanks or parsers as a source of input), to using only dense embeddings[10] learned from raw data.

The idea that these models could bypass syntax entirely, yet perform better than models using rich linguistic features can be disconcerting for those among us who were convinced that expert linguistic knowledge was important for performing well on those tasks. Instead of completely dismissing syntactic structure as an important source of information for downstream tasks (machine translation, reading comprehension...), a growing body of work has been dedicated to exploring the extent to which these representations encode something that would resemble our understanding of syntax, see [Linzen and Baroni, 2021] for a recent survey on the matter.

1.4.4 Conclusion

Treebanks, as annotated corpora that provide information on the syntactic structure of the sentences they contain, open up a wide range of possible uses. They facilitate corpus-based enquiries on linguistic research topics by providing quantitative data on

[10]These representations take the form of low-dimensional vectors, that is sequences of real numbers, which cannot be easily mapped onto the symbolic representations that were used previously.

authentic examples of language use, abstracting away from the surface string realisations which previously limited the types of possible queries. Developing and analysing treebanks and the syntactic structures they contain encourages us to revise our understanding of linguistic theory, not only in syntax but also in other disciplines such as typology or sociolinguistics. The growing availability of treebanks covering now more languages, text genres, modality and domains opens up perspectives to decenter certain types of texts on which traditional grammars are mostly based. In the classroom, treebanks provide an opportunity for learners to interact with data in their target language, to confront their current knowledge and hypothesis about the language to data, thus enabling them to develop a more active approach to language learning, developing autonomy and critical thinking skills. Manually crafted treebanks can be used as reference data to train parsers, and to evaluate their ability to accurately learn to reproduce their annotation. For this purpose, having access to many treebanks covering many languages, genres, modalities and domains is crucial to train robust parsers, capable of handling heterogeneous data. Treebanks and parsers have also been shown to be beneficial for downstream tasks in natural language processing, including semantic textual similarity, semantic role labelling, readability assessment, question answering or coreference resolution. The evaluation of various syntactic representations and parsing strategies with regards to their contribution to downstream tasks is still an ongoing effort. However, with recent developments in representation learning methods many systems have moved on from using symbolic linguistic features, instead relying on the representations learned by neural networks to solve these complex tasks. Given their surprisingly goods results on a number of tasks, much effort has been dedicated to understanding what type of information is captured in the learned representations.

Chapter 2

Retracing the history of treebank development

Chapter contents

2.1 The Genesis of Systematic Analyses 32

2.1.1 Pioneers and Their Contributions 33

2.2 Linguistically annotated corpora 35

2.2.1 Before computers 35

2.2.2 Computerized corpora 37

2.3 The first Treebanks 39

2.3.1 Swedish pioneers 39

2.3.2 The first democratized and readily available treebanks . . . 42

2.3.2.1 The Penn Treebank (PTB) 43

2.3.2.2 Combining constituency and dependency 46

2.3.2.3 Prague Dependency Treebank (PDT) 47

2.3.2.4 A paradigm shift toward dependency 48

2.3.2.5 Towards standardisation of dependency annotation 49

2.4 Is there still a need for treebanks? 52

This chapter looks into the history of treebank development, showcasing its historical context and the advances made in the domain of linguistically annotated corpora. We follow the chronological advances and influences that have shaped contemporary treebanks.

- **Systematic syntactic analysis:** Initial efforts were invested in the comprehensive syntactic analysis of sentences, with a shift from purely textual descriptions towards systematic and formalised descriptions.
- **Syntactically annotated corpora:** What started as a laborious effort of manually annotating corpora, designed to facilitate and enrich the interpretation of specific culturally-relevant texts, has become much more widespread with the emergence of corpus linguistics. The digitization of corpora, and the mergence of natural language processing have allowed large corpora to be more easily enriched with coherent annotations in the form of part-of-speech tags, lemmas and glosses, and broadened the linguistic representativity and depth of annotated texts.
- **Emergence of treebanks and their standardisation:** This evolution led to the birth of the first treebanks, corpora characterised by their detailed syntactic annotations, applied uniformly across sentences. New methods and tools have been designed to generalize these syntactic descriptions to other domains and languages, which has brought forward interesting challenges and made the standardization of these resources a necessity.
- **Is there still a need for treebanks:** We reflect on the significance and applicability of syntactically annotated corpora within the contemporary fields of computational linguistics and natural language processing. This leads us to assess the relevance of symbolic representations in an age where continuous representations have become predominant. We will discuss potential future implications.

Figure 2.1 provides a comprehensive overview of the diverse initiatives and precursors in the domain of treebank development, plotted within a two-dimensional

space. The x-axis illustrates the degree to which the analysis was done on text that corresponds to today's criteria of a corpus, transitioning from singular written sentences to linguistically representative electronic corpora. On the other hand, the y-axis shows the growing complexity of annotations, beginning from basic transcriptions and lemmatizations, towards a formalised analysis that incorporates Part-Of-Speech (POS) tags, linguistic features, and dependency relations. The works depicted in this figure will be presented in the following sections, and the numerals positioned within the diagram's quadrants correspond to the three following sections, each describing a specific phase in the history that led to contemporary treebank development methodologies.

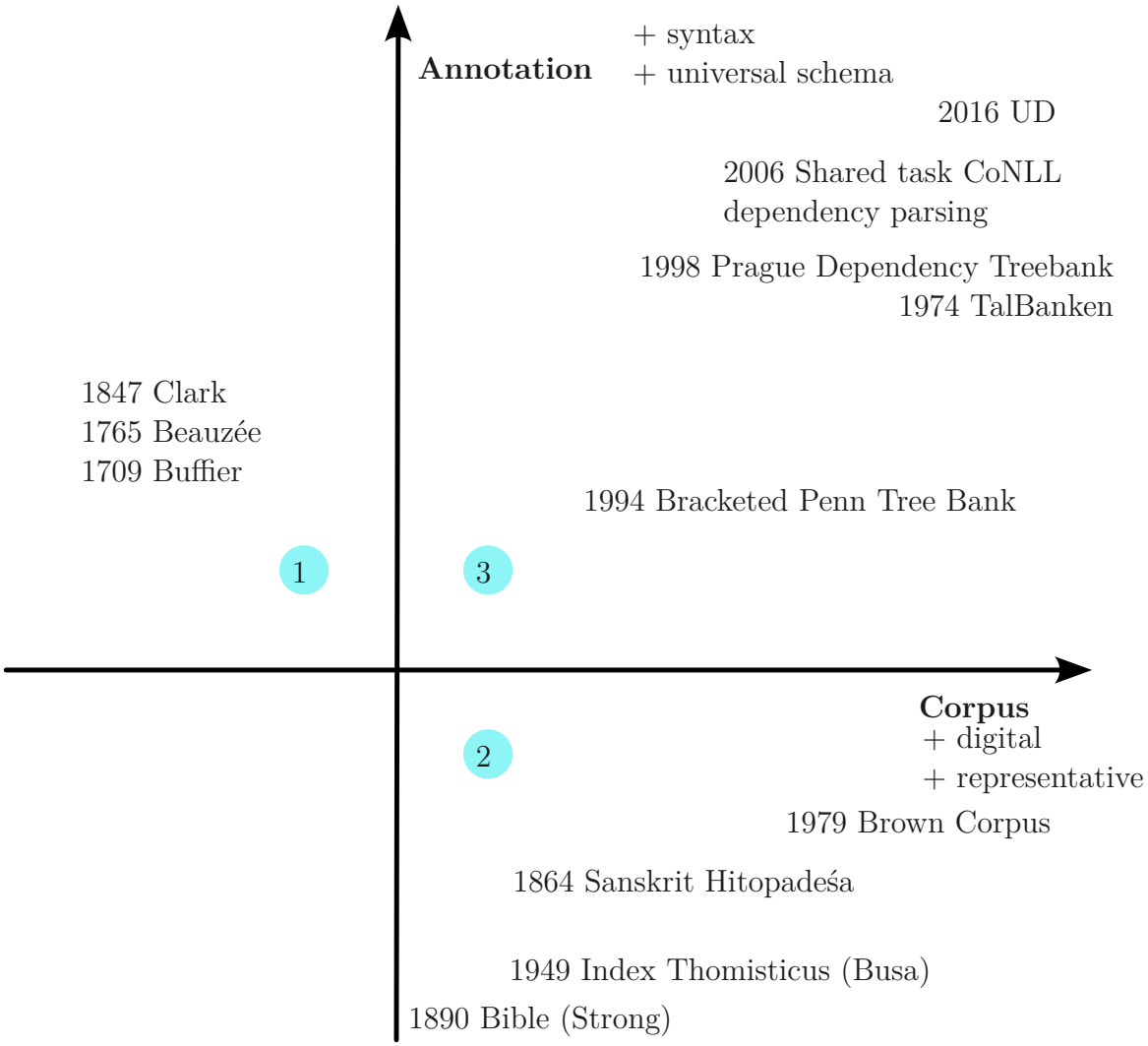


Figure 2.1: Historic Development Towards Treebanks. The illustration delineates projects along two dimensions: (x-axis) the degree to which the underlying text is a corpus, spanning from isolated written examples to representative electronic/digital corpora; and (y-axis) annotation types, evolving from simpler grammatical explanations to complex formal analyses.

2.1 The Genesis of Systematic Analyses

In this section we present the first systematic syntactic analyses, offering glimpses into the evolution of linguistic inquiry long before the digital era. By *systematic*, we mean looking at whole sentences and figuring out the relationships between all the words.

Historically, syntactic analyses were conducted even in the absence of comprehensive corpora. This encompasses both cherry-picked authentic sentences and contrived

examples formulated by syntacticians. Kahane and Mazziotta gathered these early analyses, seeing them as an older form of today's treebanks without the digital aspect [Kahane and Mazziotta, 2022] :

> Traditional resources, which we consider to be authentic non-digital treebanks, have developed following the emergence of systematic syntactic analysis approaches for attested examples in the 18th century, starting with Buffier [1709], and then in an even more formalised manner with the encyclopedists, Dumarsais [1754], and Beauzée [1765].[Kahane and Mazziotta, 2022] (our translation)

This historical introspection raises important discussions regarding the validity of constructed examples versus authentic, real-world ones, and debates between textual and diagrammatic descriptions. Remarkably, despite the common definition of a corpus emphasizing representativeness, authenticity, and systematic material sampling (excluding machine-readability), Kahane and Mazziotti designate as non-digital treebanks even textual data that does not conform to these principles. This is particularly notable given that the French term for treebank, 'corpus arboré', translates to 'treed corpus'[1].

2.1.1 Pioneers and Their Contributions

Various scholars have provided crucial stepping stones toward the systematic analysis of syntactic structures, despite their studies lacking the comprehensive, authentic, and systematically sampled nature typical of a corpus.

- [Buffier, 1709] first comprehensive sentence analysis.
- [Dumarsais, 1754] introduces a formal, systematic approach to whole sentence analyses.
- [Beauzée, 1765] coinage of the term 'complement'.

[1] 'treed' in the sense of *made into trees*

- [Gaultier, 1817] introduces the initial tabular sentence analysis, a forerunner of the CoNLL format.
- [Billroth, 1832] German scholar presenting the first hierarchical diagram of Latin syntactic structure.
- [Barnard, 1836a] A predecessor of the constituent structure in a grammar for deaf people.
- [Clark, 1847] introduces tree-like diagrams denoting syntactic structures with bubbles around words.
- [Reed and Kellogg, 1876, 1877] explore tree structures varying writing directions per part of speech.
- [Jespersen, 1937] works on multilingual systematic analyses.
- [Tesnière, 1959] pioneers a complete dependency analysis called *stemma*, i.e., trees enhanced with coreference and POS *translation* (a phenomenon by which certain words change category through their combination with other words).

While these works displayed complex syntactic analyses, most of them did not formalise them into the diagrams we have come to be familiar with. The practice became more common when it started to be used in pedagogical settings :

> It is noticeable that all the syntactic analyses of the 18th century were done in words, without any diagrams. It was not until two centuries later, with Tesnière [1934], that syntactic diagrams made their appearance in the French school, even if diagrams had appeared one century earlier, with Billroth and Billroth [1832] and Barnard [1836b]. [Kahane, 2020]

Analyzing these works gives rise to a spectrum of queries:

- Were treebanks indeed constructed by these authors, albeit inadvertently?
- What constitutes the minimum size to qualify as a treebank?

- Did these pioneers formulate their examples or utilise existing texts?
- Were written or spoken examples prioritised?

Examining the pedagogically-motivated syntactic analyses of works from the likes of Buffier provides valuable insights into the initial stages of treebank development and the historical intertwining of linguistic research and language teaching.

2.2 Linguistically annotated corpora

1864	Müller and colleagues produce an interlinear gloss of the Indian Sanskrit Hitopadeśa (fables) [Müller, 1864]
1890	Strong creates a lemmatized concordance of the Bible [Strong, 1890]
1949	Busa starts working on the Index Thomisticus [Busa, 1980]
1964	1st release of the Brown corpus [Corpus, 1964]
1969	Publication of the American Heritage Dictionary based on the Brown corpus [Fiske, 1969]
1979	Tagged version of the Brown corpus [Corpus, 1979]
1992	Release of the tagged Penn Treebank [Marcus et al., 1993]

Table 2.1: Timeline of some major works on linguistically annotated corpora and treebanks.

2.2.1 Before computers

There is a long history of people annotating corpora to provide complementary information to the readers, and to facilitate and expand the interpretation of texts. Some of these early examples of annotation were linguistic in nature, providing "root words" (or what we would call lemma in modern terminology) or grammatical analysis in the form of interlinear glossing, to make accessible the texts which were sometimes written in ancient languages that couldn't be read easily by contemporary readers, or simply in a language not mastered by the target audience. These early linguistically annotated corpora were created before the invention of computers and modern methods of data storage and processing, making their creation a time-consuming and painstaking endeavour.

Among such examples, Müller, a German professor at Oxford, published in 1864 an interlinear gloss of the Indian Sanskrit Hitopadeśa, a collection of politically and morally instructive fables.

A few decades later, Strong, an American theologist employed more than 100 people to develop a lemmatized concordance of the Bible [Strong, 1890] which aimed at facilitating theological work.

In this concordance, the author annotates the King James version of the Bible (in English) with the index of a lemma corresponding to its translation in the original texts (Hebrew for the Old Testament, Greek for the New Testament). The concordance also lists the relevant verses for each word, including a small extract of the text surrounding it, allowing the reader to look for the meaning of the associated word in the original language.

Strong's lemmatized concordance, initially a monumental printed work, laid the foundation for the subsequent electronic databases that allowed for faster and more versatile searches. With the advent of digital technology and the increasing desire for easily accessible biblical resources, Strong's concordance was digitized and integrated into several online platforms. The Blue Letter Bible website is one such platform that has harnessed the richness of Strong's data, presenting it in an electronic format and coupling it with an advanced query engine.

For example, one could look up the concordance to see how the word "light" is used in the King James version and find out that several lemmas were associated with the original Hebrew texts.

This query is presented in Fig. 2.2 using the query engine present on the Blue Letter Bible website [2].

Another work that is particularly significant in how it revolutionized the methods used for corpora annotation is Roberto Busa's Index Thomisticus, which he started workin on in 1946. It is a lemmatized concordance of the works of Thomas Aquinas, as well as a number of related authors. Busa had high ambitions and when he learned about IBM's research on punch cards, he managed to convince them to support his

[2]https://www.blueletterbible.org/search/search.cfm?Criteria=light&t=KJV&lexcSt=0#s=s_lexiconc

Strongs #	Hebrew	Transliterated	English Equivalent
Old Testament (Hebrew) for "light"			
H215	אוֹר	*'ôr*	light, shine, enlighten, break of day, fire, give, glorious, kindle
H216	אוֹר	*'ôr*	light(s), day, bright, clear, flood, herbs, lightning, morning, sun
H217	אוּר	*'ûr*	fire(s), light
H219	אוֹרָה	*'ôrâ*	herbs, light
H3313	יָפַע	*yāpaʿ*	shine, shine forth, shew thyself, light
H3381	יָרַד	*yāraḏ*	(come, go, etc) down, descend, variant, fell, let, abundantly, down by, indeed, put off, light off, out, sank, subdued, take
H3974	מָאוֹר	*mā'ôr*	light, bright
H4237	מְחֻזָּה	*meḥĕzâ*	light
H5051	נֹגַהּ	*nōḡah*	brightness, shining, bright, light

Figure 2.2: Extract of the Hebrew root words associated with the word *light* in the King James' Bible, as shown on the Blue Letter Bible website.

work. Busa's collaboration with IBM led to the creation of a computerized database. This endeavour required a tremendous amount of human and computer work [3], and remains an impressive achievement.

Later, the contents of the *Index Thomisticus* were made available in a digital, online format as technology advanced and the internet became a more prevalent tool for academic research.

2.2.2 Computerized corpora

The evolution of corpora from manually compiled to electronically stored has been instrumental in shaping the field of corpus linguistics. Initial corpora, such as those assembled in the early days, primarily served to compute word frequencies. As technology advanced during the 1960s, corpora became scientifically sampled and digitized for computational analysis. This innovation made data extraction not only faster and more efficient but also paved the way for replicable studies and verifiable findings.

One of these first corpora is the Brown corpus which was first made available in 1964 [Francis and Kučera, 1979]. This corpus of American English was quite large for the time, about 1 million words, and the texts were sampled so as to obtain a balanced corpus with various genres represented. It was later used as a citation base for the

[3] It is estimated that 1 millions hours of human work, and 10 000 hours of computer work were necessary.

creation of the American Heritage Dictionary [Fiske, 1969].

Soon enough linguists sought to extract more information than was available in the surface representation, to go beyond frequencies and word co-occurrences. To enrich the initial content of the corpora, annotations were devised by linguists in the late 1960's, in particular on the morphological and morpho-syntactic properties of the words making up the corpora. These annotations were provided with documentation which often came in the form of a technical report explaining the various labels and the decisions made by the annotators. This made many enquiries possible: querying for complex patterns that abstract from the surface text such as subject inversion, a particular sub-categorization frame, coordination of unlikes etc.

Here again, the Brown corpus was one of the precursors, with the first tagged version released in 1979. The tagging took place over several years and was done in a partially-automated process. The team used a TAGGIT [Greene and Rubin, 1971], a rule-based tagger developped by by Greene and Rubin. This tagger used a list of manually devised rules that stated which morpho-syntactic categories could co-occur. This tagger had an error rate of approximately 30%, which means that still many errors had to be post-edited, and the rules improved upon.

The excerpt below shows what the tagged format looks like for the sentence : *The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place.*

```
The/AT Fulton/NP-TL County/NN-TL Grand/JJ-TL Jury/NN-TL
said/VBD Friday/NR an/AT investigation/NN of/IN Atlanta's/NP
recent/JJ primary/NN election/NN produced/VBD no/AT evidence/NN
that/CS any/DTI irregularities/NNS took/VBD place/NN ./.
```

Each word is followed by a slash and its corresponding part-of-speech tag (for example NN for noun, NNS for plural noun, or AT for article)[4].

From the beginning, the researchers had another goal in mind: making automatic

[4]The actual Brown Corpus uses a more detailed and nuanced tag set than this simple example might suggest, totalling 82 different tags.

1974	Talbanken treebank [Teleman, 1974] [Einarsson, 1976]
1978	Ellegård treebank of English [Ellegård, 1978]
1989	1st International Workshop on Parsing Technologies [Tomita, 1989]
1994	Release of the bracketed PTB [Marcus et al., 1994]
1994	Beginning of the Czech National Corpus [ek Čermák, 1997]
1997	Beginning of the French Treebank [Abeillé et al., 1999]
1998	Prague Dependency Treebank [Hajič, 1998]
1999	1st CoNLL shared task [Tjong Kim Sang]
2011	Universal POS tagset [Petrov et al., 2012]
2016	Universal Dependencies v1 [Nivre et al., 2016]

Table 2.2: Timeline of major advances in treebank development

(or semi-automatic) parsing easier. This influenced considerably the design of their annotation scheme, as explained in the technical documentation of the corpus :

> Since the purpose of the tagged corpus is to facilitate automatic or semi-automatic syntactic analysis, the rationale of the tagging system is basically syntactic, though some morphological distinctions with little or no syntactic significance have also been recognised. On the whole, the taxonomy is traditional and should be transparent to the grammarian, but in some areas, distinctions have been made that may not be immediately obvious. [Francis and Kučera, 1979]

Soon after the Brown corpus, other projects began to develop their own linguistically annotated corpora, going beyond simple morphosyntactic tags and instead annotating full syntactic structures. We will present some of these projects in the next section.

2.3 The first Treebanks

2.3.1 Swedish pioneers

Among the very first treebanks, we find the Talbanken treebank [Einarsson, 1976] developed by Ulf Teleman and his colleagues from Lund University in the early 1970's. *Talbanken* in Swedish translates to 'Speech Bank' or 'Speech Repository' in English.

The material of the corpus comes from several genres and represents both spoken and written Swedish, as described by Nilsson and colleagues in the following excerpt :

> It is divided into four parts, which together comprise close to 320,000 words. The four parts are: *Professionell prosa* (Professional prose, about 85,000 words), *Gymnasist-svenska* (Swedish by high school students, about 85,000 words), *Samtal och debatt* (Conversation and debate, about 75,000 words), and *Boråsintervju-erna* (The Borås interviews, about 75,000 words). The first two are written language and the last two are spoken language, all syntactically annotated. [Nilsson et al., 2005]

It is quite impressive that this early treebank also included spoken Swedish, at a time when most of linguistic research focused solely on written texts. This disparity between written and spoken treebanks is still very much present today.

Unsurprisingly given the time period the treebank was developed in, the corpus was manually annotated. The data was then stored on punch cards, a popular digital data storage method at the time, which provided a limited space to encode the desired information.

> From a modern point of view, the MAMBA-format may seem a little obscure and odd. But it should be kept in mind that at the time of its creation, Talbanken was originally placed on punch cards having a limited upper line length. This does of course convey limitations on the encoding format. Talbanken was later transformed into a more convenient electronic form, but the limitations remained. [Nilsson et al., 2005]

The annotation format, called MAMBA consists of two different layers: one composed of morphological features and part-of-speech tags, the other encoding a partial phrase structure (the constituents are present but there are no phrase labels) with grammatical functions somewhat in the style of dependencies.

To give some ideas of this tabular format, you can see in Figure 2.3 the tokens provided in column 6, which also hosts pseudo-tokens, 4-digit numbers, that encode sentence and some phrase boundaries. The following 7th column hosts the part-of-speech

```
P11126050001     0000                    <<       GM          046
P11126050002     *MAN                    POZPHH   SS          046
P11126050003     FÄSTER                  VVPS     FV          046
P11126050004     STÖRRE                  AJKP     OOAT        046
P11126050005     VIKT                    NN       OO          046
P11126050006     VID                     PR       OAPR        046
P11126050007     ELEVERNAS               NNDDHHGGOADT         046
P11126050008     SPONTANA                AJ       OAAT        046
P11126050009     FÖRMÅGA                 NN       OA          046
P11126050010     1000                    IF       OAET        046
P1112605001110002ATT                     IM       IM          046
P1112605001210002UTTRYCKA                VVIV     IV          046
P1112605001310002SIG                     POXPHHGGOO           046
P1112605001410002MUNTLIGT                AJ       AA          046
P1112605001510002OCH                     ++OC     ++          046
P1112605001610002SKRIFTLIGT              AJ       AA          046
P11126050017     .                       IP       IP          046
-------------------------------------------------------------------
1    2 3   4   5    6                    7        8           9
```

Figure 2.3: Example of a sentence annotated in the MAMBA format in the Talbanken corpus.

annotation. The 8th column is the most obscure as it provides partial bracketing annotated not with Phrase labels but with functional labels. As an example consider the 4th and 5th line of the 8th columns, which both start with 'OO' indicating that these two tokens "större vikt" 'greater weight' form an object (whose head is not indicated). For a more detailed description of the annotation format see [Nilsson et al., 2005].

Because of its format among other things, the resource was challenging to use in modern settings until [Nilsson et al., 2005] reprocessed it, providing a new dependency conversion of this treebank.

Another interesting project starting in the 1970s was the C-ORAL-ROM corpus (Corpus of Oral Romance, [Cresti and Moneglia, 2005]). It contained transcribed spontaneous speech and POS annotation. Discourse markers received special marks, and prosodic groups were annotated in a way that inspired later Spoken Dependency Treebanks [Lacheret-Dujour et al., 2019]. The *annotation en grille* linked coordination and disfluency with a common annotation format. This kind of representation seeks to capture the specificities of spoken language by using a layered grid to mark different

syntactic and discursive levels. Yet, C-ORAL-ROM did not encode anything that can easily be related to (phrase or dependency) tree structure in its original annotation scheme.

2.3.2 The first democratized and readily available treebanks

In the 1980s and 1990s, treebank development gained momentum. During this time, key treebanks such as the Penn Treebank [Marcus et al., 1993], Prague Dependency Treebank [Hajič, 1998], and the French Treebank [Abeillé et al., 2000] were developed. They were more largely distributed and served as the basis for many treebanks to come. Alongside these treebanks, the field of parsing also emerged, and a larger community rallied around workshops and shared tasks where systematic evaluation of parser output on these key datasets became the norm.

Broadly speaking, treebanks were developed to provide structured, syntactically-annotated linguistic data to support both linguistic research and the burgeoning field of Natural Language Processing (NLP). Their development typically relied on a combination of automatic processing and manual correction, aiming to achieve a balance between scale and accuracy.

Although context-free parsers and their algorithms were discussed since the 1960s [Earley, 1968], it was also clear at this time that context-free parsers were too restrictive to characterise all sentences of natural languages. Given the weight of generative syntax at the time, and its disregard towards natural language data (and thus corpora), there was little incentive to build parsers that could actually analyse common sentences, even in English.

While the actual syntactic analysis therefore remained non-computational, the POS tagging had become a common part of electronic corpus annotation, done rather in the NLP departments than in the linguistics departments of research.

At the AT&T Bell Labs, the stochastic PARTS system [Church, 1988] was developed. PARTS provided a context-dependant POS tagging and could also provide a partial syntactic analysis by bracketing noun phrases. Instead of being fully built by

hand, the POS annotations was built from the output of system such as PARTS and iterative manual corrections were first put into practice.

This tool provided a starting point for the development of the first English treebank, the Penn Treebank [Marcus et al., 1993].

2.3.2.1 The Penn Treebank (PTB)

The Penn Treebank stands as a hallmark in the early history of treebank development. Covering more than 4.5 million words, the creation of this treebank of American English, was a significant endeavour, as described by Marcus and colleagues :

> In this paper, we review our experience with constructing one such large annotated corpus—the Penn Treebank, a corpus consisting of over 4.5 million words of American English. During the first three-year phase of the Penn Treebank Project (1989-1992), this corpus has been annotated for part-of-speech (POS) information. In addition, over half of it has been allocated for skeletal syntactic structure[5]. [Marcus et al., 1993]

The initial annotation was carried out using the POS parser PARTS [Church, 1988] mentioned above. The system was trained on the Brown corpus modified to use the Penn Treebank tag set. The tool learnt to assign POS tags so as to optimise the product of the lexical probabilities (the probability of a tag, knowing the word) and the product of the contextual probabilities (the probability of observing a tag X given the two following tags Y and Z). The system had an 3-5% error rate and the automatic mapping to the new tagset caused another 4% errors, which gave a combined estimated error rate of 7-9% on the Penn Treebank.

Manual post-editing played a pivotal role in refining these annotations. In order to find systematic patterns of errors, confusion matrices were built.

While the Penn Treebank's tagset found its basis on the tagset that had been devised for the Brown Corpus, the authors recognised that the latter contained redun-

[5]The *skeletal syntactic structure* that the authors refer to consists in bracketing, i.e delineating the boundaries oh phrases. While lacking expressivity, simple bracketing has the advantage that it can be added to actual text with a simple text-editor while remaining readable.

Battle-tested/NNP industrial/JJ managers/NNS here/RB always/RB buck/VB up/IN nervous/JJ newcomers/NNS with/IN the/DT tale/NN of/IN the/DT first/JJ of/IN their/PP$ countrymen/NNS to/TO visit/VB Mexico/NNP ./. a/DT boatload/NN of/IN samurai/NNS warriors/NNS blown/VBN ashore/RB 375/CD years/NNS ago/RB ./.
"/" From/IN the/DT beginning/NN ,/, it/PRP took/VBD a/DT man/NN with/IN extraordinary/JJ qualities/NNS to/TO succeed/VB in/IN Mexico/NNP ,/, "/" says/VBZ Kimihide/NNP Takimura/NNP ,/, president/NN of/IN Mitsui/NNS group/NN 's/POS Kensetsu/NNP Engineering/NNP Inc./NNP unit/NN ./.

Figure 2.4: Two sentences tagged in the Penn Treebank format

dancies that would affect their parser, given it's stochastic nature, as explained in the following excerpt from the technical documentation :

> However, the stochastic orientation of the Penn Treebank and the resulting concern with sparse data led us to modify the Brown Corpus tagset by paring it down considerably. A key strategy in reducing the tagset was to eliminate redundancy by taking into account both lexical and syntactic information. Thus, whereas many POS tags in the Brown Corpus tagset are unique to a particular lexical item, the Penn Treebank tagset strives to eliminate such instances of lexical redundancy.

Therefore, they decided to limit the number of tags and merge those that they deemed redundant. For example, verbs are usually tagged with one of five tags depending on their tense (VB for a form without an overt tense, optionally followed by a suffix (D for past tense, G for gerund, N for past participle and Z for third person singular present). This paradigm also applies to verbs such as `have` and `do`, but they have their own specific tags in the Brown corpus' taggset. The Penn Treebank doesn't follow these rules and instead uses the five main verb tags for all verbs.

The authors also had a different philosophy when it came to what the tags meant. While in the Brown corpus the syntactic function a world played didn't seem to influence which tag it would get, this consideration was taken into account in the Penn Treebank. This comes as no surprise if we consider the fact that Marcus and colleagues' end goal was to build a bracketed version of the corpus. The annotation then would require to know the function of each phrase's head so as to name them appropriately, thus requiring a more functional view of morphosyntactic tags. The authors provide

an example where both corpora chose different analyses :

> For instance, in the phrase `the one`, one is always tagged as CD (cardinal number), whereas in the corresponding plural phrase `the ones`, `ones` is always tagged as NNS (plural common noun), despite the parallel function of `one` and *ones* as heads of their noun phrase. By contrast, since one of the main roles of the tagged version of the Penn Treebank corpus is to serve as the basis for a bracketed version of the corpus, we encode a word's syntactic function in its POS tag whenever possible. Thus, `one` is tagged as NN (singular common noun) rather than as CD (cardinal number) when it is the head of a noun phrase.

The second version of the bracketing applied on the Penn Treebank, and used starting from 1993, included nodes denoting empty traces following a Chomskyan constituency analysis, but simplifying it to be applicable to actual textual data. These empty nodes were used for example for implied subjects of infinitives and gerunds, see for example 2.5 which illustrates the analysis for the sentence *What did Casey throw ?* which includes an empty noun phrase (NP) in this analysis, inside the verbal phrase (VP) that surrounds *throw*.

This provided a major challenge for the development of automatic parsers, because these traces had to be reconstructed before the syntactic structure could be built. [6]

While the Penn Treebank focused on annotating constituent structures through bracketing, other treebanks of the era adopted different annotation philosophies which will be outlined in the following sections.

[6]Note that this challenge also gave rise to the Tree-Adjoining Grammar (TAG) formalism, developed at the University of Pennsylvania [Joshi, 1985]. The basic TAG trees can contain the desired empty nodes, and thus can be used to parse real texts without the prior reconstruction of traces. Tag can thus provide parse trees (the so-called "derived trees") that were at least partially satisfactory for the syntacticians of the time.

```
(SBARQ (WHNP-2 What)
       (SQ did
           (NP-SBJ Casey)
           (VP throw
               (NP *T*-2)))
       ?)
```

Figure 2.5: Penn Treebank annotation for *What did Casey throw ?*, a Wh-question with a trace subject.

2.3.2.2 Combining constituency and dependency

The bracketing approach to language annotation proved to be impractical for languages with freer word order than English[7].

Dependency and constituency syntax have evolved as primary syntactic theories, each addressing unique linguistic phenomena and computational applications. Dependency syntax provides a simpler explanatory model for syntactic relations in languages with a freer word order, such as Russian. It avoids the complexity of movement and empty nodes found in constituency syntax which we described above. When computational linguistics began embracing multilinguality, the need to optimise syntactic parsing technologies for typologically diverse languages became crucial.

The notion of syntactic relations (or function) is crucial for free word order languages and cannot be implicitly encoded by a phrase's position in a sentence, such as the subject being the NP in front of the VP. This was acknowledged in the construction of the Negra corpus [Skut et al., 1997], that relied on an annotation scheme tailored for free word order languages (which they called "non-configurational languages"). Their annotation scheme makes explicit references to Tesnière but tries to adapt the standards set by the Penn Treebank, resulting in a hybrid format of constituent structure with edges annotated with syntactic functions. The Negra project, just like the sub-

[7]Interestingly enough the only other large Penn Treebank was developed for Chinese, another fixed-word-order language [Xia et al., 2000]

sequent larger Tiger treebank [Brants et al., 2002], was based on whole newspaper articles, following the Penn Treebank example.

2.3.2.3 Prague Dependency Treebank (PDT)

At around the same time, researchers in Prague started to work on a treebank of Czech [Hajič, 1998]. In this paper which introduces the Prague dependency annotation scheme, Hajič does not even mention the Penn Treebank and refers instead to Henry Kučera, the Czech-American pioneer of computational linguistics [Kučera, 1961] and tagged corpora (He was the second author of the seminal paper of the Brown Corpus [Francis and Kučera, 1979]). As can be expected given the the long tradition of syntactic dependency analysis in Czech [Šmilauer, 1947; Panevová, 1974; Novak and Sgall, 1968] Hajič places his work in the framework of dependency analysis.

The Prague Dependency Treebank annotation scheme has three levels:

- Morphological tagging,
- Dependency analysis (called the *analytical level*),
- *Tectogrammatical analysis* (this layer is supposed to encode meaning but is better understood as deep syntax) [Melčuk, 1988; Kahane and Gerdes, 2022].

In 2004, a connection was formed with the Penn Treebank through the development of the Prague Czech-English Dependency Treebank [Hajič et al., 2012]. This marked the starting point of the first parallel dependency treebank, offering syntactic dependency annotations for the English texts underlying the Penn Treebank (Wall Street Journal articles) and their corresponding translations into Czech. This treebank contains syntactic information for a corpus of approximately 1.2 million words.

It is worth noting that executing a Negra, Tiger or Prague style annotations would be immensely challenging using a basic text editor, as was employed in the development of the syntactic bracketing for the Penn Treebank. Functional annotation requires specific annotation tools equipped with a graphical user interface. Notably, both the

German and Czech treebank annotation groups pioneered the creation of a graphical annotation tool before inaugurating their respective projects.

The emergence of functional annotation in the late 1990s was certainly conditioned by the evolution of computers and their interfaces. The required technological advancements surrounding graphical interfaces was most likely a contributing factor as to why the transition to dependency annotation did not occur sooner. Still, the question arises: why did the shift happen at all? We will look into answering this query in the next section.

2.3.2.4 A paradigm shift toward dependency

The Negra/Tiger and the Prague treebanks were precursors to a paradigm shift in NLP: In a few years, the NLP community shifted to dependency, and in the mid 2000 they had taken the majority of parsing efforts. This shift from constituency-based parsing and treebanks to dependency-based approaches was influenced by several factors, both theoretical and practical :

As previously noted, for both the Czech and German languages, characterized by their free word order and deemed "non-configurational" by Generative Grammar, the shift towards dependency was primarily due to linguistic considerations.

Yet, even for languages such as English, dependency structures can be developed that are often simpler and more intuitive than constituency structures, which revolve around phrasal units.

> Dependency grammar is an attractive framework when working in a multilingual perspective since [...] dependency trees are not sensitive to the order of the words in a sentence, which allows us to deal with languages with free word order as easily as languages with strict word order. This is in contrast with phrase structure where the tree structure incorporates word order [de Lhoneux, 2019]

Dependency is also closer to semantic structure and a rising interest in downstream applications that need semantic relations such as information extraction might also

have played a role in the development of dependency.

The rise of dependency parsing, notably more computationally efficient than constituency parsing, is intertwined with advances in parsing algorithms and accuracy standards. With algorithms such as the Eisner algorithm for projective dependency parsing in 1996 [Eisner, 1996] (which possesses polynomial time complexities), dependency parsing successfully integrated with stochastic approaches. The advances made in dependency parsing created a virtuous circle that has greatly benefited treebank development.

2.3.2.5 Towards standardisation of dependency annotation

In the early 2010s, there was a growing effort in standardising dependency annotation, exemplified by initiatives such as the Universal Tagset [Petrov et al., 2012], Stanford Dependencies, and Google Tagset, along with the organization of shared tasks by the CoNLL conference.

An important approach of homogenising various tagsets, was done by the Universal Tagset [Petrov et al., 2012] :

> To facilitate future research in unsupervised induction of syntactic structure and to standardise best-practices, we propose a tagset that consists of twelve universal part-of-speech categories. In addition to the tagset, we develop a mapping from 25 different treebank tagsets to this universal set.

Since the Conference on Natural Language Learning (CoNLL) shared tasks in 2006 and 2007, the development of parsers has been envisioned within a multilingual context. In this context, the need for a collection of treebanks of typologically varied languages and homogeneous, not only in terms of format but also in terms of annotation scheme, became more pronounced. This data would prove crucial for two purposes: providing reliable evaluation across languages, thus ensuring that the methods could generalise, and providing homogeneous data which facilitates cross-lingual transfer methods.

One project in particular addressed this growing need. The Universal Dependencies

(UD) project [McDonald et al., 2013; Nivre et al., 2016] represents a collaborative effort to annotate multilingual corpora using a standardised dependency framework. This ensures consistent syntactic structures across various languages. The project which started under a somewhat modest setup (a collection of 6 treebanks[8] using homogeneous syntactic dependency annotation) has managed to rally hundreds of people who have built a truly impressive collection of 245 treebanks, covering 141 languages for v2.12 released in May 2023.

The choice to use a dependency framework in UD was based on computational efficiency (which was a concern for parsing). The success of UD can also be attributed to its approach which centres lexical words (also called content words), and offered an accessible entry point to parts of the NLP community not versed in formal syntax. This approach also has the benefit of making it possible to encode structures in a similar way cross-linguistically, by making the functional elements leaves, see this quote from the original UD creators, before content words were made to be governors of function words :

> Specifically, with more typologically and morphologically diverse languages being added to the collection, it may be advisable to consistently enforce the principle that content words take function words as dependents, which is currently violated in the analysis of adpositional and copula constructions. This will ensure a consistent analysis of functional elements that in some languages are not realized as free words or are not obligatory, such as adpositions which are often absent due to case inflections in languages like Finnish. It will also allow the inclusion of language-specific functional or morphological markers (case markers, topic markers, classifiers, etc.) at the leaves of the tree, where they can easily be ignored in applications that require a uniform cross-lingual representation. [McDonald et al., 2013]

Another interesting choice that made the transition smoother and facilitated onboarding new researchers was the relative ease with which annotations from previous

[8]The original language included were English, French, German, Korean, Spanish and Swedish.

versions of the treebank could be preserved. This meant that while adhering to the homogeneous annotation scheme, treebanks could store additional information (POS in the XTAG column, additional node features in the MISC columns, or tree-specific metadata for example). This was especially important for treebanks that had finer-grained language-specific analysis, which could have been diluted in the universal annotation scheme. Thus, treebanks could more easily be converted to and from UD into another annotation schemes, and several versions of the treebank peacefully coexist.

From a theoretical point of view, UD did not take a firm stance in favour or against existing syntactic theories trying to combine the best of all worlds with NLP considerations.

> Whereas theory-neutral annotation caters to a larger group of users, it runs the risk of not being informative enough or containing too many compromises to be useful for special applications. On the other hand, theory-specific treebanks are clearly more useful for people working within the selected theoretical framework but naturally have a more restricted user group. Recently, there have been attempts at combining the best of both worlds and maximise overall utility in the research community through the use of rich annotation schemes with well-defined conversions to more specific schemes (Nivre [2003]; [Sasaki et al., 2003]). In addition to minimising the effort required to produce a set of theory-specific treebanks based on the same language data, such a scheme has the advantage of allowing systematic comparisons between different frameworks. [Nivre, 2008b]

Moreover, UD was amenable to additions and extensions, such as Enhanced UD (incorporating deep syntax through graphs) [Schuster and Manning, 2016], Parseme (enabling multi-word annotation), coreference [Nedoluzhko et al., 2022] annotation, and SUD (a sibling annotation scheme, offering surface syntactic annotation) [Gerdes et al., 2018].

Looking forward, the trajectory of UD and dependency annotation ventures into potentially innovative terrains, such as mSUD with annotation at the morpheme level and the CostAction Unidive, broadening the UD community to corpus linguistics researchers exploring various facets like Multi-Word Expressions (MWE), morphology, and comparative linguistics.

Certainly, the UD project continues to influence new projects in treebank development and parsing, and stands as a key benchmark in the field.

2.4 Is there still a need for treebanks?

In the NLP field, the emergence and quick evolution of various language technologies has introduced a new paradigm where explicit syntactic structures are no longer an essential feature in many systems. These systems are transitioning from symbolic representations (in the form of syntactic structures) to continuous representations (also called embeddings) learned through language models, which allow a more direct semantic analysis while performing surprisingly well on a variety of tasks.

Thus, numerous state-of-the-art systems have adopted methodologies that either : 1) abstain from using explicit/gold representations of syntactic structures or 2) construct their own internal structures derived from downstream supervision, both distancing themselves from a reliance on treebanks.

However, treebanks continue to be relevant in linguistics and computational linguistics research for a variety of reasons. Many languages, particularly those that are less commercially and globally prevalent, remain in want of comprehensive descriptions or larger-sized treebanks to facilitate research and technological development. Furthermore, numerous genres and varieties of languages, including user-generated content and spoken language, have yet to be thoroughly explored and documented in treebank formats.

Treebanks also remain invaluable for evaluation purposes. They provide a structured and consistent medium through which the efficacy and accuracy of various language technologies, such as parsers and generators, can be meticulously assessed and

compared. For linguists, treebanks serve as indispensable resources, offering a rich, structured repository of linguistic data that is pivotal as a basis to develop and test various hypotheses.

In this context, a philosophical question surfaces regarding the intrinsic utility of linguistics, and, on a broader canvas, of science, with the advancement of technological tools. With machine learning permeating various scientific domains, the traditional pathway that associates understanding a phenomenon with improving systems is becoming progressively less obvious. The significance of treebanks, and by extension, of simpler, more interpretable models—remains undeniable, not necessarily for their direct contribution to applied systems, but as an invaluable tool to elucidate observed phenomena. Even as technological advancements provide us with alternative methods that don't require the use of explicit syntactic structures, treebanks endure as precious resources in both linguistic and computational research. They are kept relevant by acting as an explicative mechanism that simplifies the understanding of linguistic phenomena, thereby warranting their ongoing development and refinement.

Chapter 3

A Typology of Treebank Development Practices

Chapter contents

3.1 Priors of Syntactic Annotation 57

3.1.1 Transcription 58

3.1.2 Tokenization 60

3.1.2.1 Tokenization in UD 62

3.1.2.2 How to represent tokenized text? 62

3.2 Subtasks of Treebank Development 63

3.2.1 Lemmatization 63

3.2.2 Morphological Features 64

3.2.3 Part-of-Speech Tagging in Treebank Development 65

3.3 Resource Scenarios in Linguistic Annotation 67

3.3.1 What can we consider low-resource ? 67

3.3.2 Rich Resource Languages 69

3.3.3 Low-Resource Scenarios 71

3.3.3.1 Exploring Neighboring Languages: 71

3.3.3.2 Utilizing Parallel Corpora: 72
3.3.3.3 Model Transfer 73
3.3.3.4 Annotation Transfer 74
3.3.3.5 Unsupervised Methods 76
3.4 Conclusion 77

In computational linguistics, treebanks serve as key resources. These are essentially annotated datasets where sentences are linked with their syntactic structures, generally represented as trees. Over time, the methods used to develop treebanks have evolved to respond to difficult challenges.

This chapter's purpose lies in mapping out these treebank development methodologies. By providing some key elements for understanding their theoretical bases, and identifying the challenges that are addressed, we aim to give the reader a clearer picture of what these methods entail. Notably, the challenges encountered often depend on the presence (or more crucially the lack) of resources specific to a language or domain.

Here is a summary of the questions that guide our exploration :

- What are the resources used to build the treebank?
- Is there a foundational understanding of the language, perhaps formalized via typological features from a database?
- Are there any insights to be gained from linguistically similar languages?
- Do the methods rely on pre-existing tools such as taggers, or resources in the form of parallel corpora in other languages, or word embeddings (continuous representations of words in the form of vectors encoding semantic similarity)?

With these questions in the back of our mind, the following sections will describe various approaches to treebank development in very different scenarios. We'll uncover the strengths of each of them, and shed light on potential pitfalls.

3.1 Priors of Syntactic Annotation

The first challenges encountered when developping a treebank usually start during corpus collection and selection. Depending on the integrity of the raw data (which might encompass duplicate entries and texts of inconsistent quality), and given the

time and human resources at the project's disposal, it could be pivotal to judiciously sample the most illustrative texts, based on the project's objectives.

- **Audio Sources**: When starting with audio files, transcription usually comes as the primary step. Transcribing spoken language into written form, especially in linguistically diverse environments, presents an interesting set of challenges presented in subsection 3.1.1.

- **Written Texts**: Ideally, written documents come in a simple format where the raw text can be easily extracted. Initiating with more complex documents in formats such as HTML, PDF, or DOCX necessitates their conversion into raw text, which is not always a straightforward process. However this step is fundamental to ensure a clean base and allow for a smooth annotation process.

Once these preliminary steps have been taken, the focus shifts to *transcription* and *tokenization*. Both of these tasks, while seemingly straightforward, actually have significant impact on downstream tasks and imply making difficult choices. The availability (or the lack) of resources for the specific language in question might encourage researchers to follow an accepted standard to ensure compatibility with said resources. Writing technical documentation in the form of a transcription and tokenization guide is imperative, as it establishes a consistent foundation, thereby ensuring the development of a coherent treebank.

3.1.1 Transcription

For oral corpora, audio files undergo transcription into a character-based format. Two primary transcription methods can be used : phonetic, which mirrors the sound of spoken language, and orthographic, adhering to standard writing conventions. For treebanks we commonly use a transcription system that is the easiest to use for annotators and users, i.e for languages that have a standardized orthographic system, we simply use the common script of the language. For languages where there is not stable orthographic standard, or if the phonetic component of the data constitutes a

relevant aspect for future research questions, a phonetic-based transcription might be used instead, such as in the UD Treebank of Beja [Kahane et al., 2022]. However, phonetic-based transcription is not as common and orthographic transcriptions is the most common choice.

Yet, even the transcription into a standardized and easy-to-use writing system requires to make certain choices. For instance, determining whether slight phonemic variations warrant unique transcriptions can influence the subsequent analyses. A simple example is whether to standardize the French utterances `ouais` and `oui` (respectively translated as `yeah` and `yes`) into a single representation such as `oui`[1]. The choice between a transcription leaning towards phonetics or towards standardised orthographic transcription is significant. While the former is easier to align with the original data (sound or user-generated content, if it is available), the latter, being more accessible, might be preferred by non-specialists, and more easily searchable, interpretable, and analysable with standard NLP tools as it has fewer types (different tokens). Employing a pronunciation dictionary to annotate the phonetic form or the original writing style for user-generated content can bridge these two approaches.

Orthographic transcription brings its own set of challenges, especially when determining a standard, as this can have political undertones. Consider Nigerian Pidgin, also known as Naija, a language spoken in Nigeria, which we return to in Chapter 4. It primarily uses English as its lexifier language, meaning that a significant portion of its vocabulary is derived from English. However, phonetic differences are clear between pronunciations in English and Nigerian Pidgin. For instance, the English word `thing` is pronounced as `tin` in Naija. The influence of British colonizers led to the initial use of Nigerian Pidgin as a lingua franca throughout Nigeria, before it evolved into a distinct language. Choosing a spelling variant such as `thing` or `tin` becomes a political statement reflecting one's perspective on the current position of Naija. During the transcription process for the Naija treebank [Caron et al., 2019], transcribers were not given specific guidelines on these choices to ensure the collection represented

[1] Another case could be `j'sais pas` and `je sais pas` which could be translated as `I dunno` and `I don't know`.

genuine orthographic usage. While this ensures a usage-based transcriptions, it poses significant challenges to employ NLP methods. As a result, a standard English-centric feature was incorporated in addition to the more phonetic transcription in the initial stages of treebank development, to facilitate the use of English NLP tools, at least until sufficient data had been annotated.

Following stages to transcription encompass data cleaning and homogenization. Decisions here, such as addressing spelling errors or normalizing punctuations, acronyms, slang, and emoticons, in particular for user-generated content, critically shape the ensuing annotation phase.

3.1.2 Tokenization

Tokenisation: An Integral Step The act of tokenization is intricately linked to transcription and revolves around delimitating the smallest units for syntactic annotation. Within the scope of treebank development, tokens often mirror what we typically recognize as words, though in certain contexts, they might lean more towards morphemes, as observed in the Japanese UD treebanks (see Omura et al. [2021] for a discussion on the word delimitation issues in the Japanese treebanks).

Tokenization often employs whitespace and punctuation as indicative of word-boundaries. This method, however, might not always be operational. For example, languages such as Japanese, Chinese, and to some extent Thai and German do not use explicit delimiters between words, which increases the complexity of tokenization. Automatic tokenization is a hard task for these languages but even in languages that are more generous with whitespace, out-of-vocabulary technical terms and multi-word expressions might not be segmented appropriately by simple approaches.

Modern European languages gravitate towards a natural space-based tokenization, yet exceptions around the following aspects persist :

- **Punctuation**: The role of punctuation in tokenisation is multifaceted. Questions arise such as whether punctuation should be treated as independent entities or attached to words. This varies across punctuations like commas, periods, sus-

pension points, apostrophes, and hyphens. For instance, in English, apostrophes are often right-attached, as in `can't`, where `'t` could be considered a separate token. On the other hand, French usually prefers left-attachment, as seen in `l'est`, tokenized as `l'` and `est`.

- **German Compound Words**: German offers a unique challenge with its compound words. Due to the absence of space characters in compounds, lexicons become essential for accurate tokenisation. For instance, consider the compound word `Lebensversicherungsgesellschaft` (life insurance company). Its tokenization would ideally split it into its constituent words `Leben(s)` 'life', `Versicherung(s)` 'insurance', `Gesellschaft`'company'), but this is not straightforward without a lexicon[2].

- **White Space Anomalies**: Even in languages that do employ white spaces, there are instances where syntactically relevant tokens extend beyond these spaces. A case in point is the French phrase `parce que`, where the two separate words function together as a conjunction meaning `because`.

Beyond European languages, tokenization is an even larger issue:

- Scriptio Continua - languages without whitespace: Some languages, such as Ancient Greek, Latin, and modern languages like Chinese and Japanese, use scripts that do not incorporate white spaces. In these languages, tokenisation evolves into a full-fledged research topic. Parsers require distinct tokens, yet tokenisers need the underlying syntactic structures to finalize token boundaries. This cyclical dependency mirrors the classic chicken-and-egg problem. A promising answer to this challenge has been proposed : allowing the parser to handle both tasks concurrently by leveraging specific word-internal relations, as discussed by [Li et al., 2019].

[2]There is a unique challenge visible in this example: `Lebens` is the genitive form of `Leben` but `Versicherungs` is not an independent form of `Versicherung`. The 's' only appears in compound nouns. So it is unclear whether it should be tokenized separately as an obligatory syntactic element. The German Universal Dependency treebanks separate amalgams (`im` → `in` + `dem`) but do not separate compound nouns.

Tokenisation then, while seemingly straightforward, is still full of linguistic intricacies and presents challenges that require a thorough understanding of the language and its structures.

3.1.2.1 Tokenization in UD

The Universal Dependencies (UD) approach adopts a practical perspective on tokenization, leaning heavily on the whitespaces present within the text as delimiters. There are two main exceptions to this general approach :

- **Space Inside a Token**: For Vietnamese, it is considered acceptable in UD to use white space inside tokens, as this is necessary for the readability of words, but the language is considered as an exception. In other languages, certain grammatical elements, despite having spaces within them, can be seen as singular unit, at least semantically. For instance, phrases like `on top of` are often treated as single entities. To address such scenarios in UD, the tokenization keeps the white-space-induced units, but the relation *fixed* is employed to connect the tokens. We will discuss this issue related to the analysis of multi-word expression more in depth in section 4.2.

- **Amalgam Challenges**: There are instances where a singular word may require segmentation to facilitate a semantically and syntactically accurate annotation. Such cases, where spaces might be lacking but division is necessary, pose a unique challenge to tokenizers. A common example is the the Spanish `dámelo` which can be de-amalgated into three tokens : `da` 'give', `me` 'me', *lo* 'it'.

3.1.2.2 How to represent tokenized text?

Post tokenization, the output can often be represented in a simple text file where spaces delineate the desired token boundaries. For instance:

```
va -t-on maintenant
he 's Peter 's friend and wo n't bite you.
```

However, for intricate cases, such as amalgams or multi-word tokens, a more structured format becomes necessary. A tabulated format, akin to the one adopted by dependency treebanks in UD, the *CoNLL-format*[3], can be employed for tokenization, wherein the only populated columns correspond to the token number and the token itself.

Subsequent processes post-tokenization encompass lemmatizing, tagging (both for part-of-speeches and morphological features), and, lastly, integrating information about syntactic relations. As tokenization sets the stage, its accuracy and consistency remain critical in ensuring the effectiveness of the later steps in linguistic analysis.

3.2 Subtasks of Treebank Development

Treebanking is not an isolated task but is deeply interconnected with various linguistic resources that provide the fundamental knowledge and structure required for high-quality annotations. The languages worldwide showcase a wide-ranging spectrum: from languages for which there exist comprehensive resources to those where the most basic resources are absent. This section provides some thought about leveraging these available resources in the task of treebank development.

3.2.1 Lemmatization

Lemmatization can be succinctly defined as the process of reducing a word to its base or root form. For instance, `running` is lemmatized as `run`, and `better` might be lemmatized as `good`. The relevance of lemmatization is amplified for languages with rich morphological variations, where a single root word might manifest in numerous forms based on tense, case, mood, etc.

An intriguing interplay exists between lemmatization, tokenization, and subsequent syntactic analysis. Modern parsing systems often amalgamate tasks, handling part-of-speech (POS) tagging and lemmatization concurrently. Consider the word `tired`. If

[3]https://universaldependencies.org/format.html

lemmatized as `tired`, it is predominantly an adjective, indicating a state of exhaustion. On the other hand, if lemmatized as `tire`, it is inferred as a verb, suggesting the act of becoming exhausted. Such nuances underline the imperative need for precision in lemmatization, based on a joint lemmatization and annotation guide, see chapter 4, as it directly influences and shapes syntactic interpretations.

3.2.2 Morphological Features

Morphological features are an integral part of the annotation and provide fine-grained insights beyond the POS tags into the structure and function of words within a sentence. These features encompass the different grammatical categories that a word can take on. For instance, these features can indicate tense, mood, aspect, case, number, or gender.

The Universal Dependencies Approach: The Universal Dependencies (UD) initiative aims at creating cross-linguistically consistent treebank annotation including the morphological features. The UD features are divided into various categories, with seven being lexical features. Inflectional features are further subdivided, with seven focusing on nominal inflections like case and number, and then addressing verbal inflections such as tense and mood. This structured approach allows for a standardized representation across different languages. The documentation can be accessed at Universal Dependencies Feature Index.

Promoting Comparative Linguistic Research: One of the foundational ideas behind UD is to prioritise the use of standardised morphological features across different languages. This consistent approach promotes comparative linguistic research. By employing universally recognised features rather than idiosyncratic, language-specific ones, researchers can draw parallels and contrast more effectively across languages. This has the potential to uncover shared linguistic phenomena, and to shed light on universal principles governing language structure. In practice, this also comes with a risk of transferring linguistic analysis of an existing linguistic tradition to a new

language.

3.2.3 Part-of-Speech Tagging in Treebank Development

Part-of-speech (POS) tags are used to indicate the grammatical categories of words in a given sentence (noun, verb, adjective...). It serves as the foundation for many other syntactic tasks, and its accuracy significantly impacts downstream applications.

One fundamental aspect to note is that a word's grammatical category remains somewhat consistent across different usages; its function however can widely vary based on context. The Universal Dependencies (UD) guidelines succinctly encapsulate this principle:

> A word's category should be primarily determined by prototypical (expected) syntactic behavior, as typically recorded in a dictionary, rather than by the context of a particular sentence. Syntax still plays an important role, especially in cross-linguistic mapping of same-named categories. However, prototypical (expected) syntactic behavior is of more importance than function performed in exceptional contexts.

POS tags, especially when they are defined using shared tagset with universal aspirations, are tremendously useful in low-resource scenarios. They provide a shared representation between languages, and there have been successful attempts at transferring parsing models from a source language to a target, related language by relying on POS tags to derive dependencies [Zeman and Resnik, 2008].

Although POS tagging stands as one of the pioneering tasks in text annotation, it presents challenges, particularly in disambiguating POS in some contexts. The choices made during POS tagging can significantly influence the subsequent syntactic analysis as they are one of the key features used by parsers [4]. The UD guidelines touch upon this intricacy:

> Perhaps the most difficult part are ambiguous function words that do not inflect (i.e. morphology does not help us), yet they perform two or more

[4] Many parsing errors can be traced back to an incorrect part-of-speech tagging.

> significantly different syntactic functions, which we normally associate with different parts of speech. [...] For example, distinguishing PRON from SCONJ (`that` (en), `que` (es), что/čto (ru)) is important because pronouns, unlike conjunctions, may become core arguments and fill valency slots of verbs.

In summary, POS tagging remains an indispensable step in treebank development, serving as the cornerstone for accurate syntactic analysis.

Gloss: Bridging Linguistic Understanding and Annotation A gloss is a succinct annotation or explanation that accompanies a word or phrase, elucidating its meaning or grammatical properties. In the realm of linguistic research and treebanking, glosses serve as pivotal tools linking treebanks with field linguists, whose central tool are interlinear glossed texts.

Typically, a gloss couples the lemma (the canonical form of a word) with a succinct representation of its features. This compact notation facilitates quick comprehension, particularly for field linguists and researchers working on language documentation.

Leipzig Glossing Rules: There are many glossing standards, the most common being the Leipzig glossing rules—a set of conventions for consistently glossing linguistic data. These rules provide a standardized approach to interlinear morpheme-by-morpheme glosses, ensuring uniformity and clarity across linguistic research.

Gloss in Treebanking Formats: Generally glosses aren't integrated as is into standard treebanking formats, but rather they are translated into morphological features. In the Universal Dependencies (UD) framework, for instance, a straightforward gloss often equates to a translation of a lexical word and can be found in the MISC column. It's worth noting, however, that keeping complete glosses leads to redundancy—overlapping with the lemma and morphological features columns.

3.3 Resource Scenarios in Linguistic Annotation: The Standard Pipelines

3.3.1 What can we consider low-resource ?

In their survey [Hedderich et al., 2021], the researchers define several aspects which must be taken into consideration when determining whether a setting is considered low-resource : 1) the amount of task-specific labelled data in the target language (or target domain)[5], which can be directly used to train a model 2) the amount of unlabelled data (most often used to learn word embeddings) 3) the availability of auxiliary data[6]4) the complexity of the task (more complex tasks will require larger amount of data).

Depending on these factors, a resource scenario might be deemed more or less rich in resource. In addition, it seems that the language in question also plays a role in how much data is required to trained a system. For morphosyntactic tagging specifically, [Plank et al., 2016] show that the performance varies between language families given the same amount of training data, with Slavic and Germanic languages faring worse than Romance or Semitic languages when the training dataset is really small. Thus the "richness" of a given given scenario is not only dependent on the availability of many different kinds of data and resources, but also on the language itself.

The availability of unlabelled data is sometimes assumed, as it doesn't require the kind of expertise central to annotated data. In reality however, many languages still lack these types of resources.

What can be done with raw text depends of course on the available quantity of the raw text, going from a small collection of transcribed data to a language with an important online presence with millions of tokens available in various genres and domains. It is clear that the size of the online presence in general will often coincide

[5]In non-standard text genres, domains or tasks, there might be little to no training data even for the languages which are more commonly studied in NLP.

[6]As they underline, auxiliary data can take many form such as external sources of informations like gazetteers, task-specific labeled data in another language or domain, or labeled data specific to another task.

with standardisation and the availability of other resources such as (grammar) books, linguistic analyses of the language, and mono- and bilingual lexicons.

In addition, the bigger the online presence the likelier it is that samples of this language where included in the corpora used to train language models [Devlin et al., 2019; Zhuang et al., 2021]. These resources, and the embeddings they create, have become commonly used resources in many NLP tasks, such as zero-shot parsing [Tran and Bisazza, 2019] where they provide a shared representation to help leverage an existing parser model to apply to a new language. In Figure 3.1 (from [Hedderich et al., 2021]) we see that even largely used language families are not covered by multilingual transformer models BERT and RoBERTa. On the positive side, there have been recent initiative to create resources for more languages, with initiatives focussing on specific languages or languages that face similar issues (here we point in particular to the work of [Lent et al., 2021] on building language models for creoles).

Figure 3.1 gives an overview of language families covered by such LLMs according to [Hedderich et al., 2021].

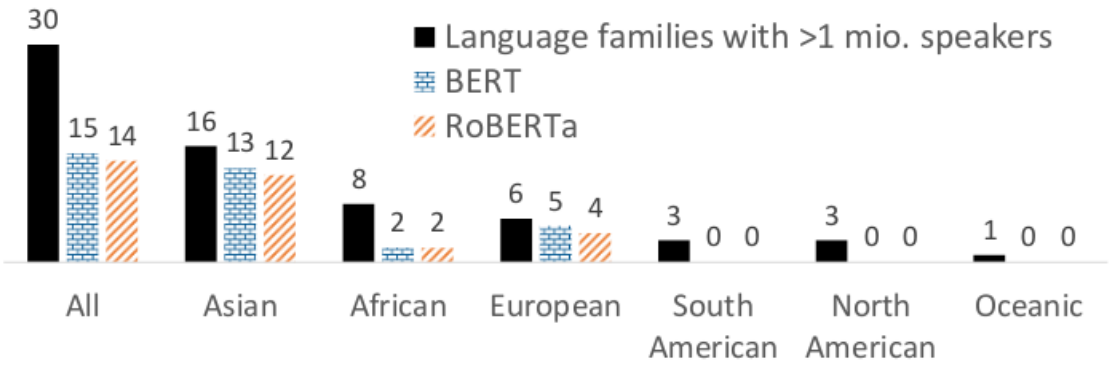


Figure 3.1: Language families with more than 1 million speakers covered by multilingual transformer models from [Hedderich et al., 2021]

More detailed information on the availability of various language resources for different languages can be found in [Joshi et al., 2020].

3.3.2 Rich Resource Languages

In linguistic annotation, varying resource availability poses distinct challenges and opportunities. One such scenario is that of rich resource languages.

Availability of Existing Treebanks: For a handful of languages, there may already exist a treebank in the desired annotation scheme that aligns closely with the genre of text targeted in the annotation project. Such a scenario is undoubtedly ideal. When this happens, the guiding objective should be to align as closely as possible with the existing resources[7]. This can be achieved by training a parser on the existing treebank. In all likelihood, this perfect scenario is not exactly what we encounter. Most likely, the existing treebanks differ slightly from what we hope to create. There might be features we want to introduce to query later on, or distinctions that were introduced in the original annotation that we deem irrelevant. We argue that in most cases, using the existing treebanks as training data to learn a parsing model is still the safest course[8], and that the added features can be built on top of the existing annotation.

Parsed Corpus as Treebanks: A question that often emerges in this context is whether a parsed corpus qualifies as a treebank. With contemporary parsers demonstrating increasingly high accuracy, one might be tempted to perceive a parsed corpus as equivalent to a treebank. However, a distinction is clear upon closer examination. Typically, treebanks, being valuable linguistic resources, necessitate a degree of human evaluation. This evaluation ensures the coherence and adherence of the annotations to syntactic criteria. Simply parsed texts, devoid of the human evaluation layer, fall short of fulfilling this criterion. As the field advances, this distinction will become increasingly nuanced, especially as annotation tools become more and more refined, and integrate human annotation with complex automatic predictions. Some tools can prioritise potential parsing errors for human review (through "active learning"), undergo

[7]This is also a clear opportunity to notice inconsistencies or errors in the original treebank, which can be discussed with their developer to improve upon the original resource

[8]Providing of course that the existing treebanks are of sufficient quality

frequent retraining to recognise unfamiliar annotated structures (via bootstrapping), and visually flag potential errors based on the low frequency of specific analyses. A glimpse of this evolving dynamic can be seen in one of the largest UD treebanks, the German Hamburg Treebank (HDT) [Foth et al., 2014]. In its creation, these cutting-edge techniques were used, making HDT straddle the line between a parsed corpus and an authentic treebank[9].

Pre-annotation bias: While using parsers to generate initial annotations is efficient, this approach is not without its pitfalls. One prominent concern is the introduction of biases into the annotation, skewing it in favour of the parser's output. This potential bias is not merely theoretical. Research, such as the study by [Fort and Sagot, 2010], demonstrates that in scenarios involving morphosyntactic tagging with pre-annotations, the annotation accuracy can significantly vary based on two primary factors:

- The training level of the annotator.
- The quality of the pre-annotation.

The study found that while untrained annotators saw an improvement in their accuracy when provided with pre-annotations, trained annotators show degraded performance when presented with suboptimal pre-annotations. In addition, the nature of errors observed also shifted. Pre-annotations decreased the number of random errors—often caused by lapses in attention—but increased the chances of systematic errors occurring.

As we will discuss in Chapter 5, when students are presented with partial or erroneous trees (some with conspicuous mistakes), there's a recurring tendency to under-correct. In many instances, students passively accept the parser output, reinforcing the notion that human review by specialists remains crucial for accurate linguistic annotation.

[9]Of course the line had already been straddled for a while, since treebanks have been built using automatic annotations for many decades now.

In summary, while rich resource languages offer a promising starting point for linguistic annotation, care must be taken to ensure the quality of the final treebank.

3.3.3 Low-Resource Scenarios

In this section, we will discuss scenarios which fall under the low-resource to varying degrees. We will showcase some of the strategies that are employed to overcome the encountered challenges. We give a short overview of different transfer learning strategies depending on the availability of data and resources, including model and annotation transfer, as well as unsupervised methods.

Transfer learning encompasses strategies to apply knowledge gained from one domain to improve performance in another related domain. It relies on the availability of auxiliary data, which can take many form. For treebank development, we focus on the two primary approaches which are model transfer and annotation transfer.

3.3.3.1 Exploring Neighboring Languages:

An effective strategy in low-resource scenarios is to investigate neighboring languages (which may by typologically, geographically or genealogically close). These languages often exhibit structural similarities or share linguistic features, providing a foundation for treebank development. While this method may not produce a completely accurate representation, it offers a valuable starting point for further refinements. With the advent of multi-lingual Large Language Models, zero-shot dependency parsing [Tran and Bisazza, 2019] has shown impressive results when training a parser across a diverse set of typological treebanks. However, the outcomes can vary significantly based on two main factors: 1) the presence of typologically similar languages in the training treebanks, and 2) the quality of the embedding, which is influenced by the similarity in writing systems and spelling found in the vast raw corpora used for the multi-lingual LLM [Wang et al., 2019]. Using this preliminary framework, coupled with zero-shot parsing results, can guide the creation of a pilot treebank for the language

lacking resources. This foundational treebank can be seen as a starting point to establish guidelines, refine methodologies, and offer a roadmap for subsequent annotations, fostering a cycle of quality and consistency improvements.

3.3.3.2 Utilizing Parallel Corpora:

Another approach is the use of parallel corpora, where the same text is available in multiple languages. This, of course, is no longer an absolute "no-resource" scenario, as at least a parallel corpus exists. For instance, the Bible, being one of the most translated books worldwide, provides a rich corpus in various languages. As demonstrated by [Agić et al., 2015], training a tagger using parallel versions of the Bible is feasible, capitalising on the consistent narrative across translations. By aligning texts from the resource-scarce language with those from a well-studied language, one can infer linguistic structures, shedding light on the former's syntactic and morphological attributes. While this method undoubtedly offers valuable insights, it does come with the risk of incorrectly transferring grammatical structures that might not align with the target language's syntax. However, such biases are not exclusive to this method; they reflect a broader challenge also present when linguists take on the annotation, where a linguist's native language and academic training will influence their analysis of the language even down to the terminology used to describe the language. It raises the question of whether an entirely unbiased analysis of a new language is even conceivable.

In conclusion, while the no-resource scenario poses significant challenges, it's not an insurmountable obstacle. With ingenuity and leveraging available resources, even in their indirect form, it's possible to lay the groundwork for treebank development in such languages. In the next section we will discuss a bit more the concept of low-resource language and explore various methods to deal with these languages.

3.3.3.3 Model Transfer

Model transfer involves leveraging models trained on one language (source) to aid parsing or annotation tasks in another language (target).

Delexicalized Transfer Originally introduced by [Zeman and Resnik, 2008], delexicalized transfer requires:

- A treebank in the source language (preferably a related one).
- A tagger for the target language, which employs the same tagset as the source.

It relies on a shared representations, provided by the part-of-speech tags in both languages. While this method is quite straightforward to employ and can reach good results, it might falter when significant word order variations are observed between the source and target languages. As noted by [Aufrant et al., 2016], shared representations may obscure intrinsic word order differences between languages, leading to biased parser outcomes.

Data Selection Building on delexicalized parsing, [Søgaard, 2011] propose an enhancement applicable to less related language pairs. The authors employ instance weighting—a technique usually reserved for sampling bias correction—to cherry-pick sentences from the source data resembling the target data. This is achieved by gauging the perplexity of language models trained on POS tag sequences, and selecting sentences where the perplexity is lower (meaning that the source and target POS sequences resemble each other). Notably, this approach has proven effective even for traditionally unrelated language pairs.

Data Transformation Data transformation, extending from [Søgaard, 2011], facilitates the adaptation of parsing models in scenario where target and source languages showcase different words orders. It requires :

- A treebank in the source language.

- Morphosyntactic tags for the target language (with a consistent tagset).

There are two strategies tested within this paradigm:

- **Data-driven approach**: Proposed by [Aufrant et al., 2016], it involves reordering the training corpus optimally using a POS-based Language Model, to resemble the target corpus. To achieve this, the authors consider all permutations within a window of three tokens.

- **Rule-based approach**: This technique, introduced in [Aufrant et al., 2016] as well, leverages linguistic features extracted from the World Atlas of Language Structures (WALS) [Dryer and Haspelmath, 2013]. Based on these linguistic features, and in order to artificially create a source treebank that more closely resembled the target language, the method automatically "prunes" or transforms the source treebank to align with the target. For example, in order to transfer a parsing model from English to Czech, the authors will prune articles, while the transfer from English to Japanese will require a reordering of adpositions and nouns.

In scenarios where treebanks from multiple source languages are available, multi-source transfer provides a promising avenue. It amalgamates knowledge from various sources to improve parsing performance in the target language.

3.3.3.4 Annotation Transfer

Annotation transfer, or projection, generally operates based on parallel corpora between the source and target languages, combined with word-level alignments. The goal is to transfer annotations from a source language with an available treebank to a target language.

Annotation Projection with Parallel Text and Parser Training For this method, the required resources include:

- **Source**: An available treebank.

- **Target**: An aligned corpus and monolingual knowledge of the target language. If the alignment is of poor qualities, heuristics might be used to filter misaligned sentences and avoid introducing noise.

[Hwa et al., 2005] initiated this approach by annotating the English side of a parallel corpus, subsequently projecting these annotations to the other language, and finally training a parser on the derived annotations. This method relies on the Direct Correspondence Assumption, which posits that syntactic dependencies in one language may lead to similar dependencies in its translated counterpart.

However, the researchers observed that this direct projection did not cater to language-specific elements, such as aspectual markers in Chinese that don't have direct English counterparts. To rectify this, they integrated post-projection correction rules based on target language-specific knowledge, which significantly boosted accuracy by 30%. These rules, expressed in a tree-based pattern-action formalism, perform local transformations based on the projected analysis. An exemplar rule is the determination that a measure word modifies the preceding number or determiner and cannot have its modifiers.

Partial Annotation Projection with Sentence-Aligned Bi-texts Building on the concept of annotation projection, [Lacroix et al., 2016] employed an automatic alignment methodology They utilised Giza++ for alignment and retained only those tokens that were candidates for alignments in both directions. They then implemented heuristics, like ensuring aligned words have the same POS tag and discarding sentences with less than an 80% alignment rate. The result is a partially annotated treebank that, while possessing an incomplete structure, is presumed to be largely accurate. This treebank was then used to train a parser. Their dataset, derived from a Europarl subset, covered languages including German, English, Spanish, French, Italian, and Swedish.

3.3.3.5 Unsupervised Methods

When looking into unsupervised syntactic structure analysis, representation learning stands out as a promising approach. In particular, architectures such as autoencoders— neurals networks (NN) designed to learn a representation for an input and subsequently reproduce the input as its output, provide a useful strategy as they learn a compressed and efficient representation of the input, commonly referred to as a latent-space representation.

Autoencoders and Syntactic Structures At its core, a syntactic structure is a form of representation, providing an organized overview of linguistic relationships. One might argue that such structures also compress linguistic information, suggesting ways in which elements combine to create meaning. This parallel between syntactic structures and the compact representations autoencoders generate suggest that the latter might be used to induce valuable syntactic structures.

Deep Inside-Outside Recursive Autoencoders (DIORA) A pivotal work in this domain is the Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Autoencoders (DIORA) [Drozdov et al., 2019]. Behind DIORA's design is the hypothesis that optimal sentence compression is achieved by adhering to the true syntactic structure of the input. The approach modernizes previous latent tree chart parsers by integrating the inside-outside algorithm. This algorithm aims to recreate each input word from its surrounding context.

Relevance and Evaluation of Latent Tree Learning Recent advancements in latent tree learning enable neural networks to parse and interpret sentences without any exposure to ground-truth parse trees during training. This unsupervised approach raises pertinent questions, especially regarding evaluation methodologies. Extrinsic evaluation methods might compare outputs to benchmarks like the Penn Treebank. Meanwhile, intrinsic evaluation focuses on the semantic utility of the method—assessing, for example, its applicability in specific semantic tasks.

Conclusively, unsupervised methods, especially autoencoders, and latent tree learning models, introduce a paradigm shift in the approach to syntactic structure identification, focusing not only on structure but also on the innate meaning and semantics derived from it.

3.4 Conclusion

In Chapter 3 we have focussed on treebank development practices, highlighting the various subtasks required, and presenting some of the challenges tied to varying resource scenarios. It highlights foundational subtasks like transcription, tokenization and tagging (both for part-of-speech and morphological features) emphasizing in particular the approach undertaken in the UD project, as it will be the basis of our framework in latter chapters. We then look at scenarios where varying degrees of resources are present, while keeping the various facets that are hidden behind the expression "low-resource". For rich resource scenarios we look at some of the difficulties that remain when developing high quality treebanks. In lower-resource scenarios, we discuss challenges and present various strategies based on transfer learning, mainly model transfer and annotation transfer. We end on a brief description of unsupervised methods used and how they bring forward legitimate questions regarding the evaluation of treebanks.

Part II

Contributions

Chapter 4

Annotation guidelines and grammars

Chapter contents

4.1 Concerns when developing annotation guidelines 85

4.1.1 Annotation guidelines : content and structure 85

4.1.2 Problems faced by annotators 90

4.1.3 Linguistic criteria and external factors 92

4.1.3.1 Learnability 93

4.1.3.2 Error vs linguistically debatable analysis 95

4.2 Case Study n°1 : Multi-word Expressions 96

4.2.1 Idioms and syntactic irregularity 97

4.2.1.1 MWE and tokenisation in UD 101

4.2.2 Propositions for the encoding of MWEs in UD 104

4.2.3 Conclusion 108

4.3 Case Study n°2 : Naija 109

4.3.1 Linguistic context 110

4.3.2 Treebank development 111

4.3.2.1 Corpus Metadata 111
4.3.2.2 Transcription 112
4.3.2.3 Morphosyntactic analysis 113
4.3.2.4 Polycategoriality and polyfunctionality 115
4.3.2.5 Surface-Syntactic UD annotation 116
4.3.2.6 Evaluation 120
4.3.3 Some idiosyncratic syntactic constructions of Naija 123
4.3.3.1 Clefts in Naija 123
4.3.3.2 Interrogatives 125
4.3.3.3 Serial Verb Constructions 126
4.3.4 Conclusion on the NaijaSyncor treebank 128
4.4 Conclusion and perspectives for the chapter 128

As we have shown in Chapter 2 the process of adding linguistic annotation onto text-based corpora has a longstanding history. What began as individual practices has been refined and benefits today from computerized resources, open formats, shared standards and communities that attempt to build consistent annotations across a variety of languages. In this chapter we will focus on treebank annotation processes, syntactic annotation schemes and their combined outputs, namely treebanks and annotation guidelines. We will investigate the following research questions :

- What content should appear in annotation guidelines and how to structure such guidelines ?
- What criteria should we keep in mind when designing annotation schemes ?
- What external factors influence the annotation scheme ?

Section 4.1 will be dedicated to discussing these 5 questions, in section 4.3 we will look at the development of a treebank for Nigerian Pidgin (Naija) for examples of how such problems can be addressed and in section 4.2 we will look at an attempt to improve annotation and annotation guidelines for the linguistic phenomena of multiword expression in the Universal Dependencies annotation scheme and see how that relates to the points made in section 4.1.

This chapter is based on several pre-existing publications:

- Nigerian Pidgin (Naija) :
  - Bernard Caron, Marine Courtin, Kim Gerdes and Sylvain Kahane. (2019). A Surface-Syntactic UD Treebank for Naija, Proceedings of the 17th international conference on Treebanks and Linguistic Theories (TLT), SyntaxFest, Paris.
  - Courtin Marine, Caron Bernard, Gerdes Kim, Kahane Sylvain. (2018). Establishing a language by annotating a corpus: The case of Naija, a post-creole spoken in Nigeria, Proceedings of the workshop on Annotation in Digital Humanities (AnnDH), Sofia.

- Multi-word expressions :
  - Kahane Sylvain, Courtin Marine, Gerdes Kim. (2018). Multi-word annotation in syntactic treebanks: Propositions for Universal Dependencies, Proceedings of the 16th international conference on Treebanks and Linguistic Theories (TLT), Prague.

Part of this work has been carried out in the framework of the French National Research Agency (ANR) project NaijaSynCor, headed by B. Caron. In the following, we will focus on our experience of the annotation process in general and more specifically, our contributions to the development of the Naija annotation guidelines.

4.1 Concerns when developing annotation guidelines

4.1.1 Annotation guidelines : content and structure

The first aspect we will focus on is how guidelines are organised. This can vary from project to project but some structures are frequently found. If we look for example at the POS Tagging guidelines of the Penn Treebank, we find that there is a short introduction, followed by a section that lists part-of-speech labels and their relevant tags, briefly defines them and gives a few examples.

There is also a section dedicated to "problematic cases" which belong to one of two cases : 1) examples where the annotator is likely to hesitate between a couple of tags and 2) collocations, which are word sequences that may function more or less as fixed expressions. An excerpt of this section illustrating the first case of multiple possible tags is provided in Figure 4.1 explaining how to discriminate between adjectives (JJ) and gerunds/ present participles (VBG).

JJ or VBG

The distinction between adjectives (JJ) and gerunds/present participles (VBG) is often very difficult to make. There are a number of tests that you can use to decide. Be sure to apply these tests to the entire sentence containing the word that you are unsure of, not just the word in isolation, since the context is important in determining the part of speech of a word.

A word is an adjective (JJ):

- if it is gradable—that is, if it can be preceded by a degree adverb like *very*, or if it allows the formation of a comparative.

EXAMPLE: Her talk was very interesting/JJ.
Her talk was more interesting/JJ than theirs.

Figure 4.1: Excerpt of the POS Tagging Guidelines of the Penn Treebank explaining how to distinguish between adjective and gerund.

To help the annotators resolve their tagging questions, help can be provided in the form of direct examples or criteria used to discriminate between the candidate tags. Decision trees will sometimes be used to guide the annotators and show them the reasoning behind the annotation choice. They are most often included to facilitate the

annotation of the more complex phenomena, where several tests are needed to infer the required annotation. They lessen the ambiguity, but some amount of interpretation may still be needed on the part of the annotator. In the PARSEME project [Savary et al., 2017], such decision trees are used for example to determine whether an expression can be classified as a verbal multi-word expression (VMWE), and if so, to identify the kind of VMWE encountered.

Before looking at the decision tree, we will give a brief idea of what VMWE are. These are expressions like `kick the bucket` or `make a decision` which are often characterized by semantic non-compositionality. To qualify as VMWE they have to include at least a verb and at least one other word. They are not necessarily contiguous (**`took me by surprise`**) and can be subject to syntactic variations (**`nobody was taken by surprise`**), it is therefore important to identify which components are lexicalized , i.e. necessary for there to be a VMWE (we have represented them in bold in the text). Now that some context is given, we will present the decision tree (see Fig. 4.2) found in the PARSEME annotation guidelines [Khelil et al., 2022]. The first test that is applied is S.1, which requires that the candidate for VMWE-status only contain one verb, which should function as the head of the whole[1]. If the candidate passes the test, the annotator can move on to test S.2, which requires that the verb has only one lexicalized dependent (the dependents will be underlined). This is the case for example of **`taking into account`**, where the verb has one dependent[2]. A candidate that wouldn't pass the test however is `to` **`let the cat out of the bag`**, as the verb has two lexicalized dependents. In this case, we would still need to apply the Verbal Idiom (VID) tests to know whether or not the expression is qualified as a VMWE (of VID type) or is disqualified. These VID tests are summarized in another decision tree which we will not describe further here.[3]

[1]Exceptions are made for expressions where the verb is not the head of the whole (**`the making of the decision`**), if they are a meaning-preserving syntactic variants of an expression that passes the test (**`to make a decision`**).

[2]Whether the dependent is `account` or `into` would depend on whether content words or functional words are preferred as heads in the schema, but the choice that is made doesn't matter for VMWE candidacy.

[3]We will however spoil the surprise that **`letting the cat out of the bag`** is indeed considered as a verbal idiom, qualifying through the lexical inflexibility test.

↳Apply test S.1 (prev. 6) - [**1HEAD**: Unique verb as functional syntactic head of the whole?]
- ↳ **NO** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
  - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
  - ↳ **NO** ⇒ It is not a VMWE, **exit**
- ↳ **YES** ⇒ Apply test S.2 (prev. 7) - [**1DEP**: *Verb v has exactly one lexicalized dependent d?*]
  - ↳ **NO** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
    - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
    - ↳ **NO** ⇒ It is not a VMWE, **exit**
  - ↳ **YES** ⇒ Apply test S.3 - [**LEX-SUBJ**: *Lexicalized subject?*]
    - ↳ **YES** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
      - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
      - ↳ **NO** ⇒ It is not a VMWE, **exit**
    - ↳ **NO** ⇒ Apply test S.4 (prev. 8) - [**CATEG**: *What is the morphosyntactic category of d?*]
      - ↳**Reflexive clitic** ⇒ Apply IRV-specific tests ⇒ *IRV tests positive?*
        - ↳ **YES** ⇒ Annotate as a VMWE of category **IRV**
        - ↳ **NO** ⇒ It is not a VMWE, **exit**
      - ↳**Particle** ⇒ Apply VPC-specific tests ⇒ *VPC tests positive?*
        - ↳ **YES** ⇒ Annotate as a VMWE of category **VPC.full** or **VPC.semi**
        - ↳ **NO** ⇒ It is not a VMWE, **exit**
      - ↳**Verb with no lexicalized dependent** ⇒ Apply MVC-specific tests ⇒ *MVC tests positive?*
        - ↳ **YES** ⇒ Annotate as a VMWE of category **MVC**
        - ↳ **NO** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
          - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
          - ↳ **NO** ⇒ It is not a VMWE, **exit**
      - ↳**Extended NP** ⇒ Apply LVC-specific decision tree ⇒ *LVC tests positive?*
        - ↳ **YES** ⇒ Annotate as a VMWE of category **LVC**
        - ↳ **NO** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
          - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
          - ↳ **NO** ⇒ It is not a VMWE, **exit**
      - ↳**Another category** ⇒ Apply the VID-specific tests ⇒ *VID tests positive?*
        - ↳ **YES** ⇒ Annotate as a VMWE of category **VID**
        - ↳ **NO** ⇒ It is not a VMWE, **exit**

Figure 4.2: Decision tree for the identification of verbal multi-word expressions (VMWE) in the PARSEME guidelines [Savary et al., 2017].

The decision trees help to formalize the tests and criteria to apply to reach the correct decision, but they are not a substitute to human interpretation. Referring back to VMWE, human judgments still intervenes in many cases : to select the initial candidates, to decide wether a word is lexicalized in the expression or not, to assess subtle differences in meaning... . While the decision tree provides a very thoughtful aid to decision-making, and greatly increases the chance of selecting the required

annotation, it is not a substitute to human judgment. A fully formalized set of tests that would always select the "correct" annotation would not require human input, as discussed by Guillaume and Fort in the following quote :

> Needless to say, it is impossible to fully formalize the guide using (rewriting) rules. Otherwise, it would mean that the annotation can be fully automated without any human judgment. (our translation) [Guillaume and Fort, 2013]

If we look at another example of guidelines, the UD guidelines, we can see that the structure is similar, but the format and the multilingual nature of the treebanks also play a role in the design of the guidelines. The guidelines are presented as a website which is compiled from markdown pages updated by the contributors, as well as pages that are automatically generated on the basis on the treebanks themselves.[4] Overall the guidelines are made up of 3 sections : 1) the guiding principles behind the universal annotation scheme, 2) guidelines for various constructions from simpler (and more frequent) to more complex and/or rarer constructions, and 3) a section which really delves into the various labels (for morphological features, part-of-speech and dependency functions) and how they are to be used (i) language-generically, (ii) language-specifically.

For each of the relation labels in Fig 4.3, there is a dedicated page describing its use across languages, this page will often include examples in English but not exclusively. If the relation is used in a specific language, there will be a language-specific page which often includes the subrelations used in that language. As an illustration, Figure 4.4 displays an example of the `acl` (clausal noun modifier) relation extracted from our work on Naija.

The guidelines are constantly evolving and these changes are tracked explicitly in the guidelines themselves which will often mention major changes that happened between the different versions of the treebanks. The content of the guidelines and the treebanks themselves are stored using a version system and hosted on github, allowing

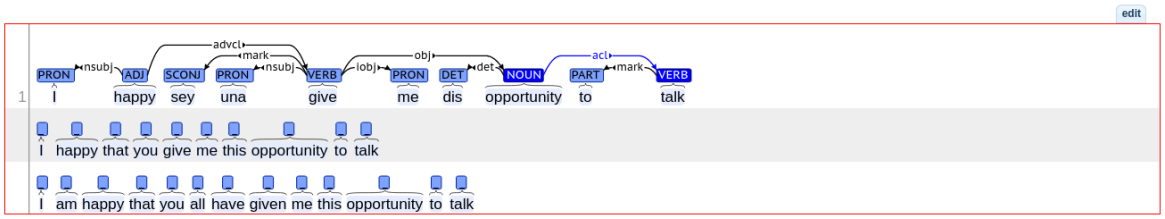
[4]See The Nigerian Pidgin treebank hub for an example of this kind of page.

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Figure 4.3: Table of dependency relation labels in the UD guidelines

acl: clausal modifier of noun

The `acl` relation is used for clauses that modify a noun or an adjective. Adjectives can also have clausal complements with their own subject, in that case we use the ccomp relation instead.



See acl:cleft for clefted modifiers and acl:relcl for relative clauses.

Figure 4.4: UD guidelines for the `acl` relation in Naija (Nigerian Pidgin).

contributors and users to observe every single update and understand the reasons motivating the changes.

The UD project also uses an issue tracking system, to address problems in the data and linguistic concerns about the annotation which allows contributors to share knowledge and experience about linguistic phenomena, discover inconsistencies in the annotation, find cases not currently covered by the guidelines, or poorly documented.

Sometimes, the discussion might result in a change in the guidelines at the universal level if enough languages are concerned and there is enough data to support it.

We only touched on a few of the challenges faced when developing treebank annotation guidelines. It is clear that this process requires a high level of language and linguistic competence on behalf of the involved collaborators. Once the guidelines established, they are to be used by annotators to create new treebanks or continue further developing existing treebank resources.

4.1.2 Problems faced by annotators

Annotation guidelines are extremely useful to produce valuable, consistently annotated treebanks. Nonetheless, human annotators may face multiple problems when processing new corpora for a couple of reasons.

First of all, language may contain ambiguities which the annotator is not able to solve using a unique decision. Furthermore, guidelines may not be developed enough to clearly specify how to handle each and every linguistic phenomenon, leading to annotation uncertainty. Some linguistic phenomena may be open to interpretation, making it difficult to achieve consistent annotations across different individuals. For example the distinction between argument and adjunct is a common sticking point. When the distinction cannot be made consistently, it may be more strategic to avoid making it. This strategy of underspecification was used for the Penn Treebank :

> It proved to be very difficult for annotators to distinguish between a verb's arguments and adjuncts in all cases. Allowing annotators to ignore this distinction when it is unclear (attaching constituents high) increases productivity by approximately 150-200 words per hour. Informal examination of later annotation showed that forced distinctions cannot be made consistently. [Marcus et al., 1993]

Indeed, the argument/adjunct distinction requires to consider the semantics of the structure, and perhaps to rely on an exterior source of knowledge such as a valency

lexicon to make an informed decision. The reliance on a larger context and an external source of knowledge have both been considered as relevant factors in increasing annotation complexity [Fort et al., 2012].

Even with clear guidelines, the complexity of the annotation which is increased through aspects such as degree of ambiguity, weight of the context and tagset dimension [Fort et al., 2012] can lead to inconsistent annotations over time, even for a single annotator.

A second problem relates to scalability issues: guidelines designed on a small dataset might not scale effectively when applied to larger, more diverse corpora. Moreover, natural language is inherently variable, with nuances, idioms, and colloquialisms. Guidelines might not account for all these variations, especially in very technical or in dialectal corpora. As annotators discover unanticipated issues, guidelines may need revisions. Delays in updating them can slow down the annotation process. Annotators may struggle identifying the lacks and once identified, they may potentially contribute to further elaborating the guidelines beyond doing their annotation job. When working on low-resourced languages, this picture tends to be the case and annotation and guideline development can be considered as iterative inter-dependent processes. However, for well-documented languages, a risk may be to end up with guidelines that are too detailed or intricate. As a consequence, annotators may struggle to remember all the details or apply them inconsistently.

In any case, to achieve faithful annotations, specific training is strongly recommended. Annotators might require extensive training to understand and implement complex guidelines effectively, making the onboarding process time-consuming. Once trained, the continuous annotation can be tiring, leading to reduced attention to detail and potential mistakes if guidelines are too cumbersome. Specific annotation tools may be very helpful to facilitate the annotation process and to avoid errors due to human inattention.

In summary, while annotation guidelines are crucial for maintaining consistency and quality in corpus processing, they come with challenges tied to their clarity, complexity, and applicability in diverse linguistic contexts.

4.1.3 Linguistic criteria and external factors

The objective of achieving high-quality annotation guidelines can be addressed through different aspects:

- the learnability of highly accurate parsers using corpora processed using these guidelines.
- a high degree of inter-annotator agreement when annotating corpora using these guidelines, which indicates stability of the annotation.

We will address these issues below and keep them in mind when dealing with the two case studies on multi-word expression and low-resource languages.

The importance of consistent annotation guidelines, consistent annotations and their link with learnability or trainablity (of parsers) can be recalled using an excerpt of [Gerdes, 2018]:

> The basic idea is that coherently annotated data is easier to learn than incoherent data. Equally, local rules are easier to learn than long-distance relations. It is thus possible to compare different annotation schemes by comparing parser performance on the same texts. Of course, we always also measure the limitations and tendencies of the parser itself, and moreover, neural network based parsers are increasingly capable of discovering complex relations that are distributed widely in a sentence. Yet, if different parsers give better results on one annotation scheme over another, we can conclude that this annotation scheme is preferable in some sense, even more so if the argument can be corroborated by theoretical considerations about the annotation scheme.

While this does apply to isomorphic schemes, which can be converted from one to the other without a loss of information, in the case of two annotation schemes with different degrees of granularity, the comparison doesn't stand. The simple one will be more easily learned, but it doesn't necessarily mean it is preferable. In the end, the

choice of the annotation scheme will depend on the application or needs that we have in mind, and what we consider feasible given the resources at hand.

While UD provides a harmonized annotation scheme across languages, the linguistic diversity and structural variations still present a challenge for parsers to learn and generalize effectively across languages. The learnability issue in the UD framework has been raised by several researchers [Gerdes and Kahane, 2016; Rehbein et al., 2017].[5] It has been observed that the price to pay for this harmonized annotation scheme, is a loss in parsing accuracy as compared to language-specific schemes.

> Universal Dependency (UD) annotations, despite their usefulness for cross-lingual tasks and semantic applications, are not optimised for statistical parsing. In the paper, we ask what exactly causes the decrease in parsing accuracy when training a parser on UD-style annotations and whether the effect is similarly strong for all languages. [Rehbein et al., 2017]

4.1.3.1 Learnability

To analyse this issue, we will focus on the question of cross-lingual word order differences, and its impact on learnability. To this end, measures such as *dependency length* (the linear distance between a dependant and its governor, in number of words) and *head-direction entropy* (a measure of the variability of word order between dependant and governor) can be used.

In their paper Gulordava and Merlo [2016] use artificial data to investigate the influence of word order properties on the performance of statistical parsers. To this end, they permute sentences from treebanks while keeping the dependency structure intact and feed this data to parsers to see how they respond. The permutations aim to minimize either dependency length or head-direction entropy[6]. They find that both

[5]One thing that should be noted on the front of 'learnability' is that learnability can only been compared between equivalent annotation schemes (annotation schemes are equivalent if they can be converted into one another, which means there is no information loss). Otherwise there will be a bias towards the simpler scheme.

[6]There are several ways to characterise head-direction entropy, here the authors chose to compute the entropy for each (rel, h, c) triplet with rel being the label of the relation, h the part-of-speech of the governor and c the part-of-speech of the dependant. These entropies are then weighed according

longer dependencies and a higher head-direction entropy negatively impact parser performance. This is useful when attempting to evaluate parser performance across languages and corpora, which might have very different word order properties, but it also seems like it might be interesting to consider from an annotation design perspective.

In the same vein [Rehbein et al., 2017] investigate how the annotation decisions in the UD framework impact the learnability of the annotation by parsers. Specifically, they try to tease apart whether a function-head encoding (similar to what is done in SUD) or a content-head encoding (similar to what is done in UD) yields higher accuracy and which order property is linked to the variation in performance. They find that most treebanks benefit from a conversion to a function-head encoding and that the relative improvement also depends on the language.[7] Considering that the conversion is rather minimalist (based on POS tags and dependency labels) with no language-specific adaptation of the rules, we think it is likely that the results are underestimated. They then show that their conversion results in a lower head-direction entropy for most languages, and that the improvement in performance is linked to this lower head-direction entropy in the converted treebanks. This result is consistent with the [Gulordava and Merlo, 2016] paper.

Their explicit goal is to facilitate the design of optimised annotation schemes :

> This provides interesting avenues for future research, as language generalisations might help us to design treebank encoding schemes that are optimised for specific languages, without having to repeat the same effort for each individual language [Rehbein et al., 2017]

If one of the intended goals is to develop a more learnable annotation, the influence of word order properties such as dependency length and head-direction entropy should be taken into account, and between two possible analyses equal on all other accounts we should favour the analysis that minimises those.

Of course, one could argue that the differences observed are merely a product of the

to the frequency of the triplet in the corpus.

[7]The Turkish treebank being the exception. One possible explanation would be that the conversion from UD to SUD requires specific rules not yet implemented for this language.

parsers used for the experiments, in that their design may be biased towards learning one scheme rather than another. For example it has long be known that parsers perform better on shorter dependencies than long-ranging ones, and a function-head encoding tends to result in shorter dependencies. There might be a point in the future where the length of dependencies is not a factor anymore when it comes to learnability, in which case selecting an annotation scheme based on this factor might be short-sighted. Indeed, as pointed out by [Fort, 2016] annotated corpora generally have a much longer lifespans than the tools we use to build them, and while the annotations from the Penn Treebank are still used to this day, the same cannot be said of Fidditch (Hindle 1983, 1989), the parser used to provide the initial parse. Nevertheless this aspect of learnability is one of the questions that often guides designers of annotation schemes.

It should be noted however that learnability is not sufficient to produce a good annotation, as trivial annotations (for example chain-linked) are highly learnable yet convey none of the information we want our system to learn. This learnability criteria should therefore intervene only to decide between several competing analyses which we all deem adequate.

4.1.3.2 Error vs linguistically debatable analysis

Among the most common ways of evaluating annotation inside treebanks two methods in particular stands : the evaluation with respect to a "gold standard" or ground truth, and the measure of an inter-annotator agreement.

In their position paper, the authors [Plank et al., 2014] question whether disagreements between annotators should be minimized rather than embraced. They present an empirical analysis of part-of-speech annotated data sets that suggests that disagreements are systematic across domains and to a certain extent also across languages.

They addressed the issue whether a lack of inter-annotator agreement is due to annotator inconsistencies and annotator errors or whether there are linguistic explanations to multiple distinct parse trees. They showed that even in the absence of annotation guidelines only 2% of annotator choices are linguistically unmotivated.

Furthermore, they show that disagreements between professional or lay annotators are systematic and consistent across domains and some of them are systematic also across languages. In addition, they present an empirical analysis of POS annotations showing that the vast majority of inter-annotator disagreements are competing, but valid, linguistic interpretations. They propose to embrace such disagreements rather than using annotation guidelines to minimize inter-annotator disagreement, which could bias the models in favour of some linguistic theory.

4.2 Case Study n°1 : Multi-word Expressions

In the following, we investigate how to analyze syntactically irregular expressions in a syntactic treebank. We place our analysis in the Universal Dependency framework (UD), which constitutes a large community of more than 100 teams around the globe [Nivre et al., 2016]. We distinguish such Multi-Word Expressions (MWEs) from comparable semantically non-compositional expressions, i.e. idioms.

We take special care to discuss the case of functional MWEs, which are particularly problematic in UD, given its principle of making content word heads, as has been discussed in previous sections. Functional MWEs are multiword expressions that fulfill the same role as a function word (determiner, adposition, coordinating conjunction...) such as `a lot of`, `on top of` or `not to mention`, as opposed to non functional (or lexical) MWE .

In every linguistic annotation project, the delimitation of lower and upper boundaries of the annotation units constitutes a basic challenge. In syntactic annotation, the lower boundaries are between morphology and syntax, the upper boundaries between syntax and discourse organisation. The problem of MWEs is one relating to the lower boundaries in syntactic treebank development. We discuss the problem caused by idioms in syntactic annotation. The literature on idioms and MWEs is immense [Fillmore et al., 1988; Mel'čuk, 1998; Sag et al., 2002]. Our goal is not to mark the extension of MWEs on top of the syntactic annotation (see the PARSEME project [Savary et al., 2017] which we discussed in section 4.1.1 and their work on verbal multiword

expressions). Our purpose is to tackle the impact of idiomaticity on the syntactic annotation itself.

Most idioms (such as `kick the bucket` or `green card`) do not cause any trouble for syntactic annotation as their internal syntactic structure is regular (and it is precisely because they have a clear internal syntax that they are idioms and not words). Some expressions, however, such as `not to mention`, `heaven knows who`, `by and large`, `Rio de la Plata` (in English), are problematic for a syntactic annotation, because they do not perfectly respect the syntactic rules of free expressions.

We propose two contributions:

- For a coherent annotation it is crucial to distinguish *syntactically irregular* structures from *semantically non-compositional* units. These notions are highly correlated but distinct and we propose criteria to distinguish them.
- We explore different ways of annotating these two kinds of Multi-Word Expressions and their combinations in a syntactic treebank, with a special focus on functional MWEs.

Section 4.2.1 proposes a simple typology of MWEs opposing semantic compositionality and syntactic regularity.

We lay the basis of our analysis by discussing the syntactic units of a dependency annotation, and point to problems in the used UD scheme (version 2.1). In section 4.2.2, we propose to analyze MWEs with an internal syntactic structure according to their level of syntactic regularity, and show how a MWE can be introduced into the CoNLL-U format as a unit with its own POS. Furthermore, we introduce two convertible dependency schemes for functional MWEs before concluding.

4.2.1 Idioms and syntactic irregularity

We distinguish idiomatic expressions from syntactically irregular constructions. Idiomaticity is a semantic notion and we want to annotate semantic apart from syntax.

An MWE is an idiom (i.e. non-compositional) if its components cannot be chosen individually by the speaker (`kick the bucket` is chosen as a whole and there is

no possible commutation on its components) [Kahane and Gerdes, 2022, section 7.2 p.182].[8]

An MWE is a collocation (i.e semi-compositional) if one of its component is chosen freely (the basis) and the other one (the collocate) is chosen according to the basis (in `wide awake`, `wide` can be suppressed and `awake` keeps the same contribution: `awake` is the basis and `wide` is a collocate expressing intensification with `awake`). We also consider three levels of syntactic irregularity. First, natural languages contain some syntactic subsystems which do not follow the general properties of syntactic relations. For instance, most languages have particular constructions for named entities such as dates or titles. English has a regular construction N N, where the second noun is the head (`pizza boy`, `Victoria Lake`) but it also has a subsystem where the first noun is the head, used for named entities (`Lake Michigan`, `Mount Rushmore`, `Fort Alamo`). These subsystems are in some sense "regular irregularities", that is, productive unusual constructions. Similarly, English produces a high number of multi-word adverbs from a preposition and a bare noun as in `on top (of)` or `in case (of)`, thus forming another sub-system that does not conform to the typical syntactic system of English.

Second, languages have non-productive irregular constructions. Most of these irregular constructions are idioms, but some are compositional. This is the case of the French `peser lourd` 'weigh a lot/be significant', lit. 'weigh heavy', where `lourd` is an adjective that commutes only with NPs (`peser une tonne` 'weigh one ton').[9] Even the commutation with its antonym `léger` 'light' is impossible. Another example is French `cucul la praline` 'very silly', lit. 'silly the praline'. It is a collocation: the adjective `cucul` can be used alone and the NP `la praline` is an intensifier. The POS of the units are clear, and the dependency structure can be reconstructed, but it is unusual to have an NP modifying an adjective. We consider four cases of non-productive

[8] An idiom can be semantically transparent [Svensson, 2008]. For example, it is quite clear that a washing machine is a machine that is used to wash something, but it is an idiom because it is arbitrary that this denotes a machine for washing clothes and not a dishwasher or a high-pressure water cleaner.

[9] It is not completely how the relation between `peser` and `lourd` should be analyzed in UD. On one hand `lourd` could be analyzed as an **xcomp** of `peser` (a "predicative or clausal complement without its own subject" cf. the UD guidelines), but on the other hand we would lose the fact that `lourd` is in the paradigm of NPs analyzed as **obj**.

irregular constructions.

a) Structures with a **clear POS and dependency structure** but that function as a whole differently than their syntactic head, for example :

• the coordinating conjunction headed by a verb `not to mention` as in `they gave us their knowledge, not to mention their helpfulness`

• the adjective `top of the range` headed by a noun as in `a very top of the range restaurant`

• the French pronoun `Dieu sait quoi` 'heaven knows what' headed by a verb.

b) For some sequences, the **POS are clear, but don't agree with the dependency structure** or vice versa, **the dependency structure is clear but the POS have to be reconstructed** (the latter has to be reconstructed diachronically). For example :

• the French pronoun `n'importe quoi` 'anything', lit. 'no matter what';[10]

• the adverb `by and large` – `by` being originally an adverb.

c) Other sequences have **no clear internal dependency structure at all, while the POS remain clear**:

• `each other`

• French `à qui mieux mieux` 'each trying to do better than the other', lit. 'to whom better better'

• French `quand même` 'at least', lit. 'when even/least'

d) Some sequences have **neither clear POS nor an internal structure** in the language of the corpus.

• The adjective `ad hoc`

• `Al Qaeda` which is a proper noun

• the French subordinating conjunction[11] `parce que` 'because' [12].

[10]Diachronically, `quoi` is the subject of `importe` but now it is recognized as an object due to its

	Compositional	Semi-compositional (collocation)	Non-compositional (idiom)
Regular construction	Typical syntax (*the dog slept*)	*(wide) awake* *(heavy) smoker* *rain (cats and dogs)*	*kick the bucket* *green card* *in the light (of)* Fr. *pomme de terre* 'potato'
Sub-system	Dates: *5th of July* *tomorrow morning* Titles: *Miss Smith*	*Ludwig van Beethoven* in German : 'van' is a Dutch word similar to Ger. 'von'	*on top (of)* *in case (of)* Fr. *à côté (de)* 'next (to)' Meaningful dates: *September 11th* *4th of July* *Mount Rushmore* *Fort Alamo*
Irregular construction	Fr. *peser lourd* 'weigh a lot' lit. weigh heavy	Fr. *cucul la praline* 'very silly' lit. silly the praline	*each other* Fr. *quand même* 'anyway' lit. when same Fr. *à qui mieux mieux* 'each one more so than the other' lit. to whom better better

Table 4.1: Classifying MWEs according to two dimensions: degree of compositionality (columns) and degree of syntactic regularity (rows).

Table 4.1 opposes degrees of syntactic regularity in the rows and semantic compositionality in the columns.

In section 4.2.2, we will propose an annotation scheme for irregular constructions and for some non-compositional sub-systems. In the following, we first discuss the intricate problem of MWE and tokenization.

position.

[11] tagged as *SCONJ*

[12] Historically `parce` is the preposition `par` 'through' and the pronoun `ce` 'that', but this is not visible in today's orthography. The attribution of a POS to `parce` seems arbitrary and the French UD treebanks are subsequently incoherent: UD_French-GSD calls `parce` an ADV, UD_French-Sequoia an SCONJ, and UD_French-ParTUT has both versions.

4.2.1.1 MWE and tokenisation in UD

The tokenization of UD follows the underlying principle that tokens must be words or parts of words. A priori no token contains spaces (except well delimited cases of polysyllabic words) and therefore multi-word expressions are described syntactically and not morphologically. This is a vital choice for practical and theoretical reasons: Ambiguous sequences cannot be disambiguated on a morphological level without taking into account the whole sentence. Therefore, the alternative choice of multi-word tokens containing spaces is problematic. In the manual annotation process, creating the tokenization and the syntactic analysis at the same time is time-consuming, annotating a special link for MWE is much more user-friendly. For automatic parsing, too, a tokenization as a separate task that precedes the actual dependency annotation is redundant because both tools need a global view on the sentence and syntactic parsers are specialized tools to do just that. Moreover, two annotations of the same sentence are harder to compare if they are based on different tokenizations and a spelling-based annotation makes that possible because it does not depend on the possibly ambiguous syntactic annotation itself. Inversely, grouping Multi-Word Expressions together in a syntactic annotation scheme can always be achieved by introducing into the set of relations special ad hoc links for multi-words. UD makes use of this approach with the links *fixed* and *flat* [13] where no internal structure is annotated. In UD terms we could reformulate the purpose of this issue simply as: When must the *fixed* relation really be used?

The proposed work springs from the observation that the treatment of functional MWEs in UD is unsatisfactory for at least four reasons:

1) The relation *fixed* is commonly used for MWEs with a very clear internal syntactic structure (see Figure 4.5).[14]

[13] *flat* is a relation used for headless constructions (such as `Bill Clinton` for which it is not easy to decide which word is the head). This relation concerns productive and regular sub-systems and will not be discussed here.

[14] UD's definition of *fixed* refers to [Sag et al., 2002] who state: "Fixed expressions are fully lexicalized and undergo neither morphosyntactic variation (cf. **in shorter*) nor internal modification (cf. **in very short*). As such, a simple words-withspaces representation is sufficient. If we were to adopt a compositional account of fixed expressions, we would have to introduce a lexical entry for "words"

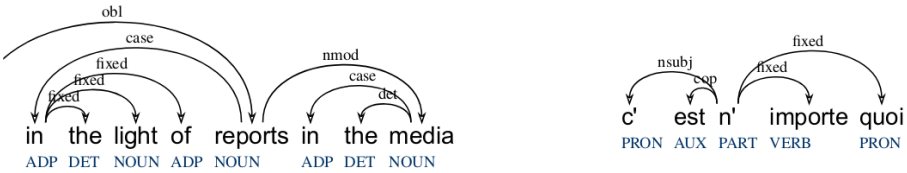


Figure 4.5: Analyses with UD *fixed* relations in UD_English-ParTUT and UD_French-GSD

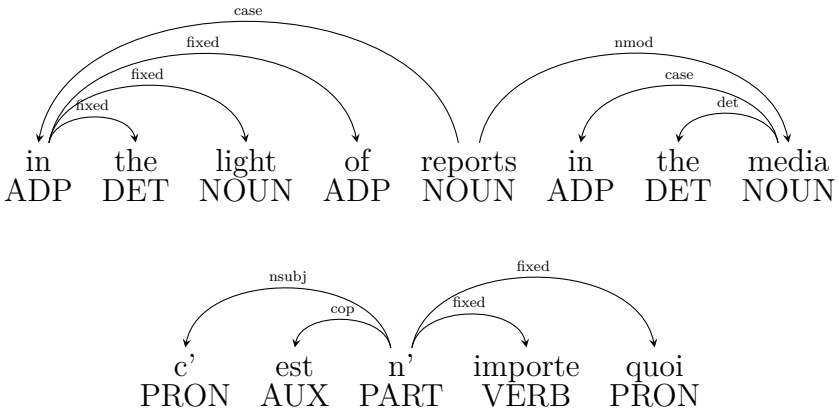


Figure 4.6: Analyses with UD *fixed* relations in UD_English-PartTUT and UD_French-GSD

When analyzing them as *fixed* MWEs, we flatten the structure, thus losing precious information in the process, which will give us fewer instances of these syntactic relations on which to train our parser (cf. the principles introduced by [Gerdes and Kahane, 2016] as well as the principles given on the UD introduction page). Moreover, the analysis is somewhat contradictory: If we recognize the POS of the components (such as the verbal nature of `importe` in Fr. `n'importe quoi` 'anything', lit. 'no matter what'), then we could also recognize the dependency relations between the tokens.

such as *hoc*, resulting in overgeneration and the idiomaticity problem (see above)." Let us remark that, first, limits on modification do not imply weird lexical entries, as the example *in short* shows itself – the two words being in the lexicon anyhow. Second, and most importantly, a MWE can have constraints on modification for a specific meaning while still remaining transparent for the speaker, not only diachronically: *in short*, for example, is identifiable as a prepositional phrase, even if 'short' is originally an adjective. This leads to multiple but syntactically constrained internal modifications of MWEs, not only in puns and journalistic style, but more generally also in ordinary coordinations and elisions as we will see below. Note also that in v2.0 of the UD_English-GSD treeebanks (the latest version at the time this work was conducted) `in short` (3 occurrences) and `for short` (1 occurrence) are consistently annotated as a compositional prepositional phrase (*case-nmod*), contrarily to Sag's paper referenced in the annotation guide.

2) For a long time, the criteria to decide which constructions enter the realm of MWEs were insufficient, and there were a lot of discrepancies between different treebanks and even inside a single treebank. For instance `along with` appeared with three analyses. In UD_English-ParTUT `along` was considered as the case marker of the noun phrase and `with` as `along`'s *fixed* dependent. On the other hand, En-Original mainly favoured a compositional analysis with both `along` and `with` as case markers, but there was also one occurrence where `along` was a *cc* dependent of the noun phrase and with `along`'s *fixed* dependent. The tables in Figure 4.7 give an overview of the usage of the MWE-relations in the English and French UD treebanks v2.0. When comparing the highlighted lines in the English and the French tables, we observe that the usage that annotators make of the three MWE relations *compound*, *fixed*, and *flat* go beyond what can be expected as language and genre differences. It rather seems to indicate that the annotators understood the relations differently. This is corroborated by the high inter-corpus variation, for French, too. The two French treebanks UD_French-FTB and UD_French-Sequoia, for example, do not use compound at all. The significant number of observed inconsistencies in these two languages suffices to show that the UD annotation guide for MWE relations clearly deserved an overhaul in order to achieve a higher inter-language, inter-corpus, and inter-annotator annotation.

3) The POS of an MWE as a whole does not appear explicitly. The assumption made is that the MWE will have the same POS as its syntactic head but many examples show that this is not the case. For example `not to mention` is a coordinating conjunction, a useful information for a syntactic parser that cannot be retrieved from the POS of its units.

4) The span of MWEs in this version of the UD scheme is questionable in some cases, especially concerning governed prepositions, which are not separated from the MWE itself (cf. `of` in Figure 4.8, below).[16]

[15]In the Figure, the suffix -Original refers to the actual -GSD treebanks, the treebanks were renamed from version 2.3 onwards to indicate their origin *Google Stanford Dependencies*.

[16]The preposition can be repeated (According to the President and to the Secretary of State – the repetition can disambiguate the scope of the shared element in the coordination) which seems

English	compound	fixed	flat
En-Original	**4,38 %**	**0,24 %**	**0,73 %**
En-Lines	**2,63 %**	**0,49 %**	**0,72 %**
En-ParTUT	**0,40 %**	**0,56 %**	**1,24 %**
total number of MWE	9194	966	1882
max freq variation between corpora	1107%	43%	59%
total nb links	11993	1091	2625
total frequency of links	3,58 %	0,33 %	0,66 %
total nb MWE types	7067	122	1215
average nb of occurrences per type of MWE	1,3	7,9	1,5
non-contiguous types	292	4	0

French	compound	fixed	flat
Fr-Original	**0,21 %**	**1,04 %**	**1,79 %**
Fr-FTB	**0,00 %**	**8,75 %**	**0,70 %**
Fr-ParTUT	**0,23 %**	**1,04 %**	**0,44 %**
Fr-Sequoia	**0,00 %**	**2,56 %**	**1,25 %**
total number of MWE	786	33190	9444
max freq variation between corpora	N/A	843%	411%
total nb links	877	55975	11858
total frequency of links	0,08 %	5,36 %	1,14 %
total nb MWE types	660	8544	7329
average nb of occurrences per type of MWE	1,2	3,9	1,3
non-contiguous types	24	58	0

Figure 4.7: Measures of MWEs of English and French UD [15]v2 illustrating important % differences in the usage of *compound*, *fixed* and *flat* across different treebanks.

4.2.2 Propositions for the encoding of MWEs in UD

All regular constructions from Table 4.1, including idioms, should be analyzed internally because:

1. Such a tree is syntactically more informative than any type of flattened structure where readily available syntactic relations have been removed.

2. We can expect a higher inter-annotator agreement on the syntactic relations if the annotation of MWE is kept independent from syntax, because of the difficulty of defining and recognizing MWEs.

3. Equally, we can expect better parsing results because we have more instances of every relation and unknown idioms can obtain a correct parse, too. The same holds for all compositional and semi-compositional constructions.

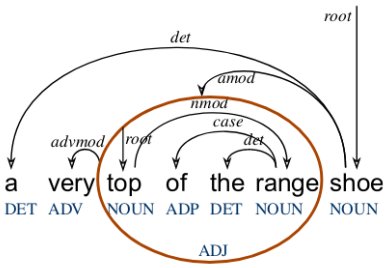
incompatible with the fixed analysis favored in the English treebanks. In other languages, such as French, the repetition is quite systematic. In English, governed prepositions are particularly cohesive with their governor, giving us what is called preposition stranding in extraction (`the girl I talk to`). But even in this case, nobody denies that the verb `talk` subcategorizes a preposition phrase and that the preposition `to` is not part of the verb form. The fact that the preposition is not a part of the idiom becomes even clearer with expressions such as `in front of X`, where the subcategorized phrase can be suppressed (`she stopped in front`) or pronominalized (`in its front`). Note that the alternative classical dependency analysis where prepositional phrases are governed by prepositions results in a more coherent analysis because the governor (the verb or the expression) always forms a subtree with the sub-categorized preposition, independently of the extension given to the MWE.

We even go as far as proposing to analyze non-productive irregular constructions in cases **a)** and **b)** (as defined in subsection 4.2.1 above) by regular syntactic relations, but for some MWEs, we need means of encoding the POS of the whole expression because its POS is not identical to its head's POS. We propose to use *fixed* only for parts of **c)** and **d)** where the regular syntax does not provide appropriate syntactic relations. In some MWE of **c)** and **d)**, some relations remain transparent and we could annotate partial structures whenever they are available. For example `à qui mieux mieux` contains a clear `à` <[*case*]- `qui` relation independent of the analysis of the rest of the expression.

For those remaining *fixed* relations, dependency distance measures would give more reliable result if the standard 'bouquet' annotation (where all words are dependents on the first token) would be replaced by a series of left-to-right relations connecting one word to its neighbour, because the absence of any recognisable syntactic relation rather implies some relation of simple juxtaposition than a structure headed by the first word. The CoNLL-U format can easily be extended to allow for a fully expressive annotation of MWEs. One solution is to devote one specific column holding the idiomatic information (or equally, put this information into a specific attribute in the feature column of CoNNL-U). This choice does not allow embedding MWEs in one another. A better choice is to extend the multi-word token format by adding a line for each MWE. This additional line could also include the POS of the whole expression.[17]

It constitutes an additional unit that can constitute a node of a semantic graph. This could be combined with a specific MWE column or simply a specific feature in the additional line's FEATS column that distinguishes different types of non-compositionality, following the PARSEME project: for instance idioms, light-verb constructions, and named entities. In the following example, the governor of the MWE `top of the range` is `shoe`. But the head/root of the MWE is `top`. UD presents a particular problem with functional MWEs, because UD favors dependencies between content words (determiners and prepositions are dependents of the noun following

[17]Currently the format is only used for contiguous items. The format can be extended to non-contiguous expressions, e.g. we could have "3-5,7-8" as an index.



1	a	a	DET	_	_	7	det
2	very	very	ADV	_	_	**3-6**	**advmod**
3-6	**top of the range**	_	**ADJ**	_	_	**7**	**amod**
3	top	top	NOUN	_	_	**3-6**	***root***
4	of	of	ADP	_	_	6	case
5	the	the	DET	_	_	6	det
6	range	range	NOUN	_	_	3	nmod
7	shoe	shoe	NOUN	_	_	0	root

Figure 4.8: Left: UD analysis of the adjective `top of the range` - case **a)**. Right: Proposed encoding of Functional MWEs in the CoNLLU format.

them). It appears that the choice made by UD to have the prepositions as dependent of their complement is the source of some "catastrophes" (in the mathematical sense of the term) as soon as prepositional MWEs are involved [Gerdes and Kahane, 2016].

The goal of this section is to present the problem and to propose a solution to smooth it. Let us consider the following examples illustrating what is often called a **complex determiner (1a)** and a **complex preposition (1b)**:

1a: `She asked me` **a lot of** `questions.`

1b: `She lives` **in front of** `my house.`

We can compare these sentences with (2a) and (2b):

2a: `She asked me` **many** `questions.`

2b: `She lives` **near** `my house.`

According to the choices made by UD, we have dependencies between `asked` and `questions` in (2a) and between `lives` and `house` in (2b) (Figure 4.9)

It is tempting to preserve these dependencies and to treat `a lot of` and `in front of` respectively as a complex determiner and a complex preposition. Let us first remark that `of` in these expressions is not part of the MWE, but is part of the subcategorization of the MWE, by parallelism with verbal subcategorization. In other words, the MWEs in question are `a lot` and `in front`. Theses MWEs are syntactically transparent and we do not want to analyze them as *fixed*.

Two analyses are possible:

Analysis A (Fig. 4.10) respects the surface syntax and `of N` is treated as the complement (*nmod*) of the MWE. This is the most common analysis in the v2.0 English

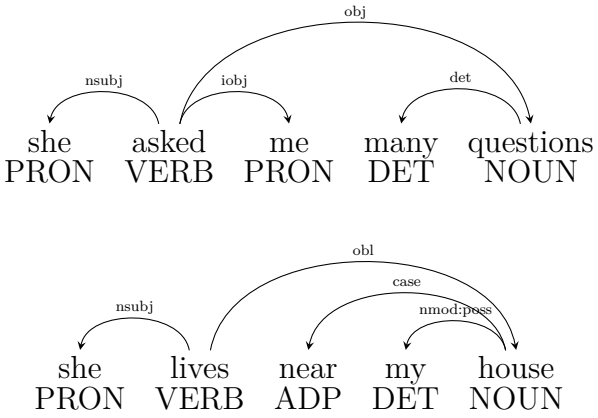


Figure 4.9: UD analysis of sentences 2a and 2b.

UD treebanks.[18]

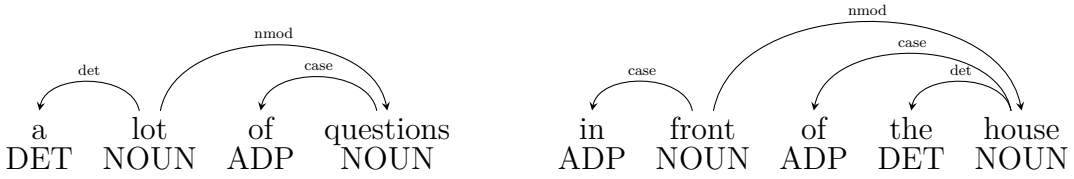


Figure 4.10: Analysis A for *a lot (of)* and *in front (of)*

Analysis B (Fig 4.12) favours the relation between content words, as in the analyses of Figure 4.9. In this analysis, we propose to introduce special relations *det:complex* and *case:complex* when the dependents of *det* and *case* are MWEs.

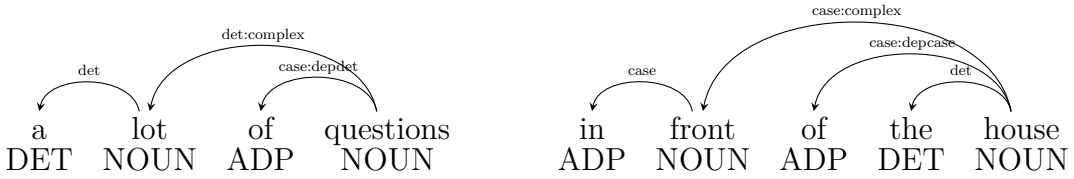


Figure 4.11: Analysis B for *a lot (of)* and *in front (of)*

The sub-categorized preposition `of` is governed by the complement noun. We intro-

[18]Since `quite a lot` (of questions) is possible, `a lot` has actually become an adverb (just like in `a lot better` – or other comparative adjectives) and the relation between `a lot` and the noun complement of questions should be of type *obl* and not *nmod*. This irregular behaviour of `a lot` can be captured by the introduction of an MWE unit as in Fig 4.8.

duce a feature on the *case* relation to indicate that this preposition is subcategorized by a dependent of the noun. We need to distinguish *case:depdet* and *case:depcase* because both can be present: `in front of a lot of houses`, where `front, lot` and the two `of` will depend on `houses`.

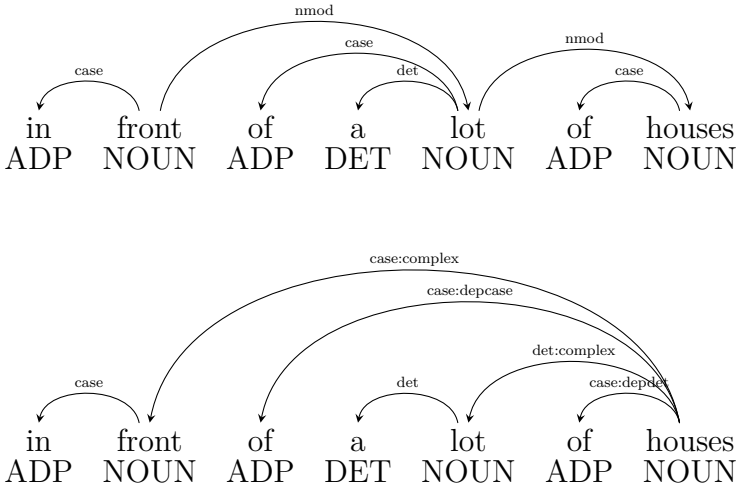


Figure 4.12: Analyses A and B for *in front of a lot of houses*

In [Kahane et al., 2017a], we also proposed a set of rules based on these analyses to convert one structure into the other. While analysis B does preserve relations between content words to a greater extent than analysis A, analysis A has the advantage that it is already in use and doesn't require the introduction of specific relation subtypes.

4.2.3 Conclusion

We have shown that irregular structures need to be introduced as units because we have to associate a POS to them. In cases **a)** and **b)** dealing with clear or rather clear POS and dependency structures the internal structure is transparent, however the POS of the complete unit is not predictable. In cases **c)** and **d)**, where we use *fixed* relations, it is all the more necessary to indicate the POS of the MWE. To this end the *ExtPos* feature (for *External Part-of-speech*) has been introduced and some of the UD treebanks are now adding this information. For regular idioms, too, we can add the MWE as a unit. For regular functional MWEs, we propose to add sub-types

to the relation to capture the relations between content words, as well as the syntactic dominance relations. A tree does not allow expressing both types of relations at the same time, but the proposed sub-types relations can be converted from one to another.

The proposed schemes and distinctions clarify some underspecifications in the current UD scheme that lead to incoherent analyses. The usage of subtypes fits in unintrusively into the current scheme and could be used for upcoming versions.

4.3 Case Study n°2 : Naija

In the following section, we present our work concerning annotation guidelines and treebank development focusing on a low-resource language, the English pidgin-creole Naija, spoken by millions of people in Nigeria. This work was carried out in the framework of the ANR NaijaSynCor project [Caron, 2017]. Figure 4.13 gives a schematic overview of the project, the primary aim of which is to deal with sociolinguistic analyses. To this end, a large speech corpus had to be collected and the necessary linguistic annotations had to be created. Our contributions to this project focused on linguistic,

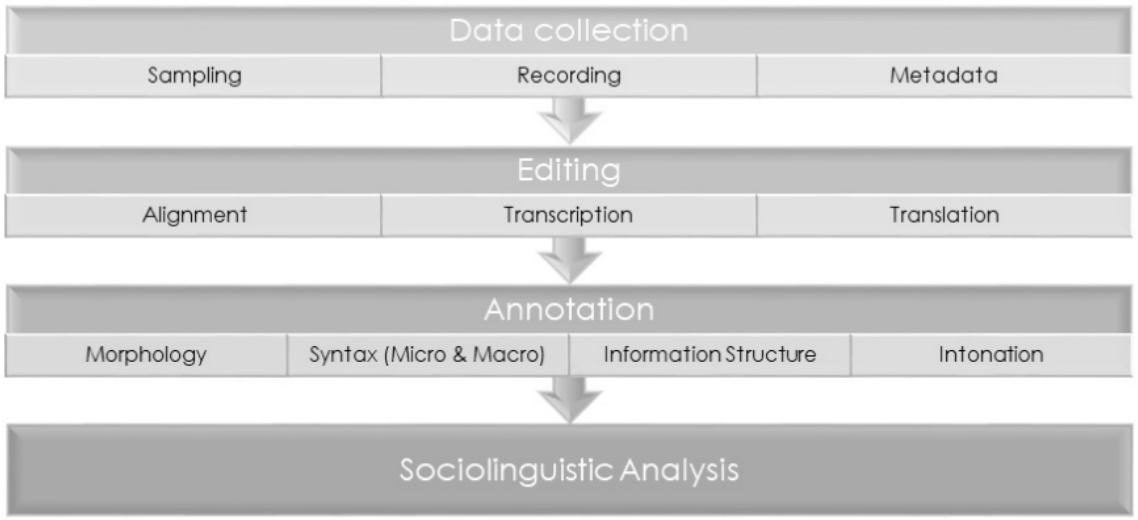


Figure 4.13: Workflow of the NaijaSynCor project.

treebank-oriented annotations and analyses, which will be described in more detail below.

First, the linguistic context of Naija is introduced in section 4.3.1. Section 4.3.2 concerns the corpus itself (metadata, transcription, translation, and glossing). The particularities of the morphosyntactic annotation, due to the fact that Naija is an English lexifier pidgincreole, are described. We present the theoretical choice of our segmentation into maximal syntactic units for this spoken corpus using/adapting the SUD annotation scheme followed by a quantitative evaluation of annotation agreement and accuracy.

The syntactic annotation is developed in the Surface-Syntactic Universal Dependency annotation scheme (SUD) [Gerdes et al., 2018, 2019a] and automatically converted into UD using a set of Grew rewriting rules [Guillaume et al., 2012b]. The choice of the SUD annotation scheme will be further explained in 4.3.2.5.

4.3.1 Linguistic context

Naija is an English pidgincreole[19] [Bakker, 2008, p. 131] spoken by an estimated 100 million speakers both in Nigeria and by the Nigerian diaspora in Africa, the UK and the USA.

It originates from Nigerian Pidgin, a creole spoken in the Niger delta [Faraclas, 1989; Elugbe and Omamor, 1991]. Nigeria is home to more than 500 languages, representing three genetic phyla (Niger-Congo for Yoruba, Afroasiatic for Hausa or Nilo-Saharan for Kanuri to name just a few). In this context, Nigeria Pidgin serves as a popular lingua franca. Since the national independance (1960), the creole has spread from its original niche in the Niger delta to cover a large area reaching up to Kaduna and Jos (see Fig. 4.14 for a map of Nigeria), and acquired new functions along the way. Although it has no official status or standard orthography, it has been adopted for private and informal communication by the educated youth and the Nigerian elite.

Initiatives such as the Wazobia radio and TV network founded in 2007, or the Lagos-based “Pidgin”[20] BBC station opened in 2017, use Naija as their only medium.

[19] As defined by Bakker : "A pidgincreole is a former pidgin that has become the main language of a speech community and/or a mother tongue for some of its speakers."

[20] “Pidgin” is the name commonly used by locals to name what we refer to as “Naija”.

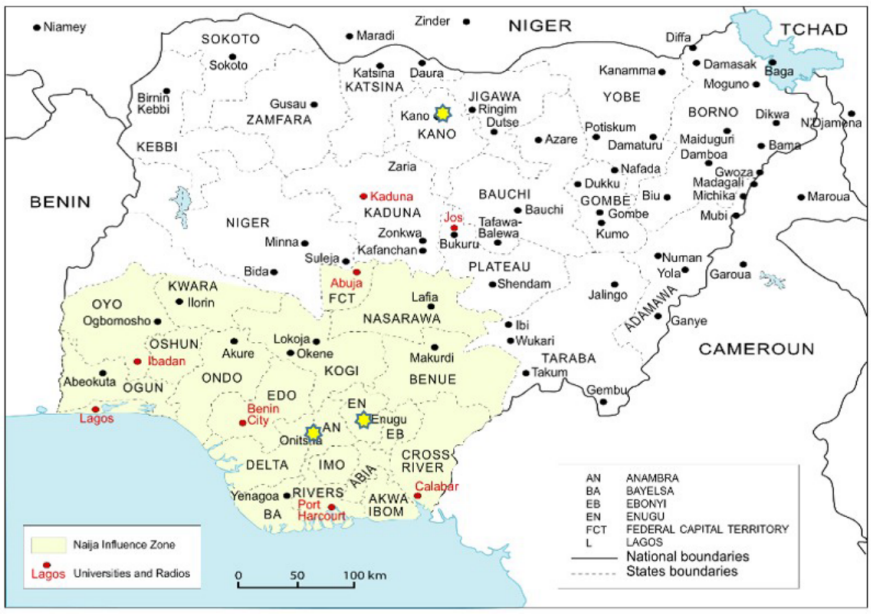


Figure 1. Map of the 11 survey locations

Figure 4.14: Map of the 11 survey locations.

In the process, Naija has developed new structures, a new vocabulary, and probably a new prosody, that differentiate it from Nigerian Pidgin.

Despite the growing importance of the language in Nigeria, and the country's place as the world's sixth-most populous country, little attention has been paid to Naija, and most of the literature surrounding the language is based on outdated descriptions of Nigerian Pidgin. It was in this context that the NaijaSynCor project emerged [Caron, 2017], with the goal of creating a corpus-based survey of Naija.

In the following section, we address the syntactic annotation of the corpus and the constitution of a 150,000 words gold standard treebank.

4.3.2 Treebank development

4.3.2.1 Corpus Metadata

.

The theoretical framework selected for the analysis of the corpus was variationist sociolinguistics [Tagliamonte, 2012] which meant that the samples had to best represent

different types of speakers, and different types of functions. A questionnaire was administered to record the relevant metadata about the speakers: time, place and conditions of recording; sex, age, education, professional activity, geographic origin, linguistic background and history. A total of 319 speakers were recorded, covering 11 locations (see Fig. 4.14). The genres recorded cover life stories, speeches, radio programs, free conversations, cooking recipes, comments on current state of affairs, etc. In [Caron, 2020], it is pointed out that the sampling of the corpus was not as balanced as initially hoped for. While the geographic sampling is satisfactory, there is an imbalance regarding speakers' age, sex, and their level of education, with highly educated men in the 16 to 45 age brackets being the most common speakers.

4.3.2.2 Transcription

Naija is commonly written, in particular on the internet, on forums, but also for example on the BBC website. Although an official orthography or normalization has not taken place, Naija speakers have strong opinions on how most words should be spelled, and we decided to follow these evolving conventions. Mostly, the speakers prefer etymological orthography (i.e. inspired by the Standard English) modified for some emblematic Naija words for which specific spellings have developed, e.g. `wetin` 'what', `moda` 'mother', `fada` 'father', `dem` 'they', 'them' or used as a plural marker. We have used a specific orthography to disambiguate certain function words, e.g. de (a variant of `dem`) vs. `dey` (the imperfective auxiliary[21]); `come` 'to come' and `con` (the consecutive auxiliary[22]), `say` 'to say', and `sey` (the reported speech complementizer[23]).

This usage-based orthography was used to to avoid promoting an artificially authoritative norm. While this transcription methodology is interesting to see what tendencies emerge in the transcription, it certainly poses a challenge to train NLP tools on the data. To counteract this, we introduced a common lemma for all variants.

[21] used to indicate ongoing processes

[22] used to indicate that something occurred directly following a previously-referenced event e.g `I con come Port Harcourt for twenty fourteen` which means 'Then I came to Port Harcourt in 2014'.

[23] a marker used to introduced reported speech as in `I tink sey "ah at least I no go suffer"` translated as 'I thought, "ah, at least I won't suffer anymore"'

For example the word for 'thing', was transcribed as either `ting`, `tin` or `thing` by the annotators. These variants are associated to a common lemma 'fing' for better generalisation.

The translation of all the sentences into English has been done by a team of native speakers of Naija. It aims at remaining as faithful as possible to the structure and style of the original oral data, keeping the hesitations, repetitions, and general disfluencies.

4.3.2.3 Morphosyntactic analysis

Two distinct annotation phases The morphosyntactic and dependency analyses were done in two distinct phases. In the first phase, an initial treebank of 13000 tokens was annotated by three annotators, the lead annotator Bernard Caron, a researcher from LLACAN (Langages, Langues et Culture d'Afrique), was familiar with Naija and its grammar, having lived many years in Nigeria. The other two, Sandra Bellato and I, were not familiar with Naija but had been trained in dependency annotation, and familiar with the Universal Dependency scheme and its application to spoken French, and written English. The sentences were glossed and translated in English, so we could get an (imperfect) understanding of the structure. During this initial phase, there was a lot of interaction between the three of us, as Sandra and I annotated examples and directed our questions to Bernard, who would answer them or ask Nigerian colleagues for feedback when needed.

On the other hand, Bernard would sometimes describe structures to us and ask what we thought the appropriate way to describe them would be in UD. It was during this phase that we developed the initial guidelines. Around the time that Surface-Syntactic Universal Dependencies [Gerdes et al., 2018] was introduced, it was decided that the treebank would continue to be developped in SUD as a proof of concept, and to facilitate the onboarding of new annotators who were more familiar with a surface syntax approach. A conversion grammar was written in the form of Grew rules [Guillaume et al., 2012b], and tested on the treebank. Once it was sufficiently refined, this converted and corrected SUD version became the native version of the treebank.

Thus, at the end of this first phase, we had a small treebank annotated in both UD and SUD, thoroughly checked both manually and more automatically for inconsistencies and disagreements. We also had two conversion grammars, one to transform UD into SUD and one to come back to UD from SUD, as well as guidelines describing the annotation for both schemes that could be used by other annotators. The second phase of annotation then started, featuring three Naija speakers, Chika Kennedy Ajede, Emeka Onwuegbuzia, and Samson Tella, and with Bernard Caron still supervising. During this annotation phase, there was only one annotator assigned to each file[24]. Together, they annotated another 127 000 tokens, to arrive at the 140 000 token NaijaSynCor treebank.

Next, we will go into more detail about how exactly the annotation was done.

Glossing and POS tagging. The initial annotation of part-of-speech tags was not done from scratch, but facilitated by pre-tagging done with a model trained on English. To transfer the model and apply it to Naija, a very minimalist gloss was used, with Naija lexical innovations translated into English (e.g. `pikin` 'child', `patapata` 'full'), function words glossed by their morphological features, and lexical items borrowed from English kept as they were (*do*, *nineteen*...), module the orthographic variation. The POS annotation was then manually corrected and a first dictionary of the function words and most common lexical items of Naija was created, containing the form, some orthographic variants, the POS tag, and an English gloss if necessary. This dictionary was then used on a dozen text samples inside the Elan-CorpA tool [Christian Chanard], an extended version of the Elan tool[25] [Sloetjes and Wittenburg, 2008], which proposes the dictionary's POS for each token for validation by the annotator. Through this semi-automatic process, the dictionary was enriched and later on used by the automatic tagger that was developed for the project[26].

The POS tags mostly follow the UD conventions with some changes made to ac-

[24]With the exception of a small sample annotated by the three of them, on which we measured inter-annotator agreement, see a few paragraphs below.

[25]https://tla.mpi.nl/tools/tla-tools/elan/

[26]The POS tags where provided by a model of the Mate parser [Bohnet, 2010], other morpho-syntactic features were added by means of a Wapiti-based CRF tagger [Tellier et al., 2010].

commodate for the specificities of Naija. For example, Naija has three copulas, among which two are tagged as VERB `be` and `dey` 'be', and one is tagged as PART `na` 'it is')[27]. The tagger was regularly trained to incorporate more manually corrected samples and improve its analysis of Naija, in a bootstrapping loop.

Annotation guidelines. Most of the development of the annotation guidelines was done during phase 1 of the annotation process. It was tested and further refined in phase 2 when three new annotators, all speakers of Naija were tasked with annotating another portion of the corpus. These guidelines were then given to the three Nigerian annotators, who were allowed to discuss difficult cases among them. At the end of this process, the annotation was consolidated through the use of a dictionary that was controlled independently and applied to the corpus. The final adjudication was done by an expert adjudicator on every single file. In this process some amendments had to be discussed more widely in the SUD community. The annotators were asked to report into the guide any decision that was not covered by it, so it could be refined.

4.3.2.4 Polycategoriality and polyfunctionality

Following the UD guidelines[28], the morphological specification of a (syntactic) word in the UD scheme consists of three levels of representation: a lemma, a POS tag, and a set of features representing lexical and grammatical properties. In order to reduce polycategoriality and its consequent multiplication of syntactic words, the annotation process has been guided by the principle of separation of the morphological tagging of a word from its syntactic function: A single lexeme can be polyfunctional, but it cannot be polycategorial. This principle applies in all languages, e.g. to adpositions (ADP) which can take a nominal, clausal or zero complement without changing their abstract lexical category [Huddleston and Pullum, 2005].

This principle has been applied to adjectives in Naija, which can function as predicates without any copula, (4.15) or noun modifiers (4.16). In both cases, the words

[27]This is true in the SUD version, but for the UD version, the copulas are converted into AUX in accordance with the UD guidelines.

[28]https://universaldependencies.org/u/overview/morphology.html

keep their morphological assignment:

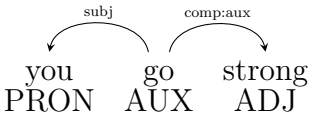


Figure 4.15: Analysis for a predicating adjective in 'You will be strong' [PRT_05_Ghetto-life_P_24]

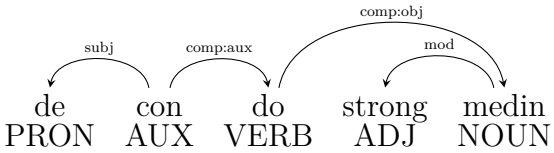


Figure 4.16: Analysis for an adjective modifying a noun in 'then they did strong magic' [IBA_04_Alaska-Pepe_P_95]

However, some lexical words are grammaticalized into new function words. An example is given by the numeral `one`, tagged NUM (4.17), which has grammaticalized into the determiner `one` 'some', 'a certain' (4.18), tagged DET, and the pronoun `one`, tagged PRON (4.19).

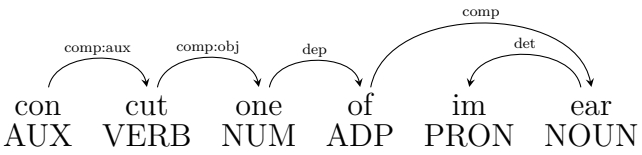


Figure 4.17: Analysis of *one* as a numeral in 'he then cut one of its ears.' [IBA_04_Alaska-Pepe_P_5]

4.3.2.5 Surface-Syntactic UD annotation

We described earlier that there were two distinct phases in the annotation process, and that while the original annotation was done in UD, we later shifted to Surface-Syntactic UD (SUD) [Gerdes et al., 2018]. Two different strands of thought, one rather practical, the other more theoretical, led us to annotate the corpus not in the standard UD dependency annotation scheme but rather in SUD . Firstly, the Nigerian annotators were trained in a standard syntactic X-bar sentence structure, where, for

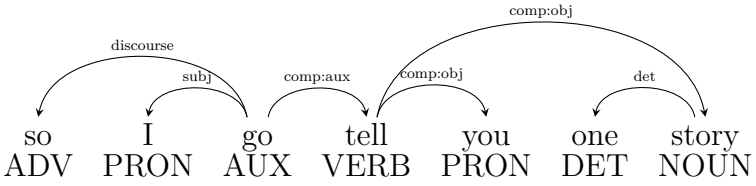


Figure 4.18: Analysis of *one* as a determiner in `so I go tell you one story` 'so I will tell you a story.' [IBA_04_Alaska-Pepe_P_5]

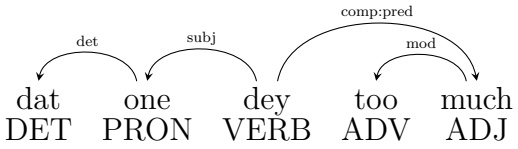


Figure 4.19: Analysis of *one* as a pronoun in `dat one dey too much` 'that one is too much.' [ABJ_INF_08_Impatience_106]

example, a PP is headed by a preposition [Osborne and Gerdes, 2019a]. In this context, SUD was much easier to acquire than UD dependencies [Gerdes et al., 2019a].

Secondly, the NaijaSynCor project has a central typological component, and language comparisons should be possible, based on syntactic differences, which is easier in a scheme based purely on distributional criteria, such as SUD, rather than on the semantic function word vs content word distinction which constitutes the basis of UD. We can add that UD is particularly problematic for multi-word expressions (MWEs) working as functional items (complex adpositions or complex conjunctions), especially when they are syntactically quite regular [Kahane et al., 2017a]. In SUD, MWEs such as the Naija adposition `base on` 'based on'[29] are connected, and in the dependency tree they occupy the same syntactic position as a simple word, see Fig 4.20 will the MWE highlighted in green.

The Naija treebank uses the SUD version proposed in [Gerdes et al., 2018], which can be automatically converted into UD. We will not go over all of the differences

[29]it is not a passive construction in Naija as there is no morphological passive.

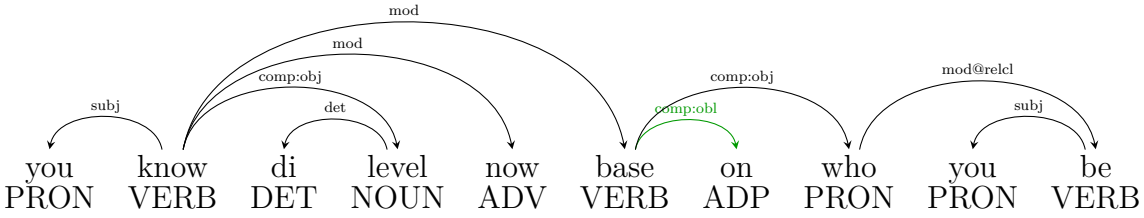


Figure 4.20: Analysis for the sentence `you know di level now base on who you be` 'You have an idea of the level, based on who you are.' [WAZK_08_Fuel-Price-Increase_MG___23]

between UD and SUD and instead refer to the two publications [Gerdes et al., 2018, 2019a], however we would like to explain how the principles behind SUD are in accordance with some of the recommendations

One of the principles behind SUD is to simplify the annotation, by reducing some of the redundancy present in UD. Thus the category of a dependant (whether it is nominal, adjectival...) does not influence the attribution of its function unlike in UD where some labels are specific to certain category of dependants (*nmod* for nominal modifiers, *amod* for adjectival modifier, and so on). In SUD however, all modifiers are *mod* as in Fig. 4.21, whether they're nouns (`mama`) or adjectives (`younger`). This also applies to clauses such as adverbial clauses which work as modifiers, as in the example `after you parboil am you go come pound am` 'After you've parboiled it, then you'll pound it.', where the underlined segment corresponds to a modifier, adverbial in nature but labelled as *mod*.

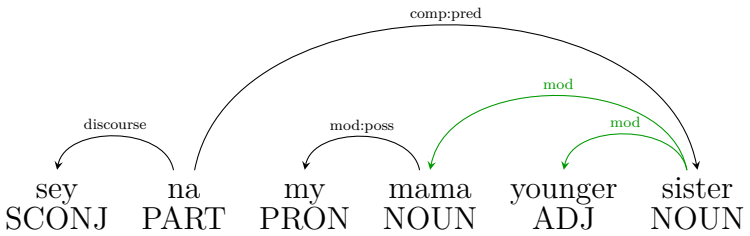


Figure 4.21: Analysis for the sentence `sey na my mama younger sister` 'I said she's my mom's younger sister.' [IBA_01_Fola-Lifestory_MG___61]

This isn't limited to modifiers any two elements that occupy the same syntactic position are linked to their governor by the same relation. Another instance, in English

this time, would be *the problem* and *you're wrong* both being labelled as *comp:obj* in *I know the problem* and *I know you're wrong*, while the first is *obj* and the second is *ccomp* in UD.

The SUD relation tagset is structured in a taxonomy with a main relation, optionally followed by a subrelation and/or a deep syntactic feature. The structure of the relation tagsets helps to limit what [Fort et al., 2012] call "the degree of freedom at any point of choice", which reduces the complexity of annotation.

For verb complements in Naija, we use the following sub-relations:

- *comp:obj*, for direct objects;
- *comp:obl*, for oblique complements such as for my family in `na me be di last born for my family` 'I am the last child in my family.';
- *comp:pred*, for predicative complements such as `someting` in `road no be someting wey I dey like like dat` 'I don't like being on the road like that.'.
- *comp:aux*, for relations between a TAM (Tense—Aspect—Mood) auxiliary and the full verb.
- *compound:svc* which we use for serial verb constructions, which are typical for Naija (see Section 4.3.3.3).

The difference of serial verb construction annotation between UD and SUD is exemplified in Figure 4.22.

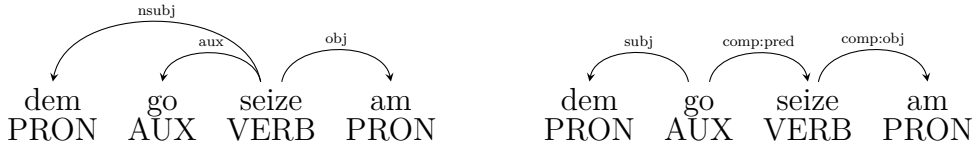


Figure 4.22: UD and SUD analyses for the sentence `dem go seize am` 'They will seize it.' [DEU_C01_D_6]

Accessing the treebank. The treebank is accessible in its native version on the Surface Syntactic Universal Dependencies website and can be queried directly using the Grew-Match Interface. It is also automatically converted into a UD annotation scheme[30].

4.3.2.6 Evaluation

Inter-annotator agreement. Once the guidelines had been solidified, and the annotators from phase 2 were familiarized with the annotation task, we decided to assess the stability of the annotation. We verified the inter-annotator agreement on three samples composed altogether of 121 sentences. These samples were pre-tagged and parsed (which will have important implications for the evaluation of the inter-annotator agreement). For this experiment, the annotation was done completely independently by our three annotators, with no communication among them.

This allows us to compare the inter-annotator agreement based on the pre-parsed structure and measure the difference on the tags and relations that have to be changed to obtain the gold annotation. The results of this experiments are presented in Fig 4.2 (originally shown in [Caron et al., 2019]).

We report the percentage of agreement on UPOS (the POS), UAS (Unlabelled Attachment Score, i.e for each token, do the annotators select the same governor) and LAS (Labelled Attachment Score, i.e for each token, do the annotators select the same governor and same function label, a partial disagreement is enough to be considered as a disagreement here). Because the pre-tagging and pre-parsing introduce a bias, we decided to also specifically measure this agreement on areas where at least one of the annotators deviated from the pre-parsed annotation.

While the agreement looks satisfying at first glance. For the LAS in particular,

[30]Initially, the treebank was developed in UD (version 2.2 to 2.5), it is only from version 2.6 and onward that the UD annotations are automatically converted from SUD.

[31]We look at agreement between pairs of annotators, A/B means we are looking at the agreement between the annotator A and annotator B

	Percentage of agreement			Percentage of agreement when the annotation differs from the preannotation		
	A/B	B/C	A/C	A/B	B/C	A/C
UPOS	95	94	95	46	41	37
UAS	93	91	91	68	60	58
LAS	89	86	87	60	51	50

Table 4.2: Inter-annotator agreement scores and the effect of pre-parsing. [31]

86-89 % of agreement is pretty impressive for a spoken treebank, in a language where there is no history of dependency annotation and therefore no conventions to rely on, and given the type of text which is quite complex. However this is nuanced by the fact that there is a pre-annotation, and that the agreement drops significantly in places where one of the annotators deviates from the pre-annotation (which could be due to the fact that the others simply "accepted" the pre-annotation at face value, whereas they would have reached a different analysis on their own).

Further experiments would have to be made to explore which structures in particular were more prone to disagreements, and how to limit the pre-annotation bias.

Evaluation of treebank coherence and the impact of macrosyntactic annotation It is no easy task to evaluate the quality of a treebank. When evaluating parsers, there are two main paradigms : intrinsic evaluation (comparison to a ground truth) and extrinsic evaluation (looking at the relative utility of the syntactic representations for different downstream applications often in the form of semantic tasks). For the annotation itself however, these evaluations paradigms don't apply as easily, especially in a low resource setting.

Here we propose to use a parser as a means of evaluating the coherence of the annotation, under the assumption that a more coherent annotation is easier to learn than an incoherent one. We also want to observe how degraded the parsing performance are if we remove the macro-syntactic markup (this markup is a kind of formalized punctuation based on prosody, developed in the Rhapsodie project of annotation of spoken French [Deulofeu et al., 2010; Pietrandrea and Kahane, 2019]). For examples of how it is used in Naija see [Caron et al., 2019].

We expect this macrosyntactic annotation to have a positive influence on dependency parsing, in particular for constructions such as coordination and dislocation, which are associated with specific macrosyntactic markups resulting in specific dependency relations. To verify this claim, we trained the Mate tagger and parser [Bohnet, 2010], first on a version of the treebank including these markups, and then on a version of the treebank where they have been removed except for "//" (the segmentation into illocutionary units which roughly correspond to sentences).

This experiment also provides a baseline for the quality of the annotation we can expect on the the rest of the NSC corpus (which is also transcribed and macrosyntactically annotated but will not be manually annotated) as well as for parsing other spoken and written Naija data (whether it be with or without markup) in the future.

At the time of the experiment, we used all the data which had been validated, which represented 52k words split in two, with 90% of the data for training and 10% for the evaluation. The results are presented in Figure 4.3[32].

While the POS tagging scores are as expected very similar whether macrosyntactic annotation is present or not, we obtain a noticeable LAS error reduction of 11% and a UAS error reduction of 18% when the treebank does include the macrosyntactic markup.

	Macro-syntax +	Macro-syntax -	Error reduction
UPOS	92.44	92.23	*
UAS	90.76	89.23	18%
LAS	84.45	82.02	11%

Table 4.3: Parsing results with and without macro-syntactic annotation - LAS (Labeled Attachment Score) and UAS (Unlabeled Attachment Score).

We also looked at the syntactic functions which most benefitted from the inclusions of the macrosyntactic markup, and to not surprise they included piles (coordinations, reformulations and reduplications). We also observe an improvement for relations that

[32] *punct* relations are excluded from the evaluation as they are exclusively used for macro-syntactic markers.

connect a nucleus and *adnuclei*, such as clefts, dislocations, and peripheric modifiers.

The parser scores on the whole seem promising, in particular for spoken texts. These metrics, were later on improved upon by [Guiller, 2020], who uses contextual embeddings adapted from English to Naija as an additional input for parsing. He also provides an error-analysis of the parsing results, which suggests potential avenues of improvement for the annotation.

Although modest in size, this treebank provides invaluable data to learn and evaluate a Naija parser. The UD version was used as evaluation data in the CoNLL 2018 Shared Task (Multilingual Parsing from Raw Text to Universal Dependencies) [Zeman et al., 2018] where participants had to learn to parse corpora without annotation and were evaluated on 57 languages, one of them being Naija.[33]

We hope to further improve the treebank coherence through an ongoing process of semi-automatic rule-based enhancement. In particular, we address this problem of annotation inconsistencies using a systematic comparison of parsing results with the gold annotation and the double SUD-UD-SUD conversion, and by different error mining tools such as the relation table proposed by the Match-Grew tool available on match.grew.fr [Bonfante et al., 2018], which shows the number of dependency relation types between any pair of categories.

4.3.3 Some idiosyncratic syntactic constructions of Naija

In this section we present three interesting constructions of Naija that show clear structural differences with English, Naija's lexifying language. These structures show that while transferring parsing models from English can work to a certain extent, other structures that are more specific to Naija would not be accurately analysed.

4.3.3.1 Clefts in Naija

Clefts are defined by [Lambrecht, 2001] as a complex sentence structure (let's take for example `It was my friend who ate the lemon cake.` made of two elements: a

[33]The test dataset being the UD 2.2 version of our treebank.

matrix clause headed by a copula (`It was my friend`) and a relative or relative-like clause whose relativized argument (which we will underline) is co-indexed (i.e they share the same referent) with the predicative argument of the copula (`who ate the lemon cake`). The cleft expresses a simple proposition which can also be expressed in the form of a single clause (`My friend ate the lemon cake.`) without a change in truth conditions.

Surface-syntax UD nicely captures the complexity of clefts in Naija, and the way in which they contrast with modifying relative clauses.

As shown in [Caron, 2021], there are three types of clefts in Naija, **wey**-clefts, **bare**-clefts and **double**-clefts. They are exemplified in Figure 4.23.

1b'	***wey*-cleft**	***na** weekend **wey** we dey do am*	'It's in the weekend that we do it.'
1b"	**bare** cleft	***na** weekend **Ø** we dey do am*	
1b"'	**double** cleft	***na** weekend **na im** we dey do am*	

Figure 4.23: The three structures of Naija clefts

We will describe what the structure of **wey**-clefts looks like, using the example `Na nineteen eighty four wey de born me.` 'It's in nineteen eighty-four that I was born'. The full dependency tree is provided in Figure 4.24, with dependency relations highlighted when they introduce a cleft complement. In Naija, clefts (regardless of their type) use the copula *na*,[34] that has two complements: First, a predicative complement, linked by the relation *comp:pred* (`na - nineteen eighty-four`, which corresponds to the matrix clause according to Lambrecht's definition) and second, a clause introduced by *wey*, a subordinating conjunction similar to *that* in English, that we link to *na* with the *comp:cleft* relation (`wey de born me`, which corresponds to the relative-like clause of Lambrecht).

It seems that, like in English, cleft sentences in Naija are, on the surface, similar to copular predications where a relative clause modifies the predicative complement. In Figure 4.25, the sentence `Na di ting wey Buhari meet.` 'This is the thing the Buhari found' corresponds to this construction.

[34]we tag as a particle rather than a verb or auxiliary because it cannot be negated or combined with TAM markers, two of the defining features of (auxiliary) verbs.

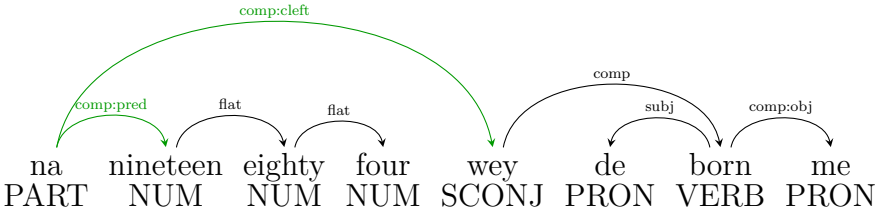


Figure 4.24: Analysis of the wey-cleft `na nineteen eighty four wey de born me` 'It's in nineteen eighty-four that I was born.' [KAD_09_Kabir-Gymnasium_P_6]]

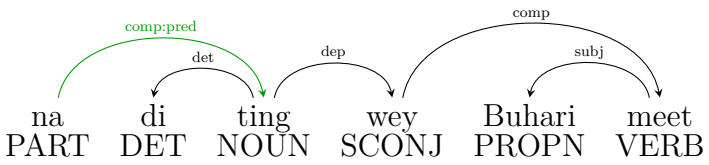


Figure 4.25: Analysis of the copular predication in *na* in `na di ting wey Buhari meet` 'This is the thing that Buhari found.' [IBA_25_Buying-Indomi_159]

The main difference is where the relative clause (`wey Buhari meet`) attaches, compared to the relative-like second complement in clefts (`wey de born me`) in Figure 4.24. In clefts, the relation between the predicative complement and the cleft clause is mediated by the copula, and the cleft clause is not dependent on the predicative complement but is raised and attached to the copula; whereas in copular predications the relative clause is governed by the antecedent (`ting` here), therefore the copula takes only one complement.

4.3.3.2 Interrogatives

In the NSC corpus, content questions are analyzed as clefts. This is corroborated by examples where the question word of a content question can be preceded by the copular particle *na* without changing the behaviour or meaning of the sentence. The following two questions (see Fig. 4.26) occur in direct sequence and show how the copula na can be present or not without semantic consequence:

This leads us to interpret question-words as focused, and the rest of the sentence as the focus-frame. In the absence of the focus particle na, the question word is promoted as the root of the sentence. In this analysis, the question word has a double function:

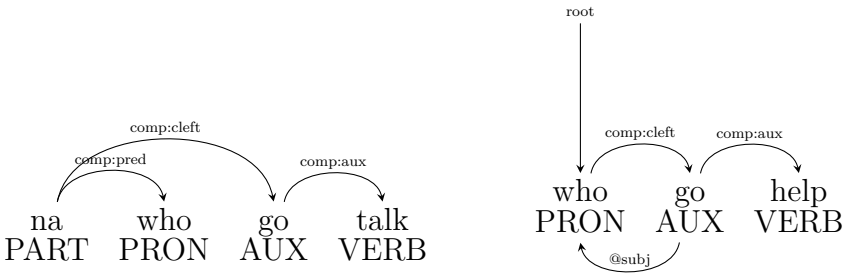


Figure 4.26: Comparative analysis of interrogatives with and without the copula *na*

It is the root of the sentence and a dependent of the verb. The complexity of the cleft structure of content questions cannot be captured by UD which treats the sentence verb as a root. Moreover, the parallelism between the two questions will not be kept by UD, as the one with *na* will be treated as a cleft, with the cleft phrase as the head, contrary to the other question without *na*. As a compromise between surface syntax and convertibility to UD, a second link has been added to the root, which annotates explicitly the dependency of the question word (this second relation is preceded by a "@", see *@subj* in 4.26).

In our Surface-syntactic representation, both cases are represented by means of a cleft structure, see the above analysis. This is congruent with many analyses of wh-words which consider that they occupy two syntactic positions, one as a complementizer and another as a pronoun inside the clause they complementize (see, in particular, [Tesnière, 1959, ch. 246]). During the conversion into UD we can only keep one of the relations, we have to keep the second relation as this follows the UD analysis of relative clauses. This leads to what Gerdes and Kahane call a *catastrophe* [Gerdes and Kahane, 2016] (representing syntactically related constructions in a different way).

4.3.3.3 Serial Verb Constructions

The influence of adstrate vernacular languages, belonging mainly to the Niger-Congo family, is observed in the use of Serial Verb Constructions, that is "monoclausal construction[s] consisting of multiple independent verbs with no element linking them and with no predicate-argument relation between the verbs." [Haspelmath, 2016, p. 292].

We used the subtyped relation *compound:svc* for these constructions. Sentence 4.27 contains an example of a serial verb construction (`carry` - `put`).

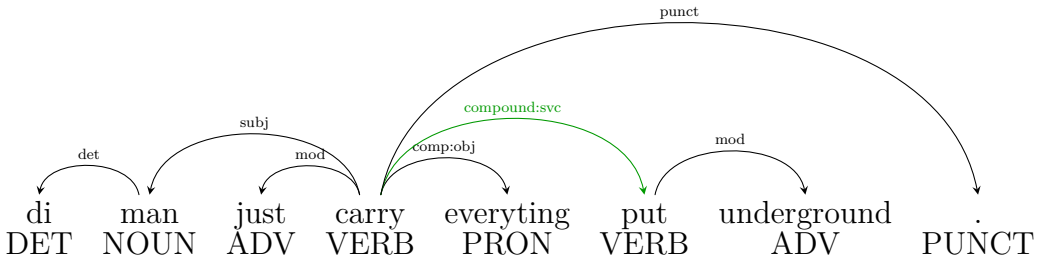


Figure 4.27: Serial verb construction annotation in Naija (SUD) for the sentence `di man just carry everyting put underground` 'The man just buried everything.'

Early on in the project the annotators reported that they were unsure on how to annotate serial verb constructions.

The *compound:svc* label was already used to describe similar phenomena in some of the UD treebanks so we kept the same label. What remained problematic however was the direction of the relation between the 2 (or more) verbs in the construction.

We had expected the head to always be on the right, but frequently found that based on semantic criteria we were sometimes swayed to put the head on the left. Preferring consistency to a finer analysis that would have very likely resulted in poorer agreement between the annotator (and worried that our parser would struggle to correctly analyze these constructions with the higher head-direction entropy), we chose to arbitrarily annotate the left verb as the head.

We still left the door open to revising this decision after a larger portion of the corpus would be annotated, and we could analyse it and find systematic criteria to select a head. This decision shows that in terms of concerns **consistency** (measured by annotator agreement), **learnability** (measured by parser performance) were more important to us in this instance than **granularity** as we conflated left-headed serial verb constructions and right-headed serial verb constructions. It was an instance were we recognized that more precise annotations were to be left to a later date, when sufficient data would be available to formulate adequate criteria and explain the distinction in the guidelines. In this sense, the iterative method of annotation seems

inevitable if we want to accurately interpret the data.

4.3.4 Conclusion on the NaijaSyncor treebank

We have described the workflow for the development of the gold section of the NaijaSynCor treebank in the SUD annotation scheme, and we have shown the SUD analysis of some interesting syntactic constructions of Naija.

In parallel with the treebank construction various interfaces have been developed to access the audio corpus, the transcription, and the various types of annotations. For example the most recent version of the SUD syntactic annotation is accessible at http://match.grew.fr/?corpus=SUD_Naija-NSC@latest.

In order to be part of the UD treebank family, the treebank is automatically converted to UD using a language specific rewriting grammar. Although the current UD platform does not foresee the joint distribution of the audio data, we expect that the increasing interest in spoken data will eventually bring the UD community to discuss the format that will best allow for phonosyntactic studies.

The perspective of this treebank creation goes beyond purely linguistic interest. It has deep sociolinguistic implications through the creation of a Naija dictionary. In order to create this treebank, we had to create an inventory of spelling variants, and we propose systematic distinctions of function and content words. The tools and resources of the NaijaSynCor treebank enhances the interest in the specificity of Naija grammar, and the project can be seen as a step in the further establishment of Naija as a language.

4.4 Conclusion and perspectives for the chapter

In the first section, we focused on treebank annotation processes, syntactic annotation schemes and their outputs, namely treebanks and annotation guidelines. We investigated various research questions dealing with the content and structure of annotation guidelines. We discussed criteria to keep in mind when designing annotation

schemes such as learnability or complexity of annotation. We have seen that annotation guidelines and treebank developments are iterative and interwoven processes. We also addressed the issue of potential users for the annotation guidelines, who are first and foremost the annotators, but also various end users with specific applications in mind such as language students and more largely all people interested in grammar and language comparisons.

In the second section, we addressed the problem of Multi-Word Expressions (MWE), and their encoding in the UD framework. We showed that for a coherent annotation it is crucial to distinguish syntactically irregular structures from semantically non-compositional units. These notions are highly correlated but distinct and we propose criteria to distinguish them. We placed our analysis in the Universal Dependency framework (UD), which involves a large community of more than 100 teams around the globe. We explore different ways of annotating Multi-Word Expressions based on their syntactic regularity and semantic non-compositionnality. We have shown that in syntactically irregular structures, the part-of-speech of the head often differs for the part-of-speech of the unit as a whole, rendering it unpredictable. As such, MWE need to be introduced as units in the annotation, so that the part-of-speech relevant to the unit can be associated. to associate a part-of-speech to them. In some of the UD treebanks, the *ExtPos* feature (for *External Part-of-speech*) has been added to serve such as purpose. For regular functional MWEs, we propose to specify the relation label to capture the relations between content words, as well as the syntactic dominance relations. We also proposed rewriting rules to transform one scheme into the other. The proposed schemes and distinctions clarify some underspecifications in the current UD scheme that lead to inconsistent analyses. The usage of subtypes fits in unintrusively into the current scheme and could be used for upcoming versions. More generally, it allows back and forth conversions of UD and more classical subcategorization-based dependency annotation schemes.

The third section describes the workflow for the development of the gold section of the NaijaSynCor treebank for the low resource Naija language. We explained our reasoning for using the SUD annotation scheme, and discussed the analysis of some

interesting syntactic constructions of Naija. We explained our process when developing the annotation, relating some of the challenges we encountered to earlier points about ambiguity, annotation complexity and underspecification of the annotation in cases of uncertainty. We also provided an evaluation of the annotation and of its stability. The treebank is available in both the native SUD version and converted into UD using a language specific rewriting grammar.

Beyond the treebank itself, various interfaces have been developed to access the audio corpus, the transcription, and the various types of annotations, as well as a wiktionary[35]. The perspective of this treebank creation goes beyond simple linguistic and interest. It has deep sociolinguistic implications. The tools and resources of the NaijaSynCor treebank facilitate the interest in the specificity of Naija grammar, and the project can be seen as a step in the further establishment of Naija as its own language.

[35] https://naija.elizia.net/

Chapter 5

Collaborative treebank development

Chapter contents

5.1 Introduction 134

5.2 Comparison to related tools 137

5.3 Architecture 140

5.4 New Features provided by Arborator-Grew 142

5.4.1 Class Sourcing 143

5.4.2 Error Mining 145

5.5 Distribution 149

5.6 Conclusion and perspectives 150

In the previous chapter, we have looked at common issues encountered when developing treebanks and the annotation guidelines that accompany them. We have shown that annotation guidelines are expected to evolve during the annotation process, which creates a need for appropriate tools that will facilitate updating the treebanks to respect those new guidelines.

Annotation campaigns will often involve several annotators, who oftentimes provide annotations that disagree with one another. To fully leverage their annotations, the annotation tools should provide a way to compare the disagreeing annotation in order to understand where the disagreement came from, and adjudicate to unify them into a final annotation.

Other useful features include querying tools to find examples to populate the guidelines, get a global view of the various syntactic structure present (or absent) in the treebank, look for inconsistencies and structures that at first glance should not exist in the treebank.

With those concerns and desired features in mind, we looked for an annotation tool that would satisfy those criteria. Many annotation tools support some of these features, but there didn't seem to be one that included all of them. We decided to integrate two tools that had complementary features : Arborator and Grew. Arborator [Gerdes, 2013] is a widely used collaborative graphical online dependency treebank annotation tool. Grew [Guillaume et al., 2012a; Bonfante et al., 2018] is a tool designed for querying graphs and rewriting them through the use of formal rules. It has been used to develop both syntactic treebanks and semantic graphbanks.

This chapter will present the development of Arborator-Grew, a collaborative annotation tool for treebank development, and show how its design was guided by those common problems encountered during the annotation process of syntactic treebanks, and during their exploitation. Arborator-Grew opens up new paths for collectively creating, revising and maintaining syntactic treebanks and semantic graph banks.

The chapter is based on the publication : Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Confer-*

ence, page 10, Marseille, France, 2020. URL https://www.aclweb.org/anthology/2020.lrec-1.651. LREC.

My contributions to Arborator-Grew relate to the backend-end development (REST Controllers, Services, Models & ORM see Figure 5.2) and design of the features. For this latter aspect, I relied heavily on my experience in annotating the Naija treebank using the legacy Arborator. At the time I also used a local installation of Grew to query my annotations, extract various types of examples to enrich the guidelines and check for inconsistencies, which gave me a good understanding of the type of workflows that could be facilitated through their integration with one another. I also used the teaching mode of Arborator while teaching undergraduate students at the University Sorbonne Nouvelle, and was familiar with using annotation and query tools in a pedagogical way.

5.1 Introduction

Dependency treebanks have become the standard resource for training syntactic parsers, and substantial efforts have been undertaken to develop large scale and multi-lingual treebanks. The flagship project is certainly Universal Dependencies (UD) [McDonald et al., 2013], which has served as the input to numerous parsers, text generators, and morphological taggers around various shared tasks [Zeman et al., 2018; Mille et al., 2019; McCarthy et al., 2019]. The impressive project with more than a hundred treebanks in the same annotation scheme for 90+ languages, combined with great online viewers and query tools have given increased visibility to the project also inside the syntax and typology communities [Croft et al., 2017; Gerdes et al., 2019b].

Yet, for UD as well as for other treebank creation projects, many of the treebanks contain substantial errors and inconsistencies, which can be attributed to three main causes:

1. Many of the UD treebanks are converted from other formats. Converting from one scheme to another can result in systematic errors especially if the original scheme is less informative, or when the converters are incomplete.

2. Some descriptions in the UD guidelines are underspecified and leave room for different analyses of the same construction, inside a language, a language group, or generally among languages, *cf.* the constantly active UD discussion group on GitHub. When a consensus is reached on how to annotate a specific phenomenon, harmonisation is required.

3. Treebank maintenance is a painstaking endeavour, which requires a lot of time and energy. This is especially true in UD as the minimal specification are updated every 6 month (with each version release), and a failure to comply with recent specifications could result in the treebank being "retired" (i.e removed from the next release).

Generally speaking, dependency parsing and tagging with recent NLP tools can overcome, to a certain degree, rarely occurring errors in the training corpus. However,

in other applications such as language teaching, the study of typological comparative measures (such as word order variations, or the distribution of specific syntactic constructions) or when looking for counterexamples to syntactic claims, these errors can influence the results significantly. See for example the differences between treebanks of the same language reported by [Chen and Gerdes, 2017]. Therefore, it has become essential not only to facilitate treebank curation for treebank maintainers but also to find ways to open treebank corrections to a wider audience of linguists and language students.

Treebank annotation errors essentially belong to one of two types : occasional slips of attention on the part of the annotator and systematic discrepancies with the desired correct analysis[1]. The former type of problems can be addressed by means of an easy access to "strange" constructions[2]. Annotators (or informed users of the treebank) can then look at these potential issues and either directly fix the error or validate the rare construction as being correct. The second type of problems (systematic discrepancies with the desired correct analysis) needs systematic corrections by means of a graph grammar and non-regression validation[3]. Examples of this kind of systematic errors are provided by [Haverinen et al., 2011] where the authors found that annotators frequently confused direct objects and nominal modifiers in Finnish, syntactically readily mistakable. But annotators also confused subjects and adjectival modifiers for the surprising reason that the annotation tool's shortcut keys are placed next to one another on the keyboard. These types of systematic errors can easily be detected using Grew [Bonfante et al., 2018].

The Arborator-Grew tool provides support for the whole process of treebank creation, error-mining and curation. It is essentially a front-end editor to the Grew graph rewriting system[4] [Guillaume et al., 2012b; Bonfante et al., 2018] with added fea-

[1] In section 4.1.3.2 we also discussed that some systematic deviation

[2] In the simple sense of being rare or in the sense of not being aligned with our knowledge and intuitions about the structures present in the language

[3] This can by realized by means of a set of correct target trees that have to be attained by the conversion grammar. The grammar can be iteratively refined until it correctly transforms the source trees into the target trees.

[4] `http://grew.fr`

tures aiming to smooth out the annotation process. The features pertain to project management, annotator training and facilitated error mining. The first version of Arborator-Grew was released in 2020 and it has been updated since then. It replicates the features of the legacy Arborator [Gerdes, 2013], in particular its class-sourcing tools [Zeldes, 2017b], while improving and modernizing queries, error-mining, versioning, and collaborative features. To the best of our knowledge, it was the first tool to integrate complex graph querying and treebank annotation software. The advantages of this combination will be discussed in Section 5.4

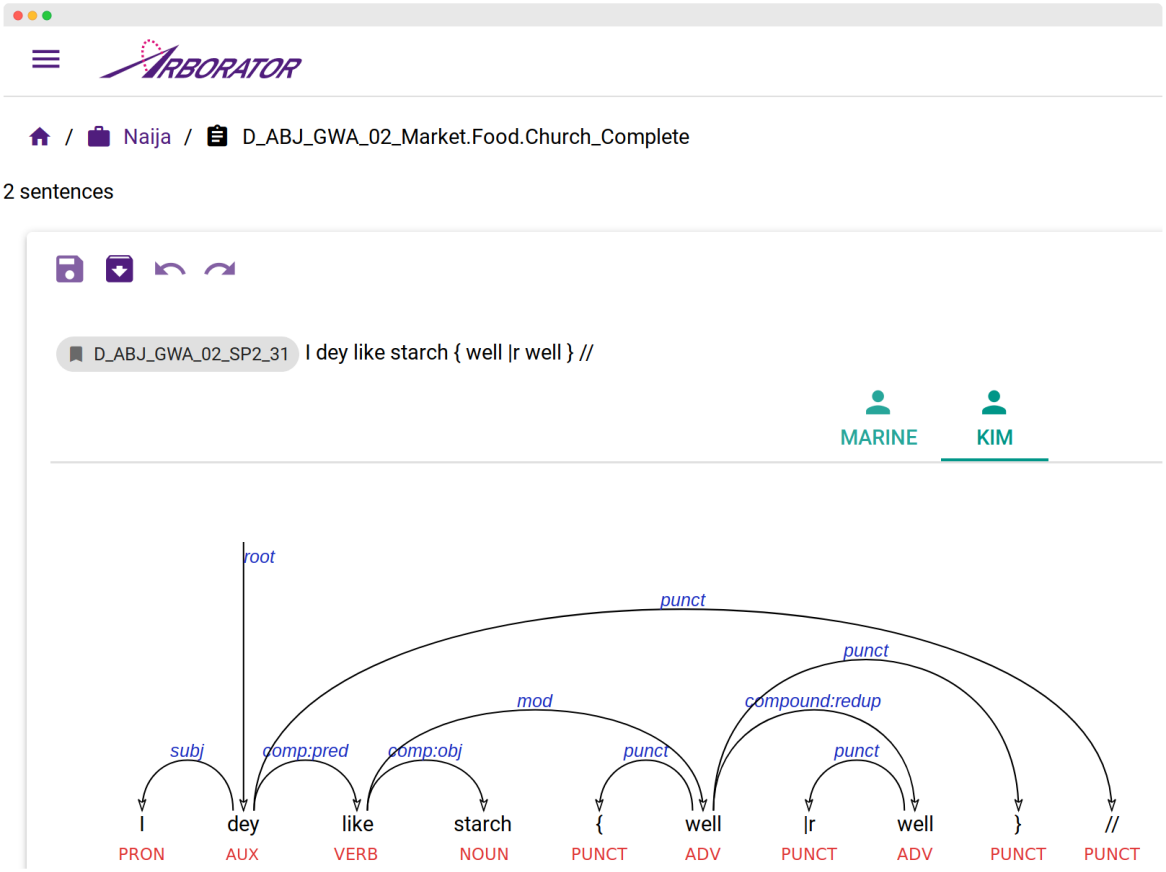


Figure 5.1: A screenshot of the user interface that gives access to the different annotations of a sentence, with one dependency tree per user. The sentence is drawn from the treebank annotation project of spoken Naija (Nigerian Pidgin-Créole), discussed in section 4.3. The sentence can roughly be translated by *I really like maize dumplings*.

5.2 Comparison to related tools

Arborator-Grew is by no means the first collaborative annotation tool to be developed. Instead it is part of a long tradition of work dedicated to facilitating tree visualisation, edition, querying and rewriting. In this section, we will present some of these tools and their features.

Dependency visualization. CoNLL-U viewer[5] and TüNDRA [Martens, 2013] provide graphical interfaces to visualize dependency trees from uploaded CoNLL-U formatted files (which has become the standard format for syntactic tree annotation). They are easy to use as they don't require specific software installation or user accounts. For the user familiar with LaTeX and the command line, conllx-to-tikz-dep[6] is a useful resource that transforms conll data into tikz-dependency format which will output tree visualisations such as the ones included in this thesis once compiled in LaTeX[7].

Dependency annotation. These tools are designed for treebank development and allow users to build dependency structures on top of their corpus, without having to resort to manually editing the CoNLL-U files. Some tools are minimalist, lightweight, and usually offline, while others allow for collaborative online annotation of multi-layer treebanks.

The most used tool is probably Brat [Stenetorp et al., 2012]. In Brat, users can annotate any highlighted span (the user decides where tokens begin and end), which makes it a great tool for (named) entity annotation and chunking. Just like Arborator, it also supports relation annotation using a drag and drop gesture to link tokens. WebAnno [de Castilho et al., 2016] is a similar tool that shares the visualization front-end of Brat, while modernizing keyboard interactions and back-end, as well as allowing for more web-based project configurations. Recent CoNLL-U files need to be converted first in Brat's and WebAnno's internal standoff formats.

[5] https://github.com/rug-compling/conllu-viewer
[6] https://github.com/tetsuok/conllx-to-tikz-dep
[7] TüNDRA also provides LaTeX as one of the export options

Single user online graphical CoNLL file editors include Arborator's Quick online tool[8], Annotatrix [Tyers et al., 2017] (providing Latex export), and the ConlluEditor [Heinecke, 2019]. The ConlluEditor is especially noteworthy for its easy token splitting and joining, its stemma-like horizontal visualization, its integration of UD validation scripts, and its interaction with GitHub versioning. In recent years, it has also been updated to include partial support for Grew Match search patterns and support for CoNLL-U Plus[9] among other features.

One last annotation tool worth mentioning in this context is ZombiLingo [Guillaume et al., 2016], a tool for crowd-sourcing of the syntactic annotation process through gamification. Users have to pass basic proficiency tests before playing: they are presented with a sentence at a time, for which they have to determine one single relation, such as finding the subject of a verb. While annotations, users can gain points and enter a public wall of fame, making the experience enjoyable. Dependency trees are obtained by combining the the annotations that most players agree on.

Query tools. The most famous linguistic annotation query tool is arguably Annis [Zeldes et al., 2009]. Annis proposes its own query language AQL (ANNIS Query Language) which can handle multi-layer annotations, a graphical query builder, chunk, phrase structure tree, and dependency visualization, integration of sound files, and queries on multiple tokenizations thanks to its stand-off format [Krause et al., 2012]. Yet, due to its complexity, Annis requires a non-trivial installation and data-insertion process. AQL is rather verbose, and it is rather slow. Other tools are more specifically designed for queries into single-layer dependency treebanks. One of these tools is Dep_Search from Turku [Luotolahti et al., 2017]. It is very lightweight and fast, with a succinct and quite powerful query language based on TGrep [Rohde, 2005], though quite unfamiliar and tricky for users trained on other query languages.

Most of these tools are designed to provide the matching trees alongside a simple count of matches, but they do not include further statistical data about the query

[8] `https://arborator.ilpga.fr/q.cgi`

[9] An extension of the CoNLL-U format, designed to facilitate the storage of additional annotation layers, see https://universaldependencies.org/ext-format.html

results. Grew-match[10] goes a step beyond that with the possibility to cluster query results based on the properties of any nodes or edges. This includes simple clustering of the form for any lemma, thus providing a list of forms, and also the clustering of the relation between any two parts of speech, providing a list of relations that link the two parts of speech. Query results can also be clustered based on the presence or absence of a subpattern[11]. The integration of this query system into Arborator-Grew and its usage will be explained in Section 5.4.2.

Another remarkable tool that goes a step further is the Trameur [Fleury and Zimina, 2014]. It applies corpus linguistics statistical tools to raw corpora, and, in its online version iTrameur[12] also to dependency treebanks. It can therefore show significant over or under-representations of specific sub-trees in one sample compared to another.

One important question in the design for multi-user annotation systems is the status of tokenization. Brat and WebAnno allow the user to annotate any span of text, while most other tools consider that the annotated object is a series of (pre-defined) tokens. It may seem like a more natural choice to actually annotate texts and to combine the tokenization and syntactic annotation step into one single task. Note however that annotator-based tokenization complicates significantly the computation of inter-annotator agreement, as we jointly observe tokenization and syntactic annotation. And most importantly, tokenization is either trivial and orthography based (i.e. a token is a sequence of letters, possible tokenization errors are corrected through the use of specififc syntactic annotations) or based on lexical and semantic criteria, which makes it a challenging task to reach a satisfying inter-annotator agreement [Farahmand et al., 2015; Savary et al., 2017]. Arborator takes a middle stand, making use of its hierarchy of user modes, see Section 5.4, and allows validators and project own-

[10] `http://match.grew.fr`

[11] An exemple query is included in the appendix (searching for the subjects of verbs, and whether these subjects are on the left or right of their governor) A.1.1, and the results for the Naija treebank can be found at the url https://universal.grew.fr/?custom=654f5cf0abe2e

[12] `http://www.tal.univ-paris3.fr/trameur/iTrameur/` iTrameur only has a French interface.

ers to modify (delete, add, join, split) tokens, but these changes are then carried out on all trees, whatever the user, of the modified sentence. Such a global modification of a sentence's tokenization can cause other annotators' trees to be disconnected or different from the desired structure. This behavior is the only exception to the basic Arborator rule which states that users can view other annotators' trees, depending on their access level, but can only create or modify their own tree.[13]

5.3 Architecture

Arborator-Grew is a complete redevelopment of the legacy Arborator, so that the only common code is the Python CoNLL-U parser. The legacy Arborator is written in Python 2. It is a simple CGI web page and uses a SQLite[14] database with the FTS4 module for fast text searches. User identification is handled via Login Tools[15], and the front-end runs in Jquery-enhanced Javascript with the rather slow Raphael.js[16] for drawing SVG tree graphs. On the other hand, Arborator-Grew consists of three completely separate pieces of software that interact via stateless REST interfaces and follows a Model-View-Controller (MVC) architecture shown in Figure 5.2.

Data Persistence. The database storage relies on the Ocaml[17]-based Grew storage system to which we have added an API accepting json queries. It should be noted that the Grew API can be run on a different server than the main Arborator-Grew.

Back-end and User Persistence. The interaction between the back-end and the user interaction is done through Flask[18], an application written in Python 3 that uses the Authomatic library for social login via Google or Github. The Flask application is tasked with controlling the reading and writing access to the storage back-end, han-

[13] If user A views user B's tree and modifies it, the new tree will be saved as the most recent tree of user A, possibly overwriting user A's original tree while leaving user B's tree untouched.
[14] https://www.sqlite.org/index.html
[15] http://www.voidspace.org.uk/python/logintools.html
[16] https://github.com/DmitryBaranovskiy/raphael
[17] https://ocaml.org
[18] https://palletsprojects.com/p/flask/

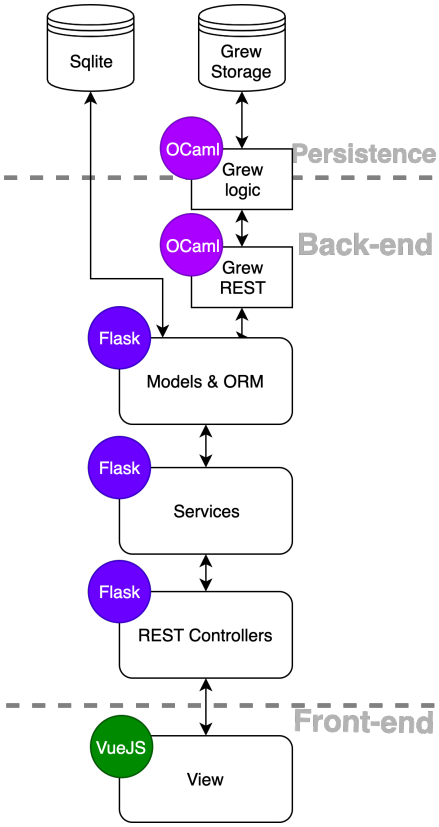


Figure 5.2: Software architecture of Arborator-Grew.

dling software logic through services, and managing resource routes for the front-end. User information and access rules are persisted in a SQLite database interfaced by an object-relation mapping[19]. The Flask application also keeps in memory frequent queries, mainly for tree comparison in the teaching mode, in order to lower the strain on the Grew server. Note that the Grew API provides CoNLL-U data that the Flask application only passes on to the front-end for visualization. Only meta-information such as the user name and time-stamping is accessed and modified at this stage. Hence, the Flask application represents the Model and Controller in the MVC architecture while the Grew server represents the Data Access Objects related to trees from the Model.

Front-end. Lastly, Arborator-Grew's front-end (Figure 5.1 shows a screenshot) is

[19]We used the ORM framework SQLAlchemy: https://www.sqlalchemy.org/

written using a Javascript based framework named VueJS[20], which facilitates the development of reactive, modular, and flexible front-end user interfaces, and enjoys rising popularity among web developers. In particular, the Arborator-Grew front-end makes the View part of the MVC independent from the back-end. This facilitates the development of mobile or desktop versions because the same base code can be automatically translated using Node ecosystem packages such as Electron and Cordova. The actual dependency graph is drawn via the Snap[21] SVG library. Arborator-Grew also deals with the taxonomic relation systems inherent to UD, the "universal" relations followed by language-specific sub-relations, such as *aux:pass* and to SUD, the Surface-Syntactic version of UD [Gerdes et al., 2019a], which incorporates a third level with deep-syntactic relations such as *comp:obj@x*. This structuring of the relation tagset helps to limit the degree of freedom at any point of choice, which reduces the complexity of annotation. Arborator Grew can be configured to show separate choices of universal and secondary relations, see Figure 5.3.

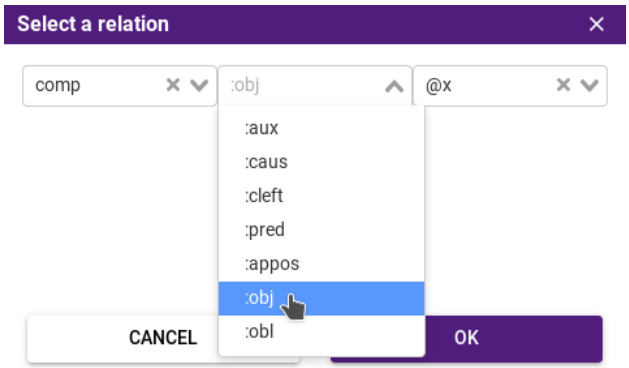


Figure 5.3: Interface for the relation selection for a SUD project configuration.

5.4 New Features provided by Arborator-Grew

Merging together Arborator and Grew results in novel ways to build treebanks and enhance existing annotations. In this section, we focus on two main aspects: classroom

[20] We used https://vuejs.org/ associated to multiple components mostly coming from the Quasar VueJs Framework: https://quasar.dev/

[21] http://snapsvg.io/

collaborative treebank creation (Section 5.4.1) and error mining inside an existing treebank using queries (Section 5.4.2).

5.4.1 Class Sourcing

One of the key features of Arborator is collaborative annotation, thus controlling who has access to the resources must be well thought out. The legacy Arborator was laid out for separate instances on separate servers and it was not designed for multiple projects on the same server with different administrators and different people teaching separate classes. In Arborator-Grew, we distinguish the following roles[22]:

Role	View	Edit	Validate	Exo mode	Project administration
Annotator	x	x			
Validator	x	x	x		
Project Admin	x	x	x	x	x

Table 5.1: Permissions according to user role in Arborator-Grew

The project administrator has access to many settings and options including setting the appropriate roles for users, setting up the list of available annotation labels, the annotations levels (UPOS, XPOS, morphological features, sentence features...) to be displayed, the ability to ensure that annotators do not see other annotators' trees (to avoid being influenced and in order to make sure that the agreement can be truthfully measured), and the ability to set up various *Exercise Modes*.

This configuration allows for *Class Sourcing*, which is a way to merge treebank creation and teaching of academic students, interns, or colleagues willing to learn about syntax. The precision that can be obtained in the setting of class-sourcing has been analyzed in [Gerdes, 2013], and the actual treebank creation has been demonstrated with the GUM Corpus [Zeldes, 2017b], using the legacy version of Arborator for the syntactic component of the corpus.

Annotating actual texts is a great way of studying the syntax of the students' first languages, as well as of foreign languages where the students have a high degree of proficiency. Arborator, right from its legacy version, was conceived to be used in the

[22]There is another role *Guest* which can view trees in both open and public projects, but can only edit trees in the former. If the project has the *Private* status guests cannot explore it.

classroom, in particular for undergraduate and graduate students.

Exercises modes. Beyond the simple exploration of an existing treebank by means of querying and in-class discussion of various structures, Arborator-Grew allows the configuration of exercises. Exercises have four modes: *graphical feedback* (the student can click on a "check" button and wrong categories or relations are marked in red), *percentage* (the student receives a feedback in the form of a percentage of correct categories and relations, and they have to find on their own where the errors are located), *teacher visible* (the students can see the reference tree of the teacher, but cannot modify it directly – they have to redraw the tree from an empty annotation), and *no feedback* where only the teacher can receive the student's score.

Teachers need to have an administrator status in the project in order to set up an exercise. They can then choose an exercise mode and determine a reference tree per sentence. Then, the reference tree will be used to provide students with the desired amount of feedback and finally, to compute the students' scores which can be exported by the teacher. Arborator's exercises have been tested on various levels and in various countries, and the feedback has been positive from the teachers as well as from students. Syntax exercises seem to feel more like a computer game, in particular the *percentage* exercise mode, where motivated students try hard to reach a 100% score.

Treebank construction in the classroom. Training is an essential first step to class-sourcing, and one can go one step further with a class of students pre-trained on a set of exercises: They can be asked to annotate samples where no reference annotation exists. Depending on the number, the level, and the syntactic and linguistic proficiency of the students, various setups have been tried out. In small groups of graduate students, with interns employed for the task, and among colleagues that want to develop a treebank, the most common configuration is that we provide a first draft of an annotation guide, and we assign a sample to one or two annotators. They annotate the sample and note difficulties that are discussed in a group meeting. Together, the annotation guidelines are updated and specified in order to answer the

questions the annotators had.

In the setting of a larger undergraduate class, samples can be distributed to larger set of students, which allows to compute trees that obtain the highest "votes" by the students in a ROVER-like fashion [Fiscus, 1997]. The evaluation of the students has then to be done manually, for example by randomly sampling a few of the student's trees. To automatize the evaluation, samples can be created from both reference sentences from a gold-standard treebank and yet to be annotated pre-parsed sentences. Optionally, errors can be inserted in both types of trees, using the parser's confusion matrix to make "plausible" errors. For each student an individual set of sentences is prepared, that is given as an assignment. Then, scores can be automatically computed for the student evaluation[23], but also for a better ROVER vote, where the students' grades are used as a confidence score in the computation [Gerdes, 2013].

5.4.2 Error Mining

Most treebank developers have already stumbled upon errors in their own, or other people's, treebanks while browsing through the trees[24]. If time allows, the tree has then to be looked up in the latest version of the samples, corrected, and the improved version uploaded. Ideally, we would like to then check whether similar trees have similar error patterns. Prior to Arborator-Grew, this task would be quite cumbersome, and this difficulty was one of the central motivation for the merging of Arborator and Grew.

Pattern matching. The central feature of Grew-match is precisely the query of syntactic patterns in a treebank project. Grew has its own query language which is easy to learn and very powerful. This query language is well-documented[25] and a tutorial is made available on the Grew-match site[26]. The pattern matching system can match

[23] In our experience this evaluation shouldn't be used to actually grade the students, students with a great understanding of the annotation have repeatedly over-corrected the initial annotation, sometimes discovering errors in the gold annotation or proposing alternative but not uncorrect analyses.

[24] This common phenomenon is sure to happen when one wants to look for an example to show a colleague.

[25] https://grew.fr/doc/request/

[26] http://match.grew.fr/

subtrees based on forms, lemmas, part-of-speech tags, presence of a morphological feature, value of the morphological feature, incoming and outgoing dependencies (typed or not, enhanced or syntactic), linear order between nodes and any combination of these features, which can result in very complex queries (see an example of a more complex query in Figure 5.4). It can also filter out these results based on negative patterns (patterns that must not appear in the graph).

Grew Matcher

```
% Search for verbs without subject.
% This kind of request is written with step by step adding of each 'wihtout' in order to find potential annotation errors
pattern { V [upos="VERB"] }  % <-- Looking for a verb
without { V -[nsubj|csubj|nsubj:pass]-> S} % <-- without subject
without { V [VerbForm = Ger|Inf|Part] } % <-- exclude verbforms that are usual without subject
without { V [Mood = Imp] }% <-- exclude imperative mood
without { N -[conj]-> V }% <-- in case of conjonction, the second verb does not directly have a subject
without { N -[fixed]-> V }% <-- remove verbs in fixed expressions
```

Figure 5.4: Pattern to look for potential errors on verbs without subjects

The nodes that match the pattern are then highlighted in the trees on the results page. See an example of a query on *comp:aux* relations[27] in the SUD treebank of Old French in Figure 5.5.

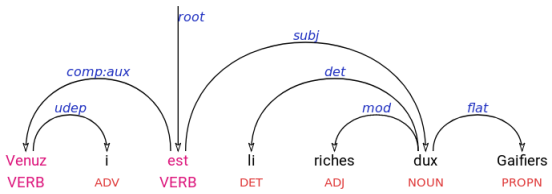


Figure 5.5: Search result querying a *comp:aux* relation, with pink highlighting of the governor and the dependent. The sentence from the Old French text *La Chanson de Roland* 'The Song of Roland' (1040 - 1115) can be translated by *The rich duke Gaifier has arrived."*

All of this could be done before the integration of Arborator and Grew, however what couldn't be done was the actual modification of the tree via a graphical interface. The next logical step then, was to integrate Grew as a perfect querying system inside Arborator, thus allowing treebank creators and users to easily find syntactic patterns inside their projects and to directly correct them. This is done throught the Grew-

[27]The Grew pattern is described in the Appendix, see A.1.2.

match component (see Figure 5.6) inside Arborator-Grew, which can be opened inside a project or a sample. The results page, which contains all of the matching trees, can then directly be edited and saved, saving us time and energy.

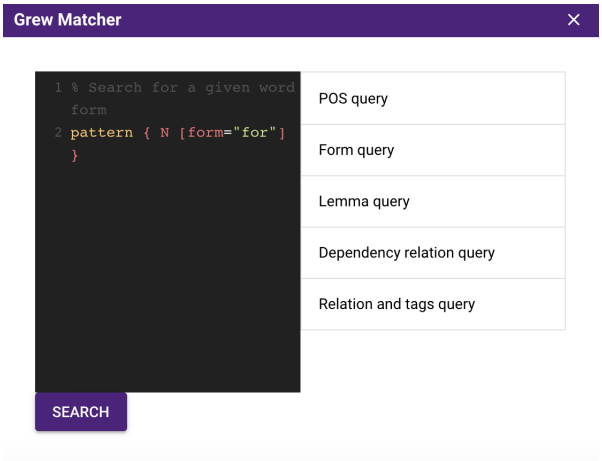


Figure 5.6: Grew-match integration component, with example queries on the right and automatic idiosyncratic highlighting of the Grew query language.

Clustering the results. Grew also has the ability to cluster the results of a query based on one or several features, thus making it easier to quickly sort through the results. For example, one could look for all the verbs with an object and cluster them based on the part-of-speech of the object. Another example, previously mentioned, would be to look at all the relations between verbs and their subject and find out when the subject comes before or after the verb in the sentence.

This functionality can also be used more systematically to build a *relation table*. This table summarizes all dependencies within a project, based on the part-of-speech of the governor and of the dependent. Having this relation table easily accessible is a great way to look for rare structures and potential errors inside a treebank, and to get an overview of the existing structures. In Arborator-Grew, the user can open this table (see Figure 5.7) and access directly the trees that match the pattern to see if the analysis is correct, and update it if they find an error. Work on this distributional relation table can easily be integrated in a pedagogical context where students try to get a feel for the various possible structures inside a language.

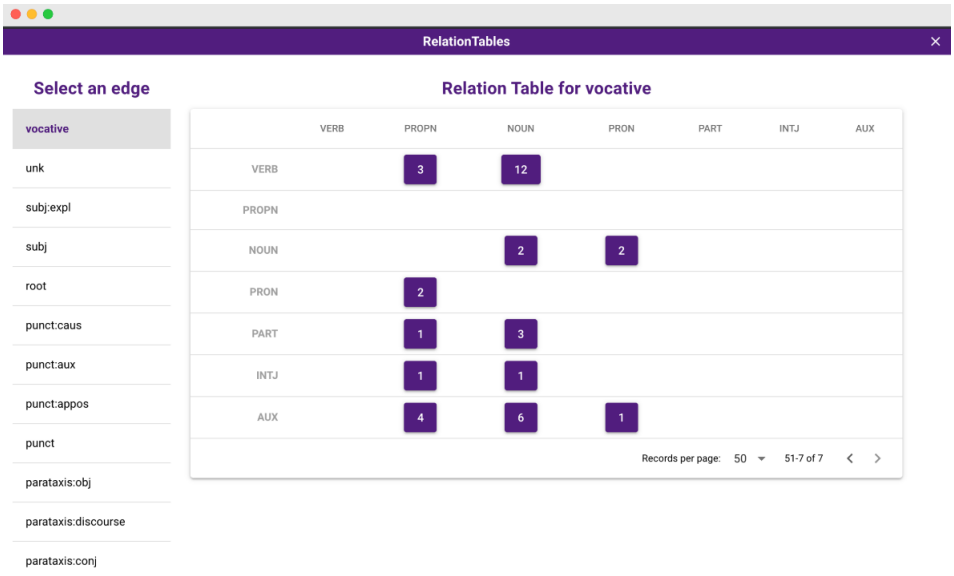


Figure 5.7: Relation table showing a count of all occurrences of *vocative* relations between a governor (category to the left) and its dependent (category on top) based on their respective parts-of-speech. Here the table query searches in the annotator's own trees; alternatively, the most recent accessible trees can be taken into account. The annotator can directly click to visualize, for example, the two pronouns that have a proper noun as a vocative dependent, in order to verify the two corresponding trees. If a tree contains an error, it can directly be corrected.

Applying rewriting rules. The rewriting functionality which was at the core of the Grew software, can be accessed by annotators and annotation project managers. The user can write rewriting rules following the Grew syntax which is thoroughly documented[28] and apply it to a sample of text. In 5.8 we provide an example of a simple rewriting rules, where subjects (S) are narrowed down to only nouns and proper nouns. For all those subjects, we remove the edge (e) that links them to their governor which is labelled as *subj* in SUD fashion, and we replace it with another edge labelled as *nsubj*, the label for nominal subjects in UD.

The resulting trees are then displayed and can manually be checked by the annotator and saved if they correspond to the desired output. This will facilitate the automatic correction of errors, adaptation of treebanks to updated guidelines and con-

[28]Documentation is available at `https://grew.fr/` where we invite interested readers to check the *Patterns*, *Command Syntax* and *Rules* pages that describe respectively how to match the correct patterns, how to write adequate commands to apply to the matched pattern and how to combine both of these into rewriting rules.

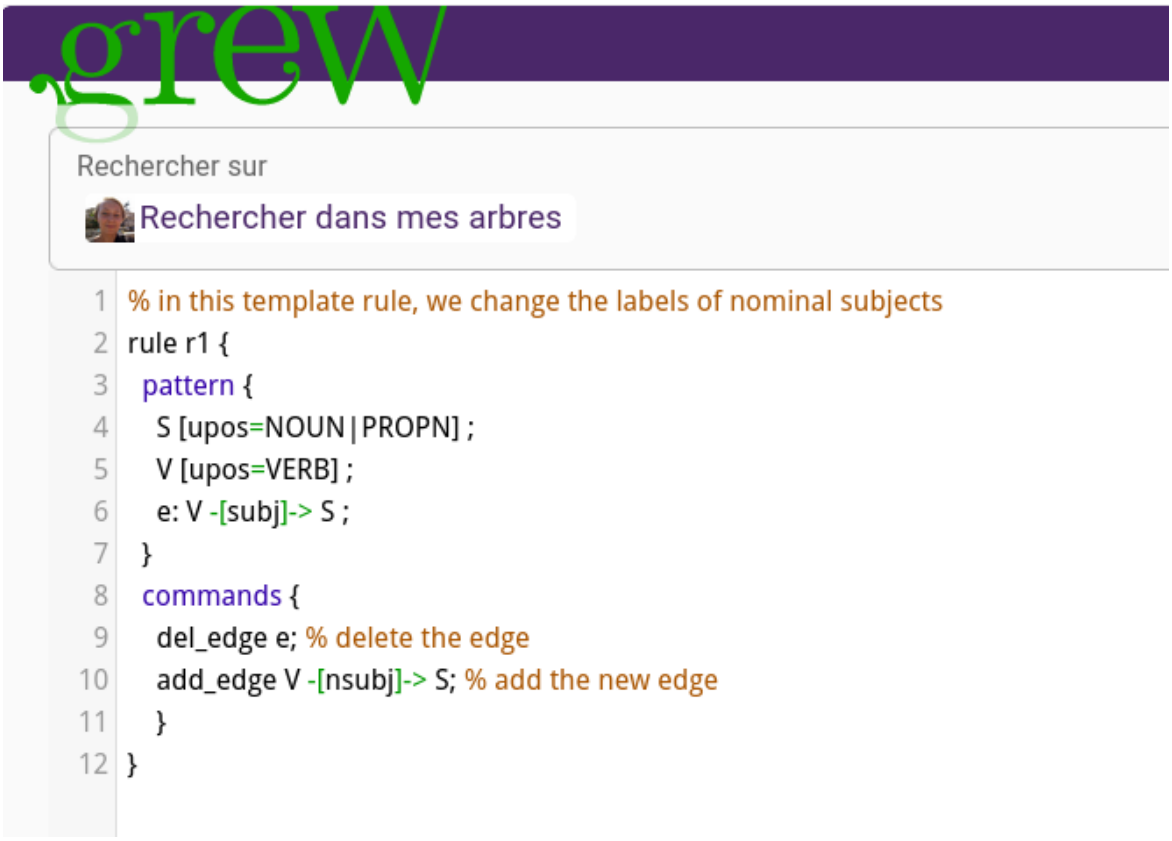


Figure 5.8: Rewriting rule to update the label of nominal subjects. This could be used to transfer from one annotation scheme into another.

version of treebanks into different annotation schemes as long as the changed can be formalised into rewriting rules.

5.5 Distribution

An instance of Arborator-Grew is accessible at https://arboratorgrew.elizia.net/#/. The source code is available through three repositories :

- Grew : https://gitlab.inria.fr/grew/grew_server
- Arborator backend : https://github.com/Arborator/arborator-backend
- Arborator frontend : https://github.com/Arborator/arborator-frontend

Arborator softwares are licenced under the GNU Affero General Public License

v3.0 while Grew server is licence under the CeCILL Licence v2.1 [29].

5.6 Conclusion and perspectives

Arborator-Grew has been designed to integrate collaborative annotation with treebank querying and rewriting functionalities. This tool allows for faster treebank development and provides a powerful query system which can be used for error-mining, extracting example sentences, and getting an overview of the structures present in the treebank. One of the main advantages of the tools is that on top of the query system, systematic transformations can be applied to the annotation whether to correct errors, or as a way of converting to another annotation scheme.

Arborator-Grew has now been successfully used in various contexts such as in a pedagogical setting and in actual annotation campaigns. Several treebanks are now being developed and maintained using Arborator-Grew, which shows that it provides an answer to the community's needs.

As the source code is open, researchers have also been able to tailor it to their specific needs by improving upon the original version. One such example is Arborator-Grew-NILC[30], developed by a team working on the Porttinari treebank [Pardo et al., 2021], a large multi-genre corpus of manually annotated Brazilian Portuguese texts.

The enhanced features focus mostly on improving the user experience of the tool. They introduce shortcuts to increase the efficiency of annotators on the most common annotation tasks (saving a tree, undoing a change..). Their layout also shows the status of the current tree so that the annotator can keep track of their progress through a sample of text to annotate.

One other interesting development is that they integrate warnings that alert to user if their annotation does not conform to validation criteria similar to those required by UD :

[29] http://cecill.info/licences/Licence_CeCILL_V2-en.html
[30] An instance of it is available at https://arborator.icmc.usp.br/

> Another additional feature is the inclusion of four different warnings. Their goal is to make the annotators aware of some potentially harmful characteristics of the annotation they are trying to save. One warning is to indicate the existence of non-projective trees. Another warning is to show that the sentence does not have a root. There is also one to show that there are tokens without a defined syntactic head. Finally, there is one to show that there are multiple tokens assigned as root. [Pardo et al., 2021]

As part of their development process, they also plan on adding a validation script where "forbidden patterns" would be described :

> To facilitate error-mining and treebank validation, we are also planning to integrate a validation script that would describe forbidden patterns. This practice has now become part of the Universal Dependencies project, where all treebanks must pass through a validation script to be accepted in the new releases that occur every 6 months, so as to maintain the overall quality of the annotation. [Pardo et al., 2021]

This is interesting to us, as this is in line with improvements we intended to add to Arborator-Grew, but did not get around to implement.

Initiatives such as this one, provide valuable feedback on what could be improved in Arborator-Grew. As treebanks take so much time, care and effort to develop, and with active maintenance becoming a requirement to be part of the ongoing UD releases, we can expect that features that target error correction, adaptation to evolving guidelines and validation will become more prevalent in future versions of treebank annotation tools.

Chapter 6

Global properties of treebanks

Chapter contents

6.1 Menzerath-Altmann & Heavy Constituent Shift 157

6.1.1 Related works on Menzerath-Altmann Law 158

6.1.2 Heavy Constituent Shift 158

6.1.3 The Co-effect Hypothesis: Combining Heavy Constituent Shift and Menzerath-Altmann Law 161

6.1.4 Methodology 162

6.1.5 Results 164

6.1.6 Conclusion 165

6.2 Adapting MAL to more syntactically defined units 166

6.2.1 Linear dependency segment 169

6.2.2 Experimental results 171

6.2.3 Conclusion 174

6.3 Syntactically grounded vs artificial trees 175

6.3.1 Looking into the properties of syntactic dependency trees . 177

6.3.1.1 Features 177

6.3.1.2 Hypotheses 178

6.3.2 Random tree generation with constraints 179
6.3.3 Results and discussion 183
6.3.3.1 Correlation between properties 183
6.3.3.2 Distribution of configurations 184
6.3.4 Conclusion 187
6.4 Conclusions of the chapter 188

This chapter is dedicated to works on global properties of treebanks which we explored on a variety of languages, benefiting from the UD annotation framework. The explorations are related to the Menzerath-Altmann Law (hereafter MAL), named after linguists Paul Menzerath and Gabriel Altmann. MAL is a principle that states that within a linguistic construct (like a sentence, phrase, or word), the larger the construct, the shorter its constituents tend to be [Altmann et al., 1989; Hřebíček, 1995; Cramer, 2005b]. In other words, as a linguistic unit becomes larger, its subunits become proportionally smaller.

In this context, a first study [Chen et al., 2022] investigates the link between MAL and another linguistic principle, namely the Heavy Constituent Shift (hereafter HCS) [Ross, 1967; Stallings et al., 1998]. This principle revolves around the ordering of constituents in a sentence, stating that heavier elements are placed later. This syntactic phenomenon has been observed in several languages, in particular in English and French. We make use of a large variety of typologically different languages to check whether a co-effect between MAL and HCS can be brought to light.

In a second study [Mačutek et al., 2021], we aim at verifying MAL on a syntactic level. Indeed, the empirical evidence of the MAL tended to remained doubtful as soon as one moves from word to clause and sentence. To this end, we introduce a new syntactic unit : the linear dependecy segment. This work has given rise to a collaboration with colleagues from Ostrava (Czech Republic) and Bratislava (Slovakia), and the experimental work was carried out both on Czech and English.

After having experienced with properties extracted from a variety of treebanks and languages, we raised the question of how similar properties would evolve if we measured them from randomly created trees. This chapter thus ends with a comparison between syntactically grounded trees extracted from Chinese, English, French and Japanese treebanks and randomly created trees. This work is based on our publication [Courtin and Yan, 2019].

The idea behind this was to be able to infuse some knowledge into the structure induction, in particular to avoid inducing structures that seemed too random or strayed too far from the properties of syntactically grounded trees. We hoped that using

"universal laws" or properties of languages would enable us to reduce the search space to give a better plausibility to structures induced from raw corpora,but we did not go so far as to integrate it into any structure induction model.

6.1 Menzerath-Altmann & Heavy Constituent Shift

This section is based on the publication [Chen et al., 2022] which investigates the link between the Menzerath-Altmann Law and the Heavy Constituent Shift principle.

The Menzerath-Altmann Law, named after linguists Paul Menzerath and Gabriel Altmann, is a principle observed in linguistics. It states that within a linguistic construct (like a sentence, phrase, or word), the larger the construct, the shorter its constituents[1] tend to be. In other words, as a linguistic unit becomes larger, its subunits become proportionally smaller.

For instance, in the context of spoken language, this law suggests that longer sentences tend to have shorter words, and vice versa. This phenomenon reflects an efficiency principle in human language – as the overall structure expands, its components become more compact to aid in ease of communication and processing. The Menzerath-Altmann Law has been observed across various languages and is considered a fundamental principle in quantitative linguistics.

The Heavy Constituent Shift (HCS) is a syntactic phenomenon observed in several languages, most notably in English. It involves the reordering of sentence elements to place heavier constituents (i.e., longer or more complex phrases) later in the sentence. This shift is often driven by considerations of sentence rhythm, ease of processing, and information structure.

Obviously, there is a conceptual link between the Menzerath-Altmann Law and the Heavy Constituent Shift (HCS) in linguistics. Both of these concepts deal with the organization and distribution of information in linguistic structures, albeit in slightly different ways. In the following we will try to investigate whether and how both laws interact with respect to size of constituents.

[1] Here constituent is not used in the sense of the constituents from phrase grammar, but rather to mean subunit.

6.1.1 Related works on Menzerath-Altmann Law

Despite the success of the research on MAL, it seems that this powerful law is still not strongly connected to other traditional linguistic discussions that go beyond a mere application of the definitions to various linguistic units. We aim to address this point by linking MAL to the Heavy Constituent Shift (HCS) phenomenon and discuss their co-effect in human natural languages.

While the Menzerath-Altmann Law is one of the most discussed linguistic laws, the majority of related studies focus on verifying this law in certain linguistic constructs with different texts and languages, as well as trying to interpret its parameters [Altmann, 1980; Cramer, 2005a; Kelih, 2010a], for example, examining whether longer words (in the number of syllables) have shorter syllables (in the number of graphemes for phonemes), or if longer clauses (in the number of words) have shorter words (in the number of syllables) in different human languages, e.g [Menzerath, 1954] for German, [Kelih, 2010a] for Serbian. A small minority of studies discuss the language features that might influence the results of MAL, such as registers [Hou et al., 2019a; Xu and He, 2020a]. Meanwhile, MAL has started to transcend the field of quantitative linguistics and is also gaining attention from other disciplines, such as biology [Ferrer-I-Cancho and Forns, 2009; Li, 2012; Gustison et al., 2016].

6.1.2 Heavy Constituent Shift

HCS [Ross, 1967; Stallings et al., 1998] is a well-known phenomenon of syntax. Based on the concept of "heavy constituents"[2] that are composed of more words (and syllables) than "light constituents", it states that heavier constituents tend to be shifted[3] to the end of the clause. Here is the example 5.56 from [Ross, 1967, p. 306]:

(1) a. I'll **give** some **to my good friend from Akron**

b. I'll **give to my good friend from Akron** some.

[2]Here, unlike for MAL, constituents is to be interpreted in the sense given by phrase grammar.
[3]This concept of "shifting" comes from the framework of Transformational grammar .

In this example, the constituent 'to my good friend from Akron' has six words and is thus deemed heavier than the constituent 'some' which only has one word. Therefore, according to the HCS principle, we can expected the former constituent to be shifted to the end of the sentence, as in the further examples form the GUM [Zeldes, 2017a] English treebank of Universal Dependencies:[4]

(2) a. [...] I might capture them and **learn** from them **the secrets which the moon had brought upon the night**. (fiction_moon-9)

b. [...] the bartender will **recount** for the customer **the definition of the santorum neologism**. (interview_coktail-15)

c. [...] a scenery made of sand and rocks which **have** vaguely **the shape of a castle**. (voyage_guadeloupe_17)

d. [...] the adjustments and calculations **take** into account **the weighted nature of the data**. (academic_discrimination-51)

e. [...] the only candidate who **embodies** both physically and philosophically **the growing diversity of the commonwealth**. (interview_libertarian-11)

This commonly observed language phenomenon has been noted by several linguists before [Ross, 1967]. Here are three citations of French linguists from the 18th and 19th centuries given in [Kahane, 2020]:

> The [complements[5]] must be as close as possible to the governing word, which would not be the case if one were to put the longest [complement] first, which would move the shortest one too far away. [Buffier, 1709, p. 313]

> When several complements fall on the same word, it is necessary to put the shortest one first after the completed word; then the shortest of those

[4]These examples have been collected using a grew pattern described in the Appendix A.1.4.

[5]In the French tradition, complement means argument constituents as well as modifier constituents depending on the verb. This is the meaning we will follow here.

> that remain and so on until the longest of all, which must be the last. It is important for the clarity of the expression, cujus summa laus perspicuitas[6], to move what serves as the complement as little as possible away from a word. However, when several complements contribute to the determination of the same term, they cannot all follow it immediately; and all that remains is to bring the one that we are forced to keep away from it as close as possible to it: this is what we do by putting first the one which is the shortest, and keeping the longest for the end. [Beauzée, 1765, p. 7]

> When several complements fall on the same word, give the most concise form to the one immediately following the complete word and, as you go along, give the complements a more developed and extensive expression. [Weil, 1844, p. 97]

Note that HCS has first been observed for SVO languages such as French and English, where the complements are produced after the verb that governs them. The term heavy constituent shift has been coined by Ross in the framework of transformational grammar, with the idea that heavy constituents were shifted from some initial position to the final place. For Buffier and Beauzée, light complements must simply be produced before heavy complements. Weil introduced an additional idea: If you want to produce two complements in a given order, make the second one heavier than the first. In other words, it is not because a complement is heavy that you put it in the second place, it is because it is in the second place that you make it heavier (and, again, there is absolutely no shift in this framing of the phenomenon). There are still debates concerning the definition of 'heavy'. Although the theoretical discussion is valuable, for the empirical data analysis in this study, we take the operational definition of 'heavy', namely, having more words.

[6] 'whose highest praise is clarity' (a variation of the famous quote from Quintilian's The Orator's Education stating that the oratory's "basic virtue is clarity").

6.1.3 The Co-effect Hypothesis: Combining Heavy Constituent Shift and Menzerath-Altmann Law

Both HCS and MAL are associated with the size of constituents. This suggests that there are probably interactions between HCS and MAL. From their mathematical formalization, we show that a hypothesis based on these two premises.

To be more specific, we investigate and compare the size of different constituents in two types of clauses that have either one or two complements to the right of the word X[7]:

(3) ~XAB (the word X has two complements A and B to its right, and A precedes B)

(4) ~XC (the word X has only one complement C to its right)

We will focus on words X, when it is the verbal head of a clause. *a, b, c* corresponds to the size (the number of words) of the constituents A, B, and C.

First, according to MAL, we can expect that the average size of two complements (case 1.) is smaller than the size of the unique element (case 2.):

$$(a + b)/2 < c \tag{6.1}$$

And then, according to HCS, we also expect that B is heavier than A:

$$a < b \tag{6.2}$$

[7]Note that this simplified definition allows any number of dependents to the left of X and does not take into account the presence and size of any elements to the left of X which might be part of the projection of X. We will see in Section 6 that taking into account possible elements to the left does not significantly alter the results.

When we combine Equations 6.1 and 6.2, we can get that[8]:

$$\begin{aligned}(a+a)/2 < (a+b)/2 < c \\ \Rightarrow a < c\end{aligned} \tag{6.3}$$

We thus presume that we should observe $a < c$ in empirical data, and if our hypothesis is validated, this language phenomenon can be seen as the co-effect of MAL and HCS in human natural languages.

6.1.4 Methodology

Constituents, from the viewpoint of dependency syntax, are projections of a node in the dependency tree, that is the subtree headed by the node in question.

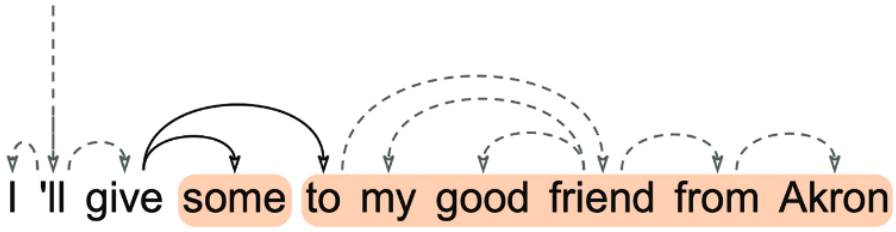


Figure 6.1: Dependency tree of the sentence *I'll give some to my good friend from Akron'.*

As we can see in Fig. 6.1, there are two dependencies (bold lines) that fall on the right side of the verb *give*. Each branch heads one constituent. These two constituents are the two complements of *give*. We will consider here that the size of a constituent is determined by the numbers of words it contains. The two complements of *give* on its right, have respectively size one (for *some*) and size six (for *to my good friend from Akron*).

What we are investigating in this paper are two types of clauses, namely, ~XAB (Ex. 3) and ~XC (Ex. 4). In which, ~ represents left branches of the tree that are not

[8]Because a < b we can replace the *b* term in Equation 6.1 with *a* which gives us (a + a) / 2, which simplifies to *a*.

taken into consideration here. For instance, the following three sentences would all be considered as ~XAB clauses:

(5) a. I definitely give some to my good friend from Akron. (including two left tree branches)

b. I give some to my good friend from Akron. (including one left tree branches)

c. Give some to my good friend from Akron. (including zero left tree branches)

And all the following three sentences would be considered as ~XC type clauses:

(6) a. I probably did the job. (including two left tree branches)

b. I did the job. (including one left tree branches)

c. Do the job. (including zero left tree branches)

While we pay no attention to the left branches of the tree, we imposed strict restrictions of the right branches, only taking into account when there was either one or two branches. For instance, the following clauses would not be considered in our analysis:

(7) a. I tried. (including zero right tree branches)

b. I told her the truth eventually. (including three right tree branches)

To test our hypothesis, we chose the Surface-Syntactic version (SUD 2.7, described in [Gerdes, 2018; Gerdes et al., 2019a]) of the Universal Dependencies treebank set [Nivre et al., 2016]. The dataset includes 183 treebanks in 104 languages from various typological groups, with a majority of Indo-European languages.

For some languages, several treebanks have been developed. In this pilot study, we are more interested in the general picture, and we combine all the treebanks of a language into one collective treebank. Therefore, we take global measures across all trees for each language.

After clearly defining all the conditions, we first filter out ~XAB type and ~XC type clauses from each dependency treebank we study. We only look at X that are verbs and A, B, C that are subjects or complements. More specifically, we only look at

the complements with the dependency tag *subj* (subjects), *comp* (complements), *mod* (modifiers), or *udep* (an underspecified label that subsumes both *comp* and *mod*).[9] For each clause we collect, we compute the size of the constituents A, B or C and store them as a, b, or c, and then we calculate the mean value of all a, b, and c on the whole treebank (or treebanks if several of them are available for the language). By comparing the mean value of a and c, we can then either accept or reject our hypothesis.

For the numbers of ~XAB and ~XC clauses in each language, see Tab. A.1 in the Appendix.

6.1.5 Results

We filter out languages with very sparse data for which we have less than 20 measures of a or c. This reduces the number of languages to 80. Our results in Fig. 6.2 show that all languages appear above the diagonal. It reflects that our hypothesis $a < c$ is verified across these typologically different languages.

The colors and shapes in Fig. 6.2 represent rough language groups :

- Indo-European languages: triangles
  - Indo-European-Romance: brown
  - Indo-European-Baltoslavic: purple
  - Indo-European-Germanic: olive
  - Other Indo-European: blue
- Sino-Austronesian: green stars
- Agglutinating languages: red plus signs
- Other languages: black squares

The actual values of a, b and c in each language are presented in Tab. A.1 in the Appendix.

[9] Of course we also take into account all possible extensions of these tags, such as *comp:obj* (direct complement), *compl:obl* (oblique complement) etc.

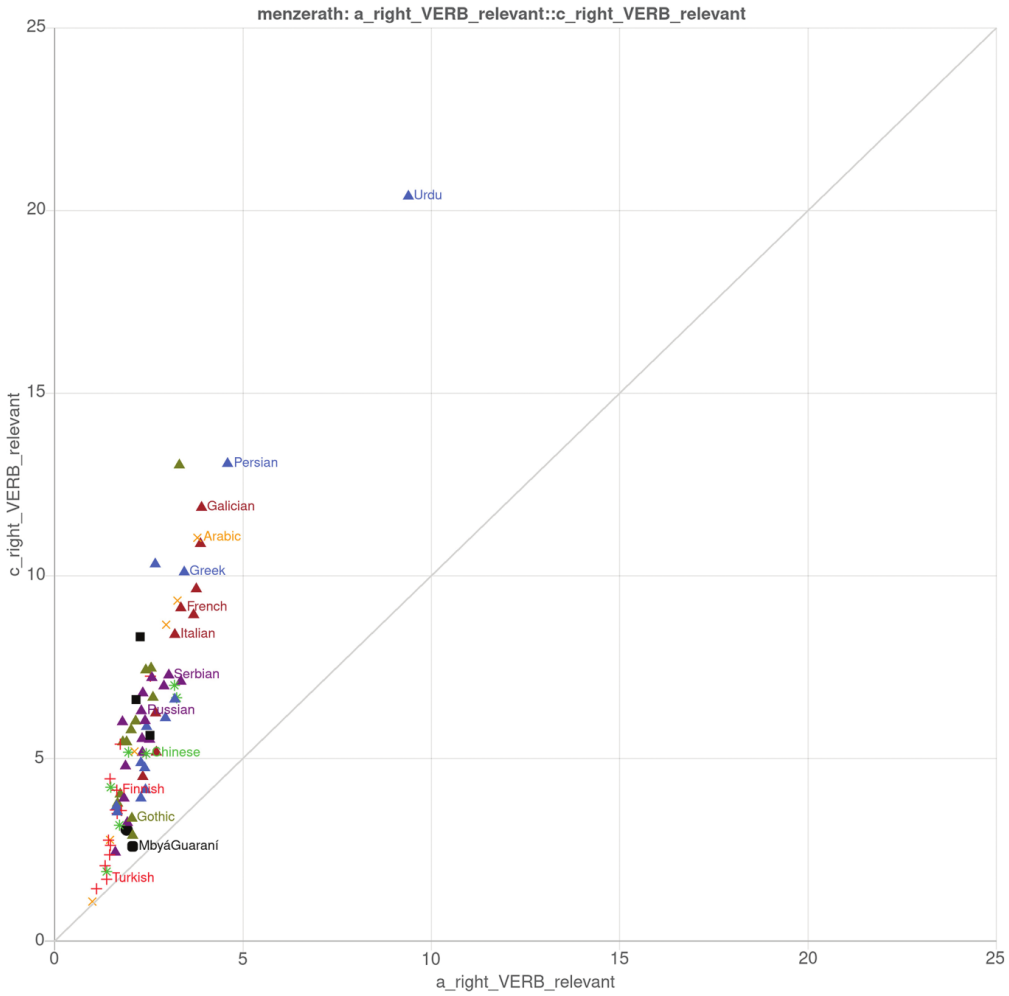


Figure 6.2: The average size c of C constituents (y-axis) is bigger than the average size a of A constituents (x-axis) across the 80 languages of SUD 2.7 where we have at least 20 occurrences of corresponding structures.

6.1.6 Conclusion

Our results show that our hypothesis is valid across the complete set of typologically diverse languages that are present in the SUD treebanks. The co-effect of Menzerath-Altmann Law and Heavy Constituent Shift appears to be a regular universal.

Our pilot study shows that by making use of the recently available coherently annotated multilingual SUD, we can bridge MAL with traditional linguistic discussions such as the HCS, and therefore expand the scope of studies on MAL.

Meanwhile, there are still various details to be investigated in the future. For

example, we need to explore what happens to the left of the governor, in particular for verb-final languages. It might also be worthwhile to verify the measures for all kinds of clauses, not only the clauses that have a verbal head.

Note also that the data is very unevenly distributed based on the language. Some languages, such as German, English, Czech, Arabic, etc., have large treebanks, while for other languages the treebanks are very limited in sizes. Moreover, the distribution of the data is not at all random with respect to typology, with some language families being surrepresented. This is a sticking point for rigourous quantitative typological findings.

We still have to evaluate in the future how much the sample size would affect the results. Also, even for the same language, treebanks annotated by different teams can vary from each other. In the future, we would like to consider the effect of fusing treebanks together. Last but certainly not least, we can gradually ease the control factors, reduce the constraints for selecting samples, to test the boundary conditions of the co-effect phenomenon.

6.2 Adapting MAL to more syntactically defined units

As we have seen in the previous section, the Menzerath-Altmann law states that there is an inverse proportionality between sizes of language units and their constituents (i.e., longer language units are composed of shorter constituents, and vice versa). The validity of this law was confirmed many times for the relation between lengths of a word and its syllables. We have shown in the section above that there is a link between MAL and the heavy constituent shift. However, the relation between lengths of sentences (measured in clauses) and clauses (measured in words) is problematic. In section 6.2.1, a new language unit – linear dependency segment – is introduced with the motivation to avoid some problems connected to the Menzerath-Altmann law on the syntactic level. The new unit is an intermediate between clause and word and

its definition takes into account both the linearity of language and the dependency syntactic structure.

According to MAL, longer units which are higher in the hierarchy (constructs) consist of shorter lower units (constituents). The formulation of the MAL developed from a verbal one (the longer the word, the shorter on average its syllables; see [Menzerath, 1954] to mathematical formula 6.4 derived by [Altmann, 1980].

$$y(x) = ax^b e{-}cx \tag{6.4}$$

In formula 6.4, $y(x)$ is the mean size of constituents in the construct of size x; a, b and c are parameters. Very often a simpler formula 6.5 is used, which is a special case of 6.4 for $c = 0$.

$$y(x) = ax^b \tag{6.5}$$

The MAL was first observed as the relation between word length in syllables and either syllable length in phonemes [Menzerath, 1954], or syllable duration in time [Geršić and Altmann, 1980]. The validity of the MAL at this lowest level has been explored in many languages (see e.g. [Cramer, 2005b], and references therein [Kelih, 2010b, 2012; Mikros and Milička, 2014; Ján Mačutek and Koščová, 2019]).

However, two fundamental problems emerge when one goes higher in the hierarchy of language units. First, it was assumed that the upper neighbours of word are clause and sentence. Although several papers in the 1980s [Köhler, 1982; Heups, 1983; Schwibbe, 1984; Teupenhayn and Altmann, 1984] claim that the relation between sentence length in clauses and clause length in words abides by the MAL, more recent results are far from clear. Thus, [Kułacka, 2010; Chen and Liu, 2019; Xu and He, 2020b] confirm the older results, while data analysed by [Kułacka and Mačutek, 2007; Benešová and Čech, 2015; Renkui Hou and Liu, 2017] display a Menzerathian ten-

dency, but they cannot be fitted by function 6.4 sufficiently well.[10] On the other hand, data presented by [Buk and Rovenchak, 2008] and by [Andres and Benešová, 2012] do not confirm to the MAL.[11] Curiously enough, [Andres and Benešová, 2012] and [Hou et al., 2019b] are, to our best knowledge, the only two papers which focus also on the relation between lengths of clause (in words) and word (in syllables).[12] This relation, again, cannot be modelled by the MAL. To put it mildly, the empirical evidence of the MAL, especially in form of function 6.5, is doubtful as soon as we move from word to clause and sentence.

[Mačutek et al., 2017] tried to measure clause length in syntactic phrases which are directly dependent on the predicate of the main clause (with phrase length being measured in words). The MAL in form 6.5 achieved a very good fit. The phrase thus became a candidate for an intermediate language unit between word and clause. It must be noted that only main clauses were analysed, and only one Czech used.

Second, although the linguistic interpretation of the parameters of model 6.4 is still not known, it has been suggested that the MAL has something to do with short term memory [Köhler, 1989; Grzybek, 2013] see also [Yngve, 1960]).[13] According to [Miller, 1956], the capacity of short-term memory is approximately seven units. With the exception of polysynthetic languages, words only seldom contain more than seven syllables (or morphemes[14]), and the same is true for sentence length in clauses. However, clauses longer than seven words are not so rare – the mean clause length in the papers cited above is often somewhere near 10, see e.g. [Köhler, 1982; Heups, 1983; Teupenhayn and Altmann, 1984].

[10] See [Mačutek and Wimmer, 2013] for an overview of goodness-of-fit criteria usually used in quantitative linguistics.

[11] Admittedly, these papers do not follow the same methodology. In most of them, either finite verbs or punctuation marks (comma and semicolon) to determine sentence length in clauses.

[12] [Hou et al., 2019b] measure word length in characters, but in written Chinese there is almost one-to-one correspondence between characters and syllables.

[13] [Torre et al., 2019] present an attempt to explain the origin of the MAL in spoken language at the level of words and syllables as a consequence of human physiology (in particular the necessity to breathe). These two tentative explanations of the MAL do not exclude each other; rather, both factors (pauses caused by breathing and a limited capacity of short-term memory) are likely to contribute to the shortening of constituents in longer constructs.

[14] See [Pelegrinová et al., 2021] and references therein for the MAL as the relation between word length in morphemes and morpheme length in phonemes.

The phrase used by [Mačutek et al., 2017] faces the same problem, e.g. there are 7,125 clauses which contain only one phrase (this represents more than 12% of the data), and their mean length in words is 9.47 (which means that there are many phrases longer than 9.47). In addition, consider a sentence consisting only of a single predicate (e.g. Czech sentence *Prší* "*It rains*"). Such a sentence contains only one clause of length zero (because there is nothing directly dependent on the predicate of the clause), and phrase length cannot be determined at all, as there is no phrase in the sense of the phrase definition from [Mačutek et al., 2017]. If the definition is modified so that phrase includes also the predicate, the question arises how to determine phrase length in clauses consisting of at least two phrases (such as e.g. in Czech sentence *Petr miluje Marii* "*Peter loves Mary*"). If the predicate is a part of the phrases, it appears more than once in all calculations. Regardless of these methodological difficulties, the use of the phrase as an intermediate language unit also has the drawback of neglecting the linearity of language.

To avoid the abovementioned problems, we suggest another approach, namely, a new language unit between word and clause: the linear dependency segment. Its definition combines both the linear and hierarchical dependency structure of sentence. We focus on the question of whether this new unit behaves according to the MAL.

6.2.1 Linear dependency segment

We define the linear dependency segment (LDS henceforward) as the longest possible sequence of words (belonging to the same clause[15]) in which all linear neighbours (i.e. words adjacent in a sentence) are also syntactic neighbours (i.e. are connected by an edge in the syntactic dependency tree which represents the sentence). Figure 6.3 presents the dependency tree of sentence *"This black book on the table costs twenty euros, which is too much for me"*. The two fully lined grey boxes represent the clauses, while the smaller grey boxes with dashed lines represent the linear dependency

[15]We use the definition of clause from Prague Dependency Treebank 3.0 (https://ufal.mff.cuni.cz/pdt3.0/documentation#__RefHeading__42_1200879062), according to which "(a) clause typically corresponds to a single proposition expressed by a finite verb and all its arguments and modifiers (unless they constitute clauses of their own)".

segments.

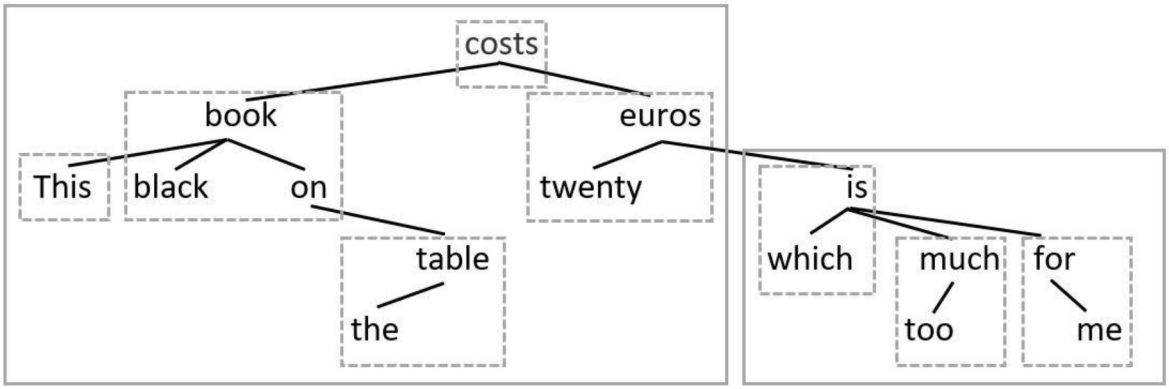


Figure 6.3: Dependency tree of sentence *"This black book on the table costs twenty euros, which is too much for me"*

Consider the first clause in the sentence. Its first word, *"This"*, is syntactically linked with *"book"*, but these two words are not linear neighbours. Therefore, the first LDS is [This]. Next, the second word, *"black"*, is syntactically linked with "book", which is also its linear neighbour, and the third and the fourth words, *"book"* and *"on"*, are again both linear and syntactic neighbours. Here the segment ends, because the next word, *"the"*, is not syntactically linked with *"on"*. Examining the whole clause we obtain the LDSs [This][black book on][the table][costs][twenty euros]. Similarly, the second clause in this sentence has LDSs [which is][too much][for me]. We remind that we define the LDSs as units of which clauses are composed, i.e. a LDS cannot go over a clause boundary.

The definition is good in the sense that every clause can be unambiguously divided into LDSs, and that the intersection of two different LDSs is the empty set (i.e. every word in a clause belongs to one and only one LDS).

From the MAL point of view, clauses are the constructs here, and LDSs the constituents (which, in turn, is a construct itself, with words being its constituents). Therefore according to the MAL, we expect that longer sentences (measured in the number of clauses) contain shorter clauses (measured in the number of LDSs). This expectation is based on the fact that dependency links which do not respect the lin-

earity of a sentence are more difficult to process.[16] The same is true for a sentence with many clauses. The MAL does not allow sentences to become too complex, as it "forces" clauses in long sentences (i.e. in ones which contain many clauses) to become shorter (i.e. to be composed of fewer LDSs). Fewer LDSs mean that there are fewer dependency distances (as defined by [Liu, 2008, p. 164] longer than one (as all dependency distances within one LDS are minimal, i.e. equal to one).

Provided that the MAL is valid as a model for the relation between lengths of sentences and clauses, a sentence can be composed either of more clauses which are shorter in terms of LDSs (which means that they are syntactically simpler[17]), or of fewer clauses which are "allowed" to contain more LDSs (and consequently to be syntactically more complex).

6.2.2 Experimental results

For the analysis, we used two Czech treebanks, the Czech-PDT UD[18] and the FicTree [Jelínek, 2017]. The code used to segment the trees into clauses and LDS is available at `https://github.com/marinecourtin/linear_dependency_segmentation`.

The treebanks were converted to the Surface Syntactic Universal Dependencies (SUD) annotation scheme [Gerdes, 2018]. The use of the Universal Dependency annotation scheme [de Marneffe et al., 2021] was also considered. However, we settled on the SUD approach because it is based on surface-syntactic distributional criteria that fit the nature of our analysis better than the Universal Dependency approach which is based on "a mixture of semantic and syntactic motivations" [Osborne and Gerdes, 2019b].

The Czech-PDT UD consists of 87,913 Czech sentences from non-abbreviated newspaper, business and popular scientific journal articles published from 1991 to 1995. The FicTree consists of 12,760 sentences from Czech literary works published between 1991

[16] The idea that dependency distance in language is shorter than a random baseline can be traced back to [Liu, 2008].

[17] If we consider the extreme case, a clause consisting of only one LDS either contains only one word, or it reaches the minimum of dependency distance (in such a clause all dependency distances are equal to one).

[18] `https://universaldependencies.org/treebanks/cs_pdt/index.html`

and 2007. The treebanks were also merged and treated as one whole in which different genres are represented. Sentences without a predicate (especially titles of newspaper articles) were removed. We thus analysed altogether 86,266 sentences.

As we study the relation between sentence length and the mean clause length, the number of clauses from which the mean is calculated cannot be too low if the result should be robust. We decided to take into account sentence lengths with frequencies which make at least 0.1% of our language material. We thus disregarded sentences containing more than eight clauses (together 76 sentences, i.e. 0.09%). Very complicated structures, such as several clauses placed in brackets, clauses separated by a colon, or citations, are typical for these long sentences. The possibility to check thoroughly the sentences which do not conform to the MAL was also the reason why we focussed only on Czech treebanks for this study – one of the coauthors is a native Czech speaker. It is obvious that our choice substantially limits the scope of this paper, but given that it is the first attempt to study the LDS as a language unit, we prefer this more careful approach.

The relation between sentence length in clauses and the mean clause length measured in LDSs is presented in Figure 6.4.

	merged			PDT			FicTree		
SL	f	rf	MCL	f	rf	MCL	f	rf	MCL
1	36559	0.424	5.02	32002	0.428	5.30	4557	0.396	3.03
2	27735	0.321	3.93	24121	0.323	4.10	3614	0.314	2.82
3	13463	0.156	3.44	11605	0.155	3.54	1858	0.162	2.79
4	5416	0.063	3.17	4537	0.061	3.25	879	0.076	2.77
5	1962	0.023	3.00	1616	0.022	3.07	346	0.030	2.69
6	727	0.008	2.94	580	0.008	3.02	147	0.013	2.64
7	236	0.003	2.84	188	0.003	2.85	48	0.004	2.82
8	92	0.001	2.79	69	0.001	2.93	23	0.002	2.36

Figure 6.4: The MAL in Czech dependency treebanks (SL - sentence length in clauses, f, rf - frequencies and relative frequencies[19]of sentence lengths, MCL – the mean clause length in LDSs).

[19]The relative frequencies do not sum to one, because sentences containing more than eight clauses were disregarded.

The MAL in form 6.5 fits the data from the merged treebanks very well[20], with $R^2 = 0.9836 (a = 4.918, b = -0.296)$.[21] The data and the graph of the function can be seen in Figure 6.5.

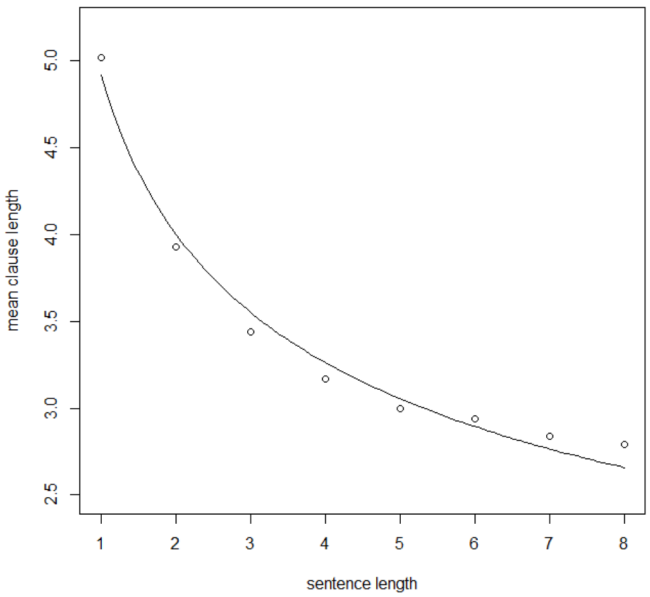


Figure 6.5: The MAL modelled by function $y(x) = ax^b$ as the relation between sentence length and the mean clause length

The value of parameter a is very close to the mean clause length (measured in the number of LDSs) in sentences consisting of only one clause. If we use this value, i.e. if we set $a = 5.02$ in formula 6.5, we obtain $b = -0.309$ and $R^2 = 0.9803$, which is still a very good fit. We thus have a very clear interpretation of the parameter a.[22] As for parameter b, its linguistic interpretation remains an open question.

In both PDT and FIC treebanks, the decreasing tendency of the mean clause length can be observed. While the fit of function 6.5 remains very good ($R^2 = 0.9739$) for

[20]The most common rule of thumb in quantitative linguistics is to consider the goodness-of-fit of a model satisfactory if the value of the determination coefficient R^2 is higher than 0.9, see [Mačutek and Wimmer, 2013].

[21]The fit also remains satisfactory if other options on how to deal with low frequency construct lengths are applied. If all construct lengths with a frequency of at least 10 are used in the computations (see [Mačutek and Rovenchak, 2011]), we have $R^2 = 0.9353$, and if we pool low-frequency construct lengths (i.e. sentence which contain more than eight clauses in our case) and compute the weighted mean of clause lengths (see e.g. [Pelegrinová et al., 2021]), we obtain $R^2 = 0.9649$.

[22]The interpretation of parameter a of the MAL in form 6.5 as the mean length of constituents of the shortest constructs is not specific to language units analysed in this paper – e.g. [Kelih, 2010b] uses the same approach when investigating the relations between lengths of words and syllables.

PDT, it is much worse ($R^2 = 0.6148$) for the data from the FicTree treebank. However, this is caused by an irregular behaviour of the mean clause length of the two highest values of sentence length, which occur with relatively low frequencies (moreover, the FicTree treebank is much smaller than PDT), and an overall decreasing tendency can also be seen also in the results from this treebank.

The two treebanks differ also in the mean values of the shortest sentences (i.e. the ones containing only one clause). Most likely, it is a consequence of different sentence length distributions in the treebanks (the mean values are 1.97 for PDT and 2.11 for FicTree; see also relative frequencies of sentence lengths in Figure 6.4). Longer sentences in FicTree are composed of shorter LDSs. We remind the reader that the treebanks consist of journalistic texts (PDT) and fiction (FicTree), and that sentence length has been shown to depend on the genre of texts (see e.g. [Kelih et al., 2006; Xu and He, 2020b]).

6.2.3 Conclusion

The results indicate that, at least tentatively, the LDS can be considered a meaningful linguistic unit which allows to model the MAL on the syntactic level. The LDS avoids the problems frequently encountered when one measures clause length in the number of words the clause contains. From the theoretical point of view, it is important that clause length measured in LDSs correspond with the capacity of short-term memory[23], which is one of the theoretical explanations of the MAL. Furthermore, we emphasize that the definition of the LDS takes into account both the linearity of language and the dependency syntactic structure.

Naturally, this paper is only a pilot study, very limited in its scope, and data from many more typologically diverse languages must be analysed before the LDS can establish itself firmly among more traditional language units. Specifically with

[23] [Miller, 1956] claims that the capacity is roughly seven (although there are also other opinions). Clause length determined in the number of the LDSs only rarely exceeds this value, while clause length in words can be, naturally, (much) higher. Similarly, phrases used by [Mačutek et al., 2017] contain more words than LDSs; in addition, the methodology from that paper allows to analyse only main clauses.

respect to the MAL, LDSs have yet to be investigated as a construct, which could be done by looking at the lengths of LDSs (in words) and their relationship with word length (in syllables or morphemes). In addition, if the LDS turns out to be a suitable linguistic unit, it would be interesting to look at how its frequencies and length follow distribution laws commonly used to model similar language properties (i.e. a Zipf-like distribution for LDS frequencies, and a Poisson-like distribution for LDS lengths, see e.g. [Popescu, 2009; Grzybek, 2007], respectively).

A possible correspondence between LDSs and dependency distance minimization deserves a closer inspection. While there is a strong evidence that words which are syntactically linked are close to each other also with respect to the linear order of the sentence (see e.g. [Liu, 2008; Ferrer-i Cancho and Liu, 2014; Futrell et al., 2015]), short sentences are quite likely to not follow this trend [Ferrer-i Cancho and Gómez-Rodríguez, 2021]. Although sentence length in these studies is expressed in the number of words they contain (as opposed to our approach where sentence length is expressed in number of clauses), we can suppose that short sentences mostly contain one or two clauses. The MAL predicts that clauses in short sentences are composed of relatively many LDSs, which means that there must be relatively many dependency distances with values more than one. The findings from [Ferrer-i Cancho and Gómez-Rodríguez, 2021] and from this paper thus support each other.

6.3 Comparing syntactically grounded and artificial trees

This sections is centered around two main contributions : the first one consists in introducing several procedures for generating random dependency trees with constraints; we later use these artificial trees to compare their properties with the properties of syntactic trees (i.e trees extracted from treebanks, representing the structure of natural languages) and analyse the relationships between these properties in the syntactic and artificial settings in order to find out which relationships are formally constrained

and which are linguistically motivated.

We take into consideration five metrics: tree length, height, maximum arity, mean dependency distance and mean flux weight, and also look into the distribution of local configurations of nodes. This analysis is based on UD treebanks (version 2.3, [Nivre et al., 2018]) for four languages: Chinese, English, French and Japanese. The source code is available at `https://github.com/marinecourtin/Alearbres`.

We are interested in looking at the linguistic constraints on syntactic dependency trees to understand what makes certain structures plausible while others are not so plausible. To effectively do this kind of work, we need to observe syntactic trees that are the results of linguistic analysis to see what this population looks like. Similar work has been done for example by [Jiang and Liu, 2015] on the relation between sentence length, dependency distance and dependency direction.

But observing only syntactic trees has its limits : we cannot see what is special about them and their properties, and we cannot distinguish the effects of the various constraints that affect them. We can only observe the structures that are the result of all these constraints and their interactions. On the other hand, if we start from a blank canvas, randomly generated trees, and incrementally add constraints on these trees, we might be able to study one by one the effects of each constraint, and to progressively add constraints that get us closer to syntactic trees.

Using artificially generated trees can also be insightful to determine which constraints are formally motivated (they are a result of the mathematical structure of the tree), and which constraints are linguistically or cognitively motivated. Research in the line of [Gildea and Temperley, 2010] who have used random and optimal linearisations to study dependency length and its varying degrees of minimisation can help us to discover constraints that would be helpful to explain why we only find a small subset of all potential trees in syntactic analyses on real data.

Our objective is therefore twofold: first we want to see how different properties of syntactic dependency trees correlate, in particular properties that are related to syntactic complexity such as height, mean dependency distance and mean flux weight, then we want to find out if these properties can allow us to distinguish between artificial

dependency trees (trees that have been manipulated using random components and constraints), and dependency trees from real data.

6.3.1 Looking into the properties of syntactic dependency trees

6.3.1.1 Features

In this work, we use the five following metrics to analyse the properties of dependency trees:

Feature name	Description
Length	Number of nodes in the tree
Height	Number of edges between the root and its deepest leaf
Maximum arity	Maximum number of dependents of a node
Mean Dependency Distance (MDD)	short description
Mean flux weight	short description

Table 6.1: Tree-based metrics

We chose these properties because we believe that they all interfere in linearization strategies, that is how words are ordered in sentences, and the effects of those linearisation strategies. Recently, there have been many quantitative works [Futrell et al., 2015; Liu, 2008] that have focused on dependency length and its minimisation across many natural languages. In complement to these linear properties we also use *flux weight*, a metric proposed by [Kahane et al., 2017b] which captures the level of nestedness of a syntactic construction (the more nested the construction is, the higher its weight in terms of dependency flux).

In addition to these tree-based metrics, we propose to look at local configurations inside the dependency trees. To look at these configurations, we extract and compare the proportion of all potential configurations of bigrams (two successive nodes) and trigrams (three successive nodes). For bigrams, we have three possible configurations: a -> b which indicates that a and b are linked with a relation on the right, a <- b which indicates that a and b are linked with a relation on the left, and a $\diamond$ b, which

indicates that a and b are not linked by a dependency. For trigram configurations (a, b, c), there are many more possible configuration, 25 in total. There are projective configurations like: a -> b -> c, (a -> b) & (a -> c), (a -> c) and (b <- c), but also non-projective cases like: a <- c and b -> c.

6.3.1.2 Hypotheses

In this section, we describe some of our hypotheses concerning the relationship between our selected properties. First, we expect to find that tree length is positively correlated with other properties. As the number of nodes increases, the number of possible trees increases including more complex trees with longer dependencies (which would increase the mean dependency distance) and more nestedness (which would result in a higher mean flux weight). The relationship with maximum arity is less clear, as there could be an upper limit, which would make the relation between both of these properties non-linear.

We are also particularly interested in the relationship between mean dependency distance and mean flux weight. An increase in nestedness is likely to result in more descendants being placed between a governor and its direct dependants, which would mean an overall increase in mean dependency distance.

For local configurations, we know that in natural trees, most of the dependencies occur between neighbours, see for example [Liu, 2008], the proportion varying depending on the language. It will be interesting to see how much that is still the case in the different random treebanks, depending on the added constraints.

For trigrams of nodes we are interested in the distribution of four groups of configurations that represent four different linearization strategies: *chain* subtrees that introduce more height in the dependency tree in with both dependants in the same direction, *balanced* subtrees that alternate dependants on both sides of the governor, *zigzag* subtrees which are similar to chains but with the second dependent going in the opposite direction as the first one, and *bouquet* subtrees where the two dependants are linked to the same governor (see examples in Figure A.3.1 in the Appendix).

If one group of configurations is preferred in syntactic trees compared to artificial

ones, it could indicate that there exists some linguistic and/or cognitive constraints that make the configuration more likely to appear. We are also interested in the hypothesis advanced by [Temperley, 2008] who proposes that languages that strongly favour head-initial or head-final dependencies will still tend to have some short phrases going in the opposite direction, which could constitute a way of limiting dependency distances.

6.3.2 Random tree generation with constraints

In this section we propose to look at various procedures which can be used to generate random dependency trees with constraints. We distinguish two steps in the dependency tree generation process : the generation of the *unordered structure*, and the generation of the *linearisation of the nodes*. Throughout this generation process, we limited ourselves to projective trees. In order to compare the properties of natural and random trees we used 3 different tree generating algorithm, to which we assign the following names : original_random, original_optimal and random_random.

original_random The first algorithm samples an unordered dependency structure from a treebank (i.e the original structure), and generates a random projective linearisation for it. The procedure is explained below, using 6.6 as our example for the original structure :

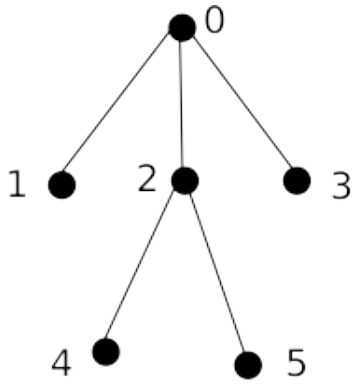


Figure 6.6: Unordered tree

- We start the linearisation at the root.
- Then, we select its dependent nodes [1,2,3] and randomly order them, which gives us [2,1,3].
- We select their direction at random, which gives us ["left", "left", "right"], and the linearisation steps [0], [20], [120], [1203].
- We repeat steps 1 through 2 until every node has been linearized, which gives us (for example) [124503].

original_optimal The second algorithm also samples an unordered dependency structure from a treebank, but instead of generating a simple projective linearisation, a second constraint is added to minimise dependency distances inside the dependency tree. The procedure is adapted from [Temperley, 2008] : to minimise dependency distances in a projective setting, dependents should be linearised alternatively on opposing sides of the governor, with the smallest dependent nodes (i.e those that are the head of the smallest subtrees) linearised first. Using the same structure unordered tree as in Figure 6.6 the procedure is done through the following steps :

1. We start the linearisation at the root.
2. Then, we select its dependent nodes [1,2,3] and order them in order of their decreasing number of descendant nodes, which gives us [1,3,2].
3. We select a first direction at random, for example "left", and order these nodes alternating between left and right, which gives us these linearisation steps [0], [10], [103], [2103].
4. We repeat steps 1 through 2 until every node has been linearised, which gives us for example [425103].

random_random The third algorithm is the only one to implement two random steps : first generate a completely random structure, then linearise it following the

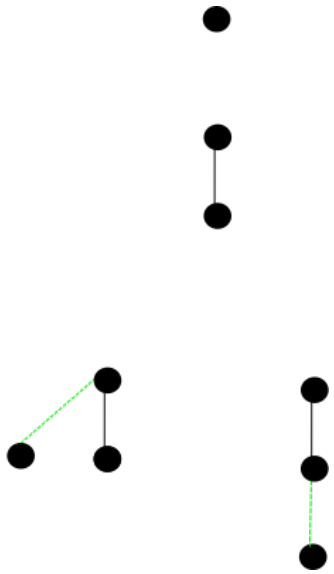


Figure 6.7: Random tree generation

same procedure as in algorithm *original_random*. The unordered structure generation step is described in Figure 6.7.

1. We start the generation process with a single node
2. We introduce a new node and randomly sample its governor. For now, since there is only one potential governor, the edge has a probability of 1.
3. We introduce a new node and randomly draw its governor. There are two potential governors which gives us a probability 0.5 of drawing the node 0 and the same probability for the node 1. These candidate edges are drawn in green on the graph.
4. We repeat this last step until all nodes have been sampled and attached to their governor[24].

These tree generation algorithms are only some of the many possible algorithms that could be implemented, but they us give us tools to analyze how different genera-

[24]Note that this algorithm gives us a uniform probability on derivations, but that some derived trees are more probable than others, for example if the length of the tree is 4 we only have 1 derivation to obtain a tree of height 4, and 2 derivations to obtain a tree with 2 dependent on the root and 1 on one of these dependents.

tion strategies will affect the properties of the generated trees, as we incorporate more and more constraints into the two generation steps.

Other algorithms have been proposed for artificial tree generation. [Alemany-Puig et al., 2021] propose the Linear Arrangement Library, which allows for the generation of both random baselines and minimum baselines (in the sense of minimising for example for dependency distance). In addition, their method provides the possibility to exhaustively generate trees which extends the kind of hypotheses that can be tested using artificial trees. They also implement varying degrees of projectivity constraints.

It is easy to see how such generation procedures could be extended, by adding constraints during the generation process. For instance, we could introduce a probability of creating a head-final edge, to produce trees that resemble more the trees of a head-final language like Japanese. For the unordered structure generation, constraints could be introduced to limit length, arity, height or any other number of features.We need to distinguish constraints that happen during the unordered structure generation step and constraints that have to do with linearisation, like constraints on dependency distances and on flux weights.

One question that remains unanswered concerns the ordering of the two steps (unordered structure generation and linearisation generation) and whether it has an influence on the obtained results. So far we have only implemented the full generation starting with the generation of the unordered structure and then moving on to the linearisation[25], but it would be interesting to try in the other direction, starting with a sequence of nodes, and then randomly producing a structure for it. Depending on the added constraints, the order of the steps could maybe have its importance and introduce biases towards some types of structures.

[25]Similar to a synthesis approach as described in Meaning-Text-Theory [Melčuk, 1988]

6.3.3 Results and discussion

6.3.3.1 Correlation between properties

For each pair of properties presented in Section 6.3.1.1 we measured the Pearson correlation coefficient to find out the extent to which the relationship between these variables can be linearly captured. We looked into these results for the syntactic treebanks (*original*) and the artificial ones (*original_random*, *random_random*and *original_optimal*). Tables presenting the full results can be found in the appendix of [Courtin and Yan, 2019].

Mean dependency distance and mean flux weight. Based on these results, we notice that mean dependency distance and mean flux weight are overall the most correlated properties with values ranging from 0.70 (jp_pud, original) to 0.95 (fr_partut, original_ optimal). This can be explained by the fact that mean flux weight increases as the number of disjoint dependencies increases [26], which in turn tends to create longer dependencies than structure with few disjoint dependencies. An interesting observation about this correlation is that it is intensified in all the artificial treebanks, and is the strongest in the *original _optimal* version. Introducing a dependency distance minimization constraint will favour shorter dependencies, which provides less opportunities for configurations that introduce disjoint flux. Therefore the mean flux weight will also decrease.

Length and height If we look at the the correlation between length and height, we find that it is strong in the original structures (0.78 correlation) as well as in the random ones (0.71 correlation in *random_ random*, which is the only format in which the height of the tree is affected by the transformation). This means that the relationship between these two properties is not motivated by linguistic factors only. From a mathematical point of view, longer sentences have the potential to introduce more hierarchy which increases the height. Thus, there is a correlation between these

[26] A set of dependencies is said to be disjoint if the dependencies do not share any nodes (see Fig. A.5 for a graphical representation).

two properties regardless of whether the structure is syntactic or random in nature. [Zhang and Liu, 2018] have proposed that the relationship between these two properties in natural treebanks of English and Chinese can be described by a power law. Further examination could tell us if it is also the case for randomly generated trees, or if the relationship is better modelled by another type of function.

Mean dependency distance and height. We also find quite strong correlations between mean dependency distance and height in the artificial treebanks (0.76, 0.79, 0.72 respectively for *original_ random*, *original_optimal* and *random_random*) while this correlation is less important for the syntactic trees (0.46). It is quite interesting that the correlation decreases in the original trees. Our interpretation is that perhaps there is a more complex relationship at play between height and mean dependency distance in real data that cannot be linearly captured, and this complex relationship would be altered by the random components when generating the various artificial trees, especially as we relinearize the nodes.

6.3.3.2 Distribution of configurations

In this section we look at the distribution of local syntactic configurations by extracting trigrams and looking at their dependency relations. First we look at the non-linearized configurations : a→b→c and b←a→c, to analyze the differences in local structures between syntactic and randomly generated trees. Then we analyze the distribution of the four different groups presented in Section 6.3.1.2, and how this distribution is impacted by language and the type of treebank (syntactic and artificial). We will discuss here a few key points.

Non-linearized configurations In Figure 6.8, we can see the distribution of non-linearized configurations for one example language, French. For the *random _random* trees, we have 45% of b←a→c configurations and 55% of a→b→c configurations in trigram windows. For all other tree types, the first configuration is by far the most frequent one at the local level. We will keep this in mind when looking at the distribution

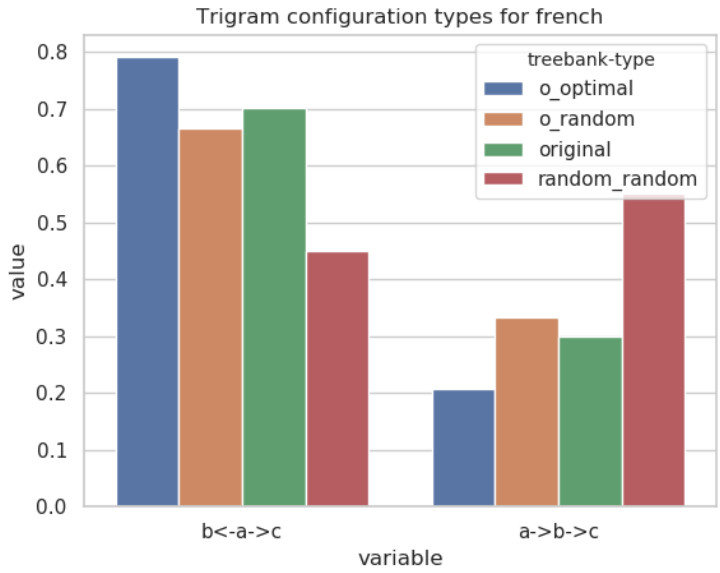


Figure 6.8: Non-linearized trigram configurations distribution for French

of linearized configurations.

We also observe that the results are fairly similar across all 4 languages, with *original_ optimal* showing the most unequal distribution (80%-20% respectively for b←a→c and a→b→c configurations), followed by *original* and *original_random* (around 60%-40%, although there is some variation depending on the language). One possible explanation for favouring b←a→c could be that it helps minimizing dependency distances, since it can lead to *balanced* configurations which are the optimal way to arrange dependents without introducing longer dependencies. If that is the case, we will see a high proportion of *balanced* configurations when we look more in detail at how these configurations are linearized. Another line of explanation could be that having too many a→b→c configurations introduces too much height in the trees, which could be a factor of complexity that natural languages try to avoid whenever possible. Differences between *original_optimal*, *original_random* and *original* can be explained by the linearization process: the optimal trees tend to favour shorter dependencies, which means that a higher percentage of triplets of nodes will all be connex, while non-optimal trees will sometimes linearize the nodes further away, thus excluding them from the extraction of triplets. It would be interesting to see if the distribution

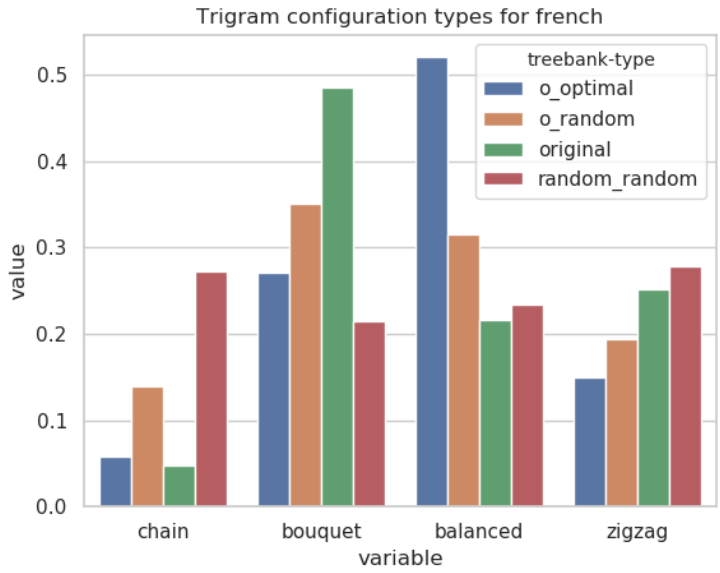


Figure 6.9: Trigram configurations distributions for French.

is similar when we look at all configurations of triplets and not just local ones.

Linearized configurations We then go on to look at these configurations once they have been subdivided according to the classification proposed in section 6.3.1.2. Note that the configurations *bouquet* and *balanced* are a result of the b←a→c configurations and that a→b→c will produce either *chain* or *zigzag*. We show the distribution for French in Figure 6.9. First we comment the results that are stable across languages: trees of the *random_random* variety show a slight preference for *chain* and *zigzag* as a result of the preference for b←a→c configurations, but inside each group (*chain* and *zigzag* / *bouquet* and *balanced*) the distribution is equally divided. This simply shows that once the structure has been selected, there is no bias towards a specific linearization strategy. The *original_ optimal* trees have a very marked preference for *balanced* which is to be expected because alternatively ordering dependents of a governor is the strategy employed to minimize dependency length. Next we find *zigzag* configurations, followed by *bouquet* and very few *chain*. Contrary to the potential explanation we advanced for the high frequency of b←a→c configurations, *balanced* configurations are not particularly frequent in the original trees (23% in Chinese, 14%

in English, 21% in French and 27% in Japanese), especially when compared to the *bouquet* configurations (37%, 52%, 48%, 30% respectively). Bouquet configurations are much more frequent in the syntactic trees than in the artificial ones. We have yet to find a satisfactory explanation for this. Even if we know that some arbitrary choices in the UD annotation scheme inflate the percentage of bouquet (*conj* (for conjuncts), *fixed* (to link words inside grammaticized expressions) and *flat* (for headless expressions) relations are always encoded as a bouquet), this does not seem sufficient to explain the difference with the other configurations, especially as those relations are not very frequent. We also remark that, if we were to use a schema with functional heads most of these *bouquet* configurations would become *zigzagz* or *chain*, so we could potentially find an explanation by investigating there. For the optimal model, the bouquet is not an optimal strategy to minimize dependency distances, so the bouquet configuration will, of course, be less critical in the optimal model. Compared to the other languages, Japanese has an interestingly high percentage of *zigzag* configurations. This can be partly explained by the segmentation used in the Japanese treebanks. The particles and agglutinated markers (for polarity, aspect, politeness...) have been annotated as separate tokens, which often creates many dependents on a single governor. A lot of these dependencies fall outside the trigram windows and are excluded from our analysis. Japanese being a head-final language, the configurations captured will often contain a head-final dependency and a marker of the dependent, which means that it will often fall into the *zigzag* bin. Nonetheless *bouquet* are still quite frequent as a governor often has several marks, and *balanced* capture nominal modifiers or compounds, and their case or topic marker.

6.3.4 Conclusion

We introduced several ways to generate artificial syntactic dependency trees and proposed to use those trees as a way of looking into the structural and linguistic constraints on syntactic structures for 4 different languages. We propose to incrementally add constraints on these artificial trees to observe the effects these constraints produce

and how they interact with each other. We limited ourselves to generating projective trees, which is a strong constraint that severely restricts the types of structures available, and therefore the variations of the different observed properties, and think that it would be interesting to also look at the result when allowing non-projective edges.

To expand on this work we would also like to see how the observed properties and the relations between them are affected by the annotation scheme, in particular contrasting schemes where content words are governors (as is the case in UD) and schemes where function words are governors (for example using the SUD schema proposed by [Gerdes et al., 2018]), as it will have an impact on height, dependency distances, and the types of configurations that can be extracted from the treebanks.

The syntactic configurations we extracted are only local, but it would be interesting to extract subtrees regardless of whether the nodes are neighbours in the sentence, especially as some syntactic relations are more likely to appear in more global configurations. In the future, we plan on looking at these larger configurations by extracting subtrees and analyzing their distribution. We also intend on digging deeper into the analysis of the present data, and propose predictive models that could help us clarify the relationship (whether they be linear or not) between the different features in order to build a more solid basis to verify our hypotheses and propose explanations for the observations we made.

The authors propose a suite dedicated to the analysis of some of the beforementionned properties and to the generation of random and artificial trees. Having such as resource available will greatly facilitate the type of work we have started.

6.4 Conclusions of the chapter

In this chapter we studied several scenarios where treebanks were used as a resource to observe some properties about the language they describe. The first study delves into the interaction between the Menzerath-Altmann Law (MAL) and Heavy Constituent Shift (HCS), exploring how these principles might influence the size of constituents. The study leverages a large set of treebanks in diverse languages, to test the hypoth-

esis that the size of certain linguistic constituents follows predictable patterns based on MAL and HCS principles. The results confirm this hypothesis across a wide range of languages, suggesting that the co-effect of MAL and HCS could be considered a linguistic universal. The result highlights the potential of using multilingual and coherently annotated treebanks to bridge traditional linguistic studies with quantitative analyses. However, the chapter also notes the limitations of the current dataset, such as uneven language representation and potential biases in the treebanks. Future research directions include considering data from verb-final languages, examining the impact of sample size and treebank variations on the results and exploring different types of clause structures.

The second study introduces the concept of Linear Dependency Segment (LDS) as a new linguistic unit positioned between the word and the clause. The LDS is defined as the longest possible sequence of words inside a clause, where all linear neighbours are also syntactic neighbours. This new unit aims to address issues formerly raised, related to the application of the Menzerath-Altmann Law (MAL) at the syntactic level, particularly in the relationship between the lengths of sentences and clauses. The empirical analysis conducted on Czech treebanks (Czech-PDT UD and FicTree) demonstrates that the LDS can be a meaningful unit for modelling the MAL on a syntactic level. The results show a clear decreasing tendency of mean clause length measured in LDSs as sentence length (measured in clauses) increases, fitting well with the MAL's predictions. This suggests that longer sentences, which typically have more clauses, tend to consist of shorter clauses when measured in LDSs. The study also indicates that the LDS avoids problems encountered when measuring clause length in words and aligns with the capacity of short-term memory, supporting one of the theoretical explanations of the MAL. However, this research is preliminary, and further investigation across more linguistically diverse datasets is necessary to establish LDS as a standard linguistic unit. Moreover, the potential relationship between LDSs and dependency distance minimisation, as well as the possible applications of LDS in text classification and understanding sentence structure, present interesting avenues for future research.

The last study is centred around the comparison of various properties of random trees and syntactic trees (extracted from treebanks). Various procedures for generating random dependency trees with constraints are explored. Three different tree-generating algorithms are used: original_random, original_optimal, and random_random, each introducing varying levels of constraints in the tree generation process. We used these artificial trees to compare their properties with the properties of syntactic trees and analyse the relationships between these properties in order to find out which relationships are formally constrained and which are linguistically motivated. The proposed analysis is based on UD treebanks (version 2.3, [Nivre et al., 2018]) for four languages: Chinese, English, French and Japanese. We took into consideration five metrics: tree length, height, maximum arity, mean dependency distance and mean flux weight, and also looked into the distribution of local configurations of nodes. The *original_random* algorithm samples an unordered dependency structure from a treebank and generates a random projective linearization. The *original_optimal* algorithm also samples an unordered structure but aims to minimise dependency distances, leading to more balanced trees. The *random_random* algorithm generates both the structure and linearization randomly, providing a baseline for comparison. Among the findings of this study we observed a correlation between Mean Dependency Distance and Mean Flux Weight: These properties are highly correlated in all treebanks, with the correlation being strongest in the *original_optimal* trees. This suggests that minimising dependency distances in tree structures tends to reduce the opportunities for configurations introducing disjoint dependencies, thereby affecting the mean flux weight. There is also a strong correlation between the length and height of trees in both syntactic and random structures. This indicates that the relationship isn't solely due to linguistic factors, but rather is formally constrained. The distribution of non-linearised configurations like b←a→c and b a→b→c varies across languages and treebanks at the local level. Notably, *original_optimal* trees shows a stronger preference than other settings for the b←a→c configurations that minimize dependency distances. The distribution of linearized configurations such as *chain*, *zigzag*, *bouquet*, and *balanced* varies significantly across treebank types. Syntactic trees show a prefer-

ence for *bouquet* configurations, while optimal trees favour balanced configurations to minimize dependency distances.

The studies offer numerous perspectives for future research. We are interested in the various procedures for artificial tree generation, and to experiment with varying degrees of randomness and constraints. For syntactic trees, the influence of the annotation scheme (especially insofar as contrasting content-word-governor schemes with function-word-governor schemes) would also be an interesting aspect to investigate. Another future direction is to explore larger syntactic configurations, going beyond simple sequences and instead examining subtrees or other relevant linguistic units. Developing predictive models to clarify the relationships between different syntactic features and verify hypotheses more robustly is another possible direction. Overall, this research provides insights into how different constraints and generation strategies impact the properties of syntactic trees, offering a framework for understanding the structural and linguistic constraints on syntactic structures.

The idea behind this comparison between syntactically grounded and artificial tree properties was to be able to infuse some knowledge into a structure induction procedure, in particular to avoid inducing structures that seemed too random. This knowledge would be translated into some constraints which would reduce the search space and give better plausibility to the induced structures. We explored ideas in this vein (reducing the search space for syntactic structure induction) using 'universal laws' or observed properties of languages, but we did not go so far as to integrate it into any structure induction model.

Chapter 7

Structure Induction from Raw Corpora: Mining Syntactic Fragments

Chapter contents

7.1 Objectives 195

7.2 Autonomy and Syntactic Units 196

7.2.1 Autonomy Measure 196

7.2.2 Syntactic Units 198

7.3 Data 198

7.4 Methodology 201

7.5 Results and Discussion 203

7.5.1 Size of the Training Corpus 203

7.5.2 Influence of the Annotation Scheme 204

7.5.3 Evolution of Scores Depending on the n Best 205

7.5.4 Comparison with the baseline 207

7.6 Conclusion and Perspectives 208

CHAPTER 7. STRUCTURE INDUCTION FROM RAW CORPORA: MINING SYNTACTIC FRAGMENTS

In this chapter, we focus on the process of inducing syntactic structures from raw corpora. Our focus is on understanding and formalizing the interplay of syntactic units, boundaries, and relations as they emerge from textual data. Exploring rather simple correlations between statistical measures and syntactic structure is interesting in view of the explainability of the results and is not attempting to optimize performance, for example, in the domain of zero-shot parsing.

This chapter aims to answer several critical questions:

- How effectively can we retrieve syntactic data from a raw corpus?
- To what extent do these data summarize syntactic structures?
- What insights and limitations arise from this approach?

To this end, we will explore our proposed method for extracting syntactic fragments. This method is based on a boundary view of syntactic units and relies on properties that we use to approximate the strength of boundaries at specific interword positions.

7.1 Objectives

This section is based on [Courtin, 2021]. It explores the unsupervised induction of syntactic structures within the framework of dependency syntax, a syntactic theory introduced by [Tesnière, 1959]. The goal is to determine if contiguous segments within a sentence form connected fragments of the dependency tree. [Gerdes and Kahane, 2011] demonstrated that the connection structure of a sentence could be entirely defined based on the set of fragments of that sentence, which are defined particularly by their ability to be autonomized. Thus, we propose to use an entropy-based measure of autonomy to induce syntactic fragments. This autonomy measure has been successfully used in the past to identify smaller units such as words, but we seek to explore its efficacy in extracting larger units for the task of syntactic structure induction. Our hypothesis is that changes in entropy across a sentence can help make informed predictions about the boundaries between syntactic units, potentially aiding in deciding which sequences are syntactic units and which are not.

This work follows in the footsteps of other research focusing on tasks like unsupervised syntactic parsing, the induction of syntactic structures, or the search for syntactic information in dense vector representations (or embeddings) using *structural probes* [Hewitt and Manning, 2019].

Computationally, our proposed method is lighter than these latter approaches, which require training heavier models like BERT [Devlin et al., 2019] and ELMO [Peters et al., 2018], necessitating large training corpora and the mobilization of extensive, resource-intensive computational infrastructures [Strubell et al., 2018]. Furthermore, we hope that the entropy-based autonomy measure will be more interpretable, as it is associated with a solid theoretical foundation in linguistics, notably with Harris's hypothesis [Harris, 1955], which proposes that a larger paradigm of successors or predecessors at a position between two tokens (in his case, characters) indicates the presence of a linguistic boundary (for him, morpheme boundaries). This theory seems naturally adaptable to syntactic unit boundaries, though it remains to be seen if entropy will provide sufficient information, given the greater variability of tokens

compared to characters.

Other studies have sought to establish links between predictors and the presence of dependency relations, such as [Futrell et al., 2019], who observed a connection between mutual information for a pair of words and the presence of a dependency relation between them. Since mutual information is related to entropy, it seems particularly interesting to use an autonomy measure based on the latter.

We will begin in section 7.2 by presenting the autonomy measure used to predict the syntactic nature of a unit. In section 7.3, we will introduce the corpus that will be used to train the autonomy estimation model, as well as the tree-annotated corpora on which our predictions will be evaluated. In section 7.4, we will briefly describe the process of extracting random fragments that will serve as our reference method. Finally, in section 7.5, we will present our initial results.

7.2 Autonomy and Syntactic Units

7.2.1 Autonomy Measure

The autonomy measure we employ is described in [Magistry, 2013]. It considers a unit autonomous if its elements are cohesive and its boundaries are unpredictable, located at positions of high entropy.

The autonomy measure is constructed as follows: first, **branching entropy** is assessed at each inter-word position. This branching entropy accounts for the diversity of tokens that can follow or precede a certain context. Then, **branching entropy variation** is calculated by subtracting the branching entropy at the previous position from the branching entropy at the current position. This measure helps to observe how entropy increases or decreases upon adding a new token.

The autonomy of an n-gram is then calculated by summing the branching entropy variations[1] from both left-to-right and right-to-left traversals of the text. Higher autonomy indicates that the n-gram's boundaries have stronger entropies compared to

[1] A normalisation is put in place to centre the measure on 0 for every size of ngram, so that shorter ngrams are not favoured

inter-word positions, making it more likely that the n-gram is a syntactic unit.

The formal calculation to arrive at this autonomy (following [Magistry, 2013]) is:

Given an n-gram $x_{0..n} = x_{0..1}x_{1..2}...x_{n-1..n}$ with left context $X_{\rightarrow}$, the right branching entropy is defined as:

$$h_{\rightarrow}(x_{0..n}) = H(X_{\rightarrow}|x_{0..n})$$

$$h_{\rightarrow}(x_{0..n}) = -\sum_{x \in X_{\rightarrow}} P(x|x_{0..n}) log P(x|x_{0..n}).$$

For left branching entropy, we note $X_{\leftarrow}$ as the right context of $x_{0..n}$, giving us:

$$h_{\leftarrow}(x_{0..n}) = H(X_{\leftarrow}|x_{0..n})$$

.

The branching entropy variation in both directions is then calculated from the branching entropies of the n-grams $x_{0..n}$ and $x_{0..n-1}$:

$$\delta h_{\rightarrow}(x_{0..n}) = h_{\rightarrow}(x_{0..n}) - h_{\rightarrow}(x_{0..n-1})$$

$$\delta h_{\leftarrow}(x_{0..n}) = h_{\leftarrow}(x_{0..n}) - h_{\leftarrow}(x_{1..n})$$

After applying the normalization mentioned in footnote [1], the autonomy of the n-gram $x_{0..n}$ is determined:

$$a(x_{0..n}) = \widetilde{\delta} h_{\leftarrow}(x_{0..n}) + \widetilde{\delta} h_{\rightarrow}(x_{0..n})$$

This method assigns autonomy to each n-gram, allowing for the calculation of a segmentation score by summing the product of each n-gram's autonomy and its size (in terms of tokens). This provides a method for ranking different segmentations according to their overall score and gives an autonomy score for each n-gram.

7.2.2 Syntactic Units

This autonomy measure was originally conceived to identify words, but we believe it can be applied to identify other, larger syntactic units. Specifically, we aim to identify sequences of tokens that form a connected part in the dependency structure, i.e., catenas [Osborne et al., 2012].

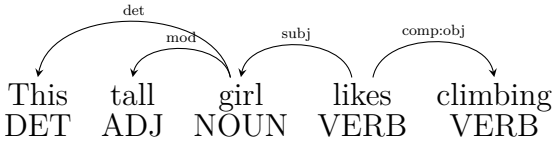


Figure 7.1: Dependency tree for the sentence *This tall girl likes climbing*.

For example, in the sentence *This tall girl likes climbing*, with dependency structure shown in Figure 7.1, we can identify 15 catenas: (This), (girl), (likes), (climbing), (this girl), (tall girl), (girl likes), (likes climbing), (This tall girl), (This girl likes), (tall girl likes), (girl likes climbing), (This tall girl likes), (This girl likes climbing) and (This tall girl likes climbing). However, (This tall) is not a catena, as it does not form a connected part of the dependency structure.

Thus, it is these connected portions of the dependency structure, a type of syntactic unit known as catena, that we aim to extract in the following sections.

7.3 Data

We use two French corpora: a raw corpus for training the autonomy model, and a dependency-annotated corpus, which we continue to train the autonomy model on (using only the text). Including the text from the annotated corpora ensures that the vocabulary appearing there is well covered. The dependency structures from the annotated corpus are used solely for evaluating the syntactic unit predictions by comparing the predicted units with the reference structure.

The first corpus consists of literary works, segmented into sentences and tokens. We sample sub-corpora of varying sizes to study the impact of training corpus size

on predictions. Regarding the annotated corpora, we use 6 corpora from the Universal Dependencies project [Zeman et al., 2020], version 2.7: FQB, GSD, ParTUT, PUD, Sequoia, and Spoken. Altogether, these corpora comprise 26,555 sentences and 509,257 tokens. They form a heterogeneous corpus in terms of modality and genre, including written and spoken language, and genres covering press articles, medication instructions, wikis, blogs, legal texts, and transcribed speech.

Within these annotated corpora, nodes that do not correspond directly to tokens have been introduced to account for amalgamations like "au" (à+le) 'at the', or "du" (de+le) 'of the'. Since these disamalgamated forms will not appear in our raw training corpus, we choose to apply a rewriting grammar to these annotated corpora using Grew [Guillaume et al., 2012a], to restore the original tokens by merging the amalgams. [2] An example of this transformation is presented in Figure 7.2.

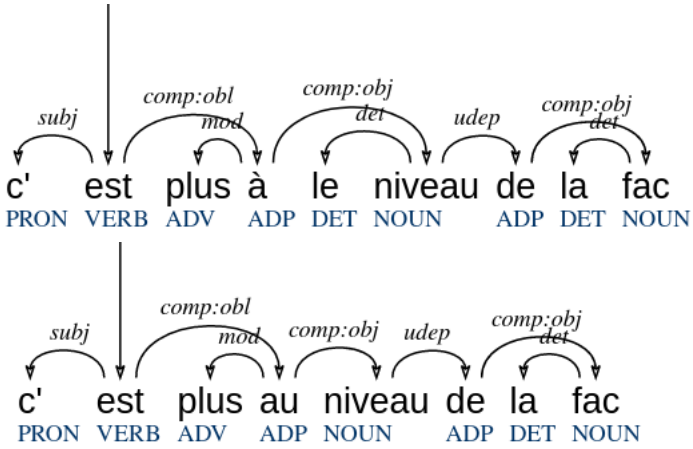


Figure 7.2: Example transformation to merge an amalgamation (à+le → au)

Another aspect we find very interesting is the influence of the annotation scheme on the evaluation of our method. Depending on the chosen scheme, the sequences considered as syntactic units will vary, meaning the model's performance will be conditioned by this scheme. For example, a fragment extracted by the model may be a catena in a scheme with functional heads but not in one with lexical heads. To test how crucial this criterion is, we evaluate our predictions on 4 different versions of the

[2]The corresponding grammar by Bruno Guillaume is available here: https://github.com/surfacesyntacticud/tools/blob/master/textform_wordform/remove_amalg_fr.grs

annotated corpora, obtained after applying dependency graph rewriting grammars. The differences between these 4 versions can be described as follows:

- UD version: the original annotation scheme for all annotated corpora except GSD and Spoken, which are maintained in the SUD version. In this version, heads are lexical elements, and function words are dependents, creating generally flatter structures. A more detailed description of the differences between the UD and SUD schemes can be found in [Gerdes et al., 2018].

- SUD version: the native annotation scheme for the GSD and Spoken corpora. Unlike the UD scheme, the heads are functional, leading to generally deeper structures.

- SUD+ version: a more extreme version of the SUD scheme, identical in every aspect except for the relations between nouns and determiners, which are reversed so that the determiners become heads. Other relations remain the same.

- SUD++ version: identical to the previous version, with the former dependents of the noun now attached to the determiner, so that it dominates all elements within a nominal group.

We know these choices regarding the annotation scheme will more or less significantly modify the encountered structures, impacting the model evaluation. As an initial observation, we calculate the proportion of bigrams, trigrams, and quadrigrams that are catenas in the different versions of the annotated corpora. The higher these proportions, the more likely the model will extract a significant number, though this does not necessarily mean an improvement. The results in Table 7.1 show that the SUD+ version is by far the richest in observed catenas for bigrams, trigrams, and quadrigrams, and the UD version presents the fewest, with SUD and SUD++ versions having similar proportions and falling between the UD and SUD+ versions.

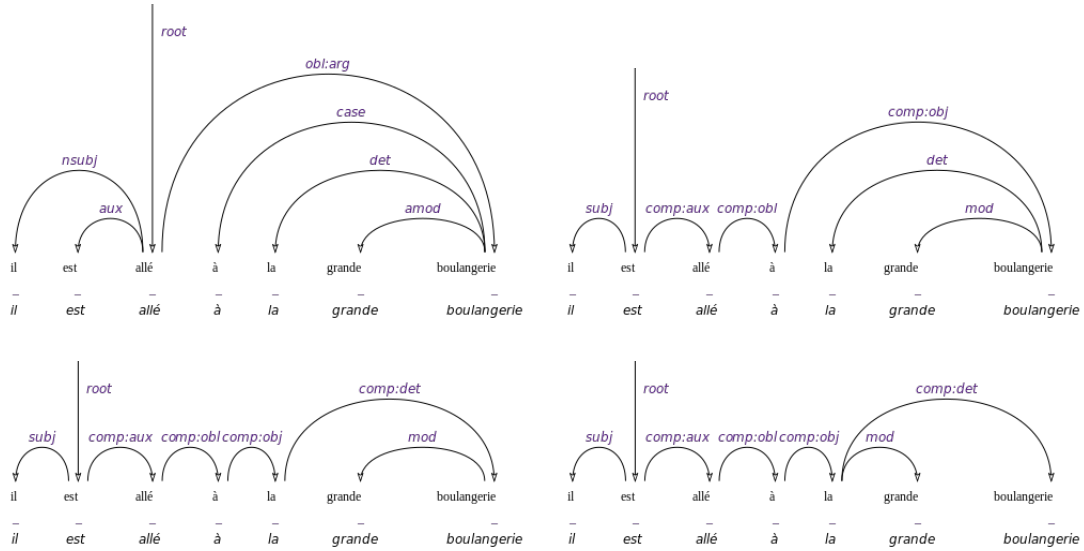


Figure 7.3: Example annotation for the 4 schemes (from left to right and top to bottom: UD, SUD, SUD+, SUD++)

Scheme	Bigrams	Trigrams	Quadrigrams	All
UD	0.47	0.39	0.33	0.40
SUD	0.62	0.53	0.46	0.54
SUD+	0.72	0.60	0.51	0.61
SUD++	0.63	0.52	0.46	0.54

Table 7.1: Proportion of sequences of length 2 to 4 that are catenas in the different annotation schemes of the annotated corpora.

7.4 Methodology

In section 7.2, we described the autonomy measure we use to extract fragments that we hope are syntactic units. This measure is implemented in the ELeVE tool [3] [Magistry and Sagot, 2012], which we use to obtain the fragments.

The tool allows us to calculate autonomy for all n-grams, but also to rank segmentations according to their overall score and possibly extract the n best. These pieces of information are particularly interesting because the autonomy score of a sequence does not depend on the context in which it is found, as it is calculated from all its contexts. However, to determine if the sequence is indeed a syntactic unit, we would like this context of occurrence to be taken into account, which is the case when focusing on

[3]The tool is available online here: https://github.com/kodexlab/eleve

the overall score of a segmentation. Thus, a segmentation in which only one segment obtains a very high score and all other segments have mediocre scores will appear lower in the ranking than a segmentation that allows obtaining several segments with good scores, even if each of these scores is individually lower than that of the very good segment in the first segmentation.

Therefore, for each sentence, we extract a list of fragments, each associated with a unique autonomy score, and the ranking of different segmentations in which it appears. We also set the maximum size of n-grams to count at 5 (estimates for longer segments would not be reliable enough), which will give us fragments ranging from 1 to 4 in length.

In terms of evaluation, we focus on two aspects. The first is to see if the selected fragments indeed form catenas in the dependency tree of the reference corpus. The proportion of these selected fragments that are catenas will provide us with our precision score. We also measure how much of the dependency structure is covered by the extracted fragments, i.e., what proportion of the catenas present in the structure we have managed to extract, which will constitute our recall.

Baseline We propose inducing random fragments to compare our method with a baseline. If our hypothesis holds, we should observe better compatibility of fragments induced using the autonomy measure compared to those induced randomly.

Firstly, it's important to clarify the difference between two random processes for sampling token sequences:

Random segmentation involves proposing a unique division of the sentence, usually by traversing it and assigning a probability of introducing a boundary at each inter-word position.

Conversely, **random fragmentation** aims to induce multiple segmentations, allowing for overlapping fragments. We prefer this second option since its output will more closely resemble the induced fragments.

Among the many possible ways to propose random fragmentation, we suggest the following:

Each token in the sentence is considered the nucleus of a random fragment. For this fragment, we randomly draw a length between 2 and 4 (since these are the possible lengths for our candidate fragments). Once the fragment length is defined, we randomly draw the position of the token within the fragment (first, second, third, fourth). If the position is incompatible with the token's position in the sentence, we repeat until obtaining a compatible position.

For example, for the sentence "We draw something at random", we could have the following proposition for "draw":

- fragment length: 3
- position within the fragment: third (impossible), first (possible)

This would yield a random fragment "draw something at".

This first random fragmentation is called uniform, as there is no particular weighting on the length of fragments; they are equally probable.

We also propose a second version, called weighted, with weighting on fragment lengths, so that the distribution of lengths in the original and random fragmentations are similar. The selected weights are as follows: 0.77 for fragments of length 2, 0.15 for length 3, and 0.08 for length 4.

7.5 Results and Discussion

In this section, we present and analyze initial results from our experiments, showing that autonomy could allow us to extract syntactic fragments.

7.5.1 Size of the Training Corpus

To obtain good entropy estimates on bigrams, trigrams, and quadrigrams, a sufficiently large training corpus is needed. We start by examining the precision of extracted fragments for different corpus sizes: 1,000 tokens, 10,000 tokens, 100,000 tokens, 500,000 tokens, and 1 million tokens.

We extract fragments appearing in the best segmentation of each sentence and verify their catena status in the SUD version of the annotated corpora. The corresponding results are presented in Figure 7.4, where we mainly note that the best overall precisions (respectively 0.83 and 0.82) are obtained for the two largest corpus sizes. This precision gain comes primarily from better predictions on trigrams and quadrigrams, which are too rare in the smaller corpora to reliably estimate their autonomy.

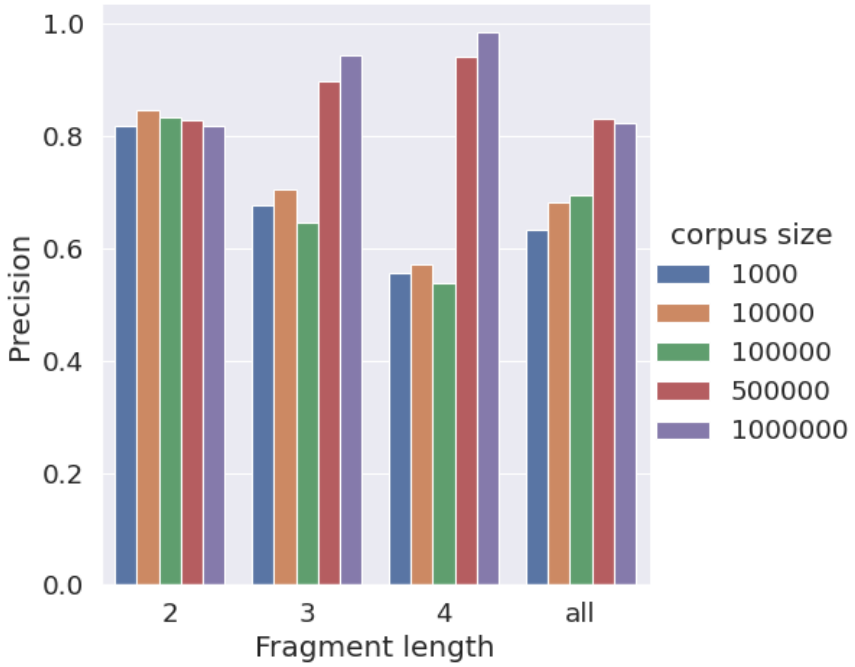


Figure 7.4: Influence of training corpus size on the precision of extracted fragments (scheme: SUD, n=1)

7.5.2 Influence of the Annotation Scheme

We now focus on variations in precision evaluation according to the annotation scheme of the annotated corpora.

The overall precision scores indicate that the extracted fragments more closely follow the SUD+ scheme (0.81), compared to the SUD scheme (0.68). Changing only the relationship between nouns and determiners so that determiners become heads

gains us 0.13 in precision, which is considerable. It is also interesting to note that there are ultimately few differences between the scores for the SUD and UD schemes, although the structures in these two versions are very different.

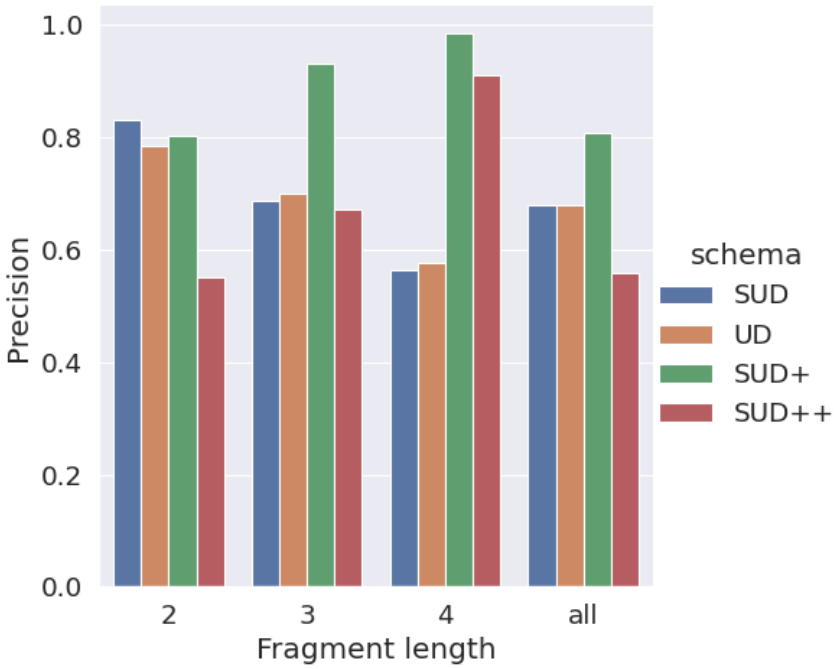


Figure 7.5: Influence of the annotation scheme on the precision of extracted fragments (size: 1 million tokens, n=1)

7.5.3 Evolution of Scores Depending on the n Best

So far, the evaluation has only concerned fragments belonging to the best segmentation of each sentence. We note a good improvement compared to the baseline for these fragments compared to the random baseline, but it would be incomplete to stop here without talking about recall. In order to visualize the evolution of precision and recall as a function of the n best segmentations selected, we choose to focus on sentences of fixed length, which will enable us to set a maximum n that corresponds to the total number of possible segmentations.[4] We select all sentences of length 10 and vary n

[4]The number of possible segmentations for a sentence of length m with segments of length between 1 and p can be achieved from the Fibonacci p-numbers [Olaiju and Taiwo, 2015].

between 1 and 401 to cover all possible segmentations. Accuracy starts out fairly high, with 86% of extracted fragments being catenas, then decreases rapidly in the 25 best segmentations. It then decreases more slowly, reaching 0.57 when all segmentations are taken into account. On the recall side, we observe 3 phases: a very sharp increase in the first 25 segmentations, where we reach 0.48, then a sharp increase until around the 250th segmentation (0.96) and a much slower increase towards the end. This suggests that appropriately selecting a n cutoff for the n-best segmentation is crucial.

To be able to set an n that would allow both good precision and sufficient recall, we'd have to look at the extent to which certain catenas can be deduced from other catenas that combine together (for example, a catena of length 2 that combines with a catena of length 3, with one of the nodes in common, could allow us to deduce the catena of length 4 that encompasses both).

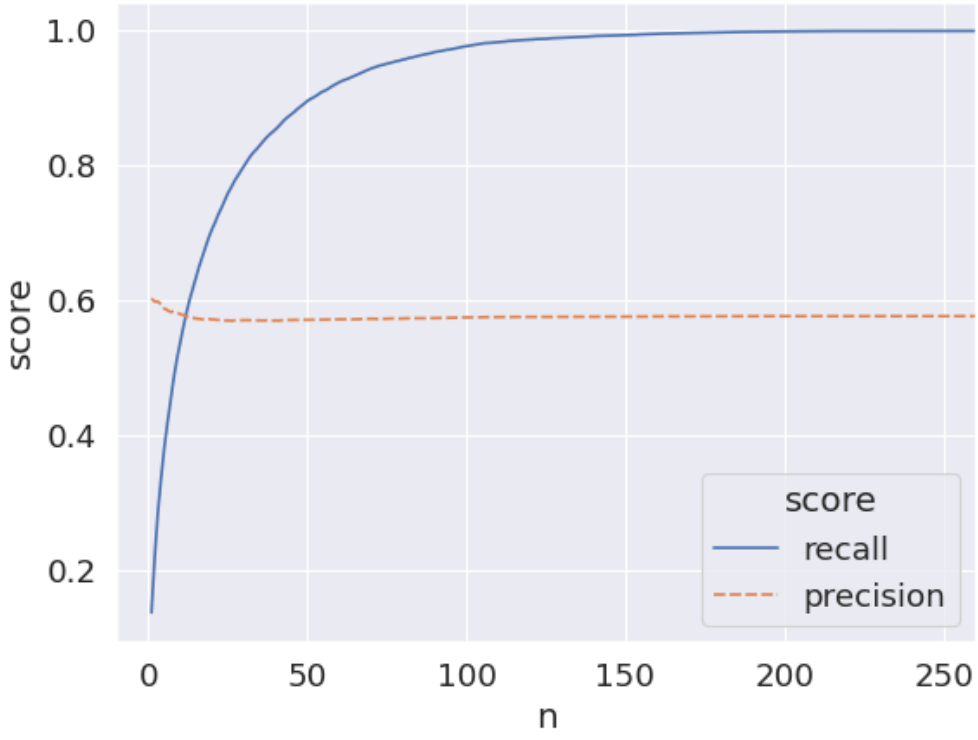


Figure 7.6: Precision and recall on extracted fragments from the n best segmentations (SUD schema, 1M tokens)

7.5.4 Comparison with the baseline

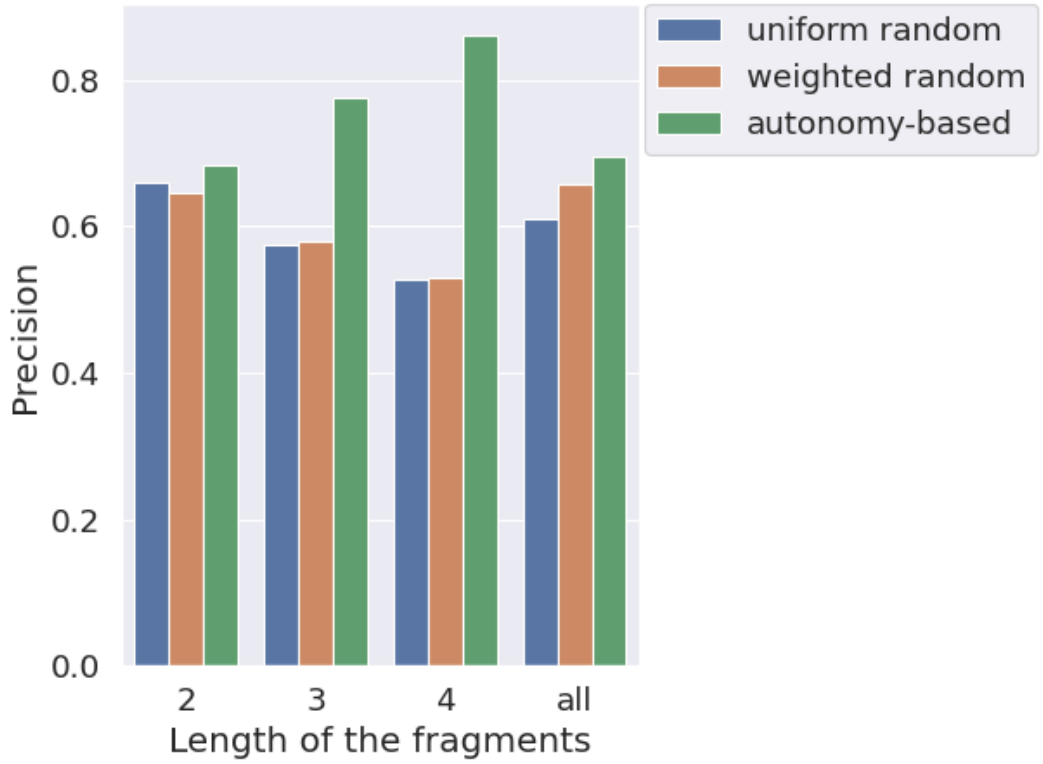


Figure 7.7: Precision on random fragments and fragment candidates based on the size of the fragment. Candidate fragments are from the 10 best segmentations (SUD schema, 1M tokens)

In Figure 7.7, we can see the extent to which the fragments extracted (whether randomly or using our method) are indeed catenas in the reference tree corpus. As far as random fragments are concerned, we have similar accuracies for both methods, with respectively for the uniform method and the weighted method: an accuracy of 0.66 and 0.65 for fragments of length 2, 0.58 for fragments of length 3, 0.53 for fragments of length 4 and 0.61 versus 0.66 if we don't take length into account. The longer the fragment, the less likely it is to be a catena, which corresponds to the frequencies described in table 1. For fragments extracted using our method, the overall scores are higher: 0.68 for fragments of length 2, 0.78 for fragments of length 3, 0.86 for

fragments of length 4 and 0.70 if length is disregarded. It's particularly interesting to see that, unlike with random fragments, accuracy here increases with fragment length. We think this is an encouraging sign, as these catenas are important if we are to have any hope of inducing a dependency structure, due to their overlap with the other catenas.

The performance of our model is highly dependent on the n we choose here: the higher the n, the noisier the predictions will be, and the closer they will be to the random method we use as a reference. On the other hand, with a small n, we will get much better predictions than with random, but at the expense of recall.

7.6 Conclusion and Perspectives

We aimed to demonstrate the feasibility of predicting the syntactic nature of a token sequence in French, based on entropy.

We propose extracting fragments using an entropy-based autonomy measure and show that these are more often syntactic units (specifically catenas) than with a random reference method. We also demonstrate that the training corpus must reach a certain size to hope to extract longer fragments effectively.

Experiments on French suggest that the structure induced in this manner more closely aligns with the SUD+ schema with functional heads and determiners as heads of the nouns they combine with, as this scheme achieves the best precision.

There are still many avenues to explore, particularly in determining the minimum coverage required to induce good structures from a limited number of identified units. This would allow us to select only a portion of the best fragments and avoid introducing overly noisy fragments.

Another aspect could involve focusing on sequences that seem least likely to be syntactic units. Identifying these non-units could allow us to eliminate a number of connections right away, reducing the complexity of the problem of inducing the structure.

Such a method could be used in future work to propose an unsupervised induction

of syntactic dependency structures.

This chapter has laid out a basic framework for understanding and extracting syntactic structures from raw text. The journey from theoretical constructs to actionable data points is complex, necessitating a nuanced approach to linguistic analysis. As we refine these methods, our understanding of language structure and its computational modeling will undoubtedly be refined, opening new pathways in the field of linguistics.

Chapter 8

Conclusion and perspectives

In this thesis we have given a sense of the importance and varied uses of treebanks, which are syntactically annotated corpora. They provide a valuable window into the structure of human languages and constitute essential resources in linguistic research, language teaching, and language technology development. Treebanks contribute to areas such as linguistic typology, prosody, and semantics. With the development of language technologies, they have become an essential resource notably as training and evaluation materials for natural language processing systems.

The history of treebanks reveals an evolution from manual analyses of example sentences to the modern treebanks we know today. As the field as developed, standards have been adopted to allow large multilingual sets of treebanks to be developed using common annotation schemes, which has far reaching consequences. We have also covered the methods adopted to develop such treebanks, under varying resource scenarios, and presented some of the challenges that arise when dealing with previously unannotated languages, text genres and domains.

We have showcased our experiences in dealing with hard to annotate linguistic phenomena such as multi-word expression, where tokenization syntax and semantics meet, and proposed a way to detangle those aspects more clearly. We also describe the development of a treebank of Naija, a pidgin-creole of Nigeria ,in a surface syntactic annotation scheme. This has led us to describe some of the specific syntactic con-

structions of Naija and relate the challenges we encountered with previous discussions relating to the development of annotation guidelines and the principles that underpin them. The Naija treebank constitutes a precious resource for the analysis of the language itself, and as a building bloc for the development of ulterior resources. While the field of natural language processing has evolved towards more multilingualism, there is still a crucial need for treebanks for many less-described languages. We are pleased to see that in recent years, more attention has been dedicated to this aspect.

Our experience in treebank annotation, has shown the need for appropriate tools to facilitate the iterative and collaborative enrichment of the annotation. This has led us to integrate two existing tools with had complementary functionalities : Grew and Arborator. With Arborator-Grew, both annotation and querying of the annotation can be done, which opens up new possibilities to develop, maintain and curate treebanks. As a result, Arborator-Grew has been adopted inside classrooms and annotation campaign, demonstrating that it answers to the needs of the community.

Later on, we have used treebanks as datasets to explore the properties of syntactic trees. We explored the interaction between two principles : Menzerath-Altmann Law and the Heavy Constituent Shift, which have both been studied for a variety of languages. Since both laws deal with the size of constituents, we investigated the relationship between the two. Our hypothesis was confirmed on a large dataset representing 80 languages. While working on Menzerath-Altmann Law, we found that there were difficulties in applying this law at the syntactic level using clauses as a unit. Instead we introduced the Linear Dependency Segment as a unit, and found that it integrated well with the Menzerath-Altmann Law, at least for our pilot study on Czech.

In our last chapter, we experimented with an entropy-based measure of autonomy to extract syntactic fragments from unannotated texts. These fragments presented some similarity with well-defined syntactic units. There is however still much work to be done to propose a fully developed method to induce syntactic structures based on these preliminary results.

Appendix A

Appendix

A.1 Grew patterns

A.1.1 Filtering query results based on a subpattern

This pattern can be translated as : Look for the subject (S) of a verb (V). Then clusters results based on whether the subject is found before the verb or after the verb.

```
pattern {
    S [];
    V [upos=VERB];
    V -[subj]-> S
}

% wether
S << V
```

A.1.2 Complement of auxiliaries

```
pattern {
    GOV [];
```

```
DEP [];
GOV -[comp:aux]-> DEP
}
```

A.1.3 Heavy constituent shift extraction

This pattern can be translated as : Look for a governor with both an oblique or adverbial modifier and an object, where the governor precedes the the oblique/adverbial modifier, which in turn precedes the object. The governor should have no other dependent behind it.

```
pattern { X -[obl|advmod]-> B;
X -[obj]-> C;
X << B;
B << C;
}
without {
X -> D;
X << D
}
```

A.1.4 Co-effect of Menzerat-Altmann and Heavy Constituent Shift

In this pattern we are looking for a governor (X) with two dependants (B and C) located on its rights. B is a dependant with function *obl* (oblique) or *advmod* (adverbial modifier), and C if a dependant with function *obj* (object). X precedes B, which in turn precedes C (so that it can be deduced that X also precedes C). We also require X not to precede another dependant, which we call D.

```
pattern {
X - [obl | advmod] -> B;
```

```
X - [obj] -> C;
X << B;
B << C}

without {
X -> D;
X << D}
```

A.2 Tables of experimental results

A.2.1 Co-effect of MAL and HCS : number of selected clauses per language

Language	a	b	c	Number of ~XAB	Number of ~XC
Afrikaans	3.31	13.37	13.03	211	1281
Akkadian	1.9	4.4	1.71	10	276
Akuntsu	0	0	1.1	0	10
Albanian	2.53	6.16	5.98	19	51
Amharic	1.0	1.08	1.08	49	326
AncientGreek	2.41	5.2	4.14	8725	25192
Apurinã	1.08	2.67	1.61	12	54
Arabic	3.79	13.43	11.04	27627	25993
Armenian	3.19	8.91	6.62	182	2274
Assyrian	1.14	1.43	3.46	7	26
Bambara	2.53	9.27	5.63	168	1122
Basque	1.91	3.4	3.05	551	3997
Belarusian	2.33	5.07	5.16	4060	14177
Bhojpuri	7.1	9.9	9.95	10	74
Breton	2.29	4.44	3.91	228	480

Language	**a**	**b**	**c**	**Number of ~XAB**	**Number of ~XC**
Bulgarian	2.52	5.56	5.52	2491	9637
Buryat	1.0	12.33	5.2	3	81
Cantonese	1.96	5.21	5.17	85	647
Catalan	3.87	10.7	10.88	9561	22072
Chinese	2.43	5.14	5.13	564	21956
Chukot	1.11	1.94	1.43	54	254
ClassicalChinese	1.38	2.34	1.9	1895	27462
Coptic	2.12	7.84	5.19	1634	2163
Croatian	2.9	7.22	6.98	2290	9913
Czech	2.58	7.14	7.21	32884	97789
Danish	2.15	6.44	6.02	2456	4502
Dutch	2.42	7.19	7.42	3039	7525
English	2.61	6.3	6.67	13502	36424
Erzya	1.43	2.88	2.76	291	972
Estonian	1.74	4.81	5.39	9123	17000
Faroese	1.74	6.18	4.02	1144	1892
Finnish	1.65	3.99	4.13	10145	22200
French	3.35	9.74	9.12	21348	47025
Gaelic	2.44	8.51	5.87	2160	1247
Galician	3.9	12.0	11.87	2542	7803
German	2.56	7.27	7.47	30612	36399
Gothic	2.05	4.77	3.36	1598	3739
Greek	3.44	10.36	10.1	1348	3094
Hebrew	3.26	10.29	9.32	3527	6222
Hindi	5.71	6.71	16.93	17	4741
HindiEnglish	2.29	4.94	4.88	250	932
Hungarian	2.54	8.82	7.25	386	1334
Icelandic	1.81	6.55	5.45	23643	35949

Language	**a**	**b**	**c**	**Number of ~XAB**	**Number of ~XC**
Indonesian	3.18	7.66	7.0	2801	9876
Irish	2.94	6.94	6.11	2654	1706
Italian	3.19	9.18	8.39	12833	34151
Japanese	4.29	1.88	2.74	17	61
Karelian	1.77	3.74	3.57	69	138
Kazakh	0	0	1.37	0	19
Khunsari	0	0	3.4	0	5
Komi	1.66	4.59	3.49	80	371
Komi-Permyak	1.1	2.0	3.04	10	53
Korean	0	0	2.59	0	233
Kurmanji	1.64	7.07	3.69	28	304
Latin	2.7	7.19	5.17	14020	43318
Latvian	2.32	6.01	5.54	3130	15899
Lithuanian	3.35	7.33	7.11	709	5271
Livvi	1.61	4.63	3.59	38	82
Maltese	2.96	7.9	8.66	873	3547
Manx	2.39	7.05	4.74	223	88
Marathi	2.8	1.6	3.1	5	20
MbyáGuaraní	2.07	3.07	2.59	43	300
Moksha	1.46	2.17	2.36	24	88
Mundurukú	0	0	2.11	0	19
Naija	1.67	4.53	3.77	10948	35429
Nayini	0	0	5.0	0	3
NorthSami	1.48	2.61	2.62	822	1669
Norwegian	2.03	6.1	5.78	14499	29070
Old Turkish	1.0	16.0	5.0	1	1
OldChurchSlavonic	1.61	3.67	2.43	1666	4045
OldEastSlavic	1.93	3.61	3.24	4560	8697

Language	**a**	**b**	**c**	**Number of ~XAB**	**Number of ~XC**
OldFrench	2.34	5.64	4.5	3952	11790
Persian	4.59	11.33	13.07	228	9922
Polish	1.88	4.84	4.79	10347	27611
Portuguese	3.69	9.56	8.93	10412	26994
Romanian	2.68	6.68	6.24	18678	45215
Russian	2.3	6.23	6.3	19485	72153
Sanskrit	1.66	3.18	3.53	204	1001
Serbian	3.03	7.22	7.28	1314	4870
SkoltSami	1.36	2.09	3.34	11	50
Slovak	1.84	4.01	3.91	1687	6982
Slovenian	1.8	5.98	6.0	2215	9008
Soi	0	0	6.0	0	1
South Levantine Arabic	1.47	3.4	2.77	30	56
Spanish	3.76	10.3	9.64	19650	43411
Swedish	1.91	6.01	5.45	5196	8793
SwedishSign	2.07	3.52	2.89	27	88
SwissGerman	1.57	5.29	5.93	7	30
Tagalog	1.72	3.1	3.17	72	69
Tamil	1.0	1.0	1.08	1	39
Telugu	0	0	1.32	0	25
Thai	3.24	6.94	6.66	781	2326
Tupinambá	8.0	5.0	0	1	0
Turkish	1.38	2.79	1.69	101	1561
TurkishGerman	1.47	4.96	4.44	212	606
Ukrainian	2.4	6.08	6.03	1946	5985
UpperSorbian	2.34	6.11	6.79	122	244
Urdu	9.39	7.32	20.39	31	1737
Uyghur	1.34	2.54	2.06	50	102

Language	**a**	**b**	**c**	**Number of ~XAB**	**Number of ~XC**
Vietnamese	1.49	4.45	4.21	1226	3946
Warlpiri	1.0	1.0	1.31	2	16
Welsh	2.67	9.84	10.32	798	656
Wolof	2.16	7.14	6.61	881	3890
Yoruba	2.27	9.66	8.33	160	376

Table A.1: Values of a, b, c and the numbers of selected clauses in each language.

A.3 Diagrams illustrating specific tree configurations

A.3.1 Trigrams configurations organized by type

Figure A.1: Example of a Balanced configuration

Figure A.2: Example of a Chain configuration

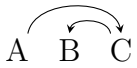


Figure A.3: Example of a Zigzag configuration

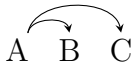


Figure A.4: Example of a Bouquet configuration

A.3.2 Disjoint dependencies

A set of dependencies is said to be disjoint when the nodes are not shared between the dependencies.

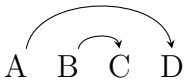


Figure A.5: Example of a disjoint set of dependencies

Bibliography

Anne Abeillé, Marie Candito, and Alexandra Kinyon. Ftag: current status and parsing scheme. In *Proc. Vextal*, volume 99, 1999.

Anne Abeillé, Lionel Clément, and Alexandra Kinyon. Building a treebank for French. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May 2000. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2000/pdf/230.pdf.

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, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2044. URL https://aclanthology.org/P15-2044.

Lluís Alemany-Puig, Juan Luis Esteban, and Ramon Ferrer-i Cancho. The linear arrangement library. a new tool for research on syntactic dependency structures. In *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, pages 1–16, Sofia, Bulgaria, December 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.quasy-1.1.

Ramadan Alfared. *Learning large-scale categorial dependency grammars*. PhD thesis, 12 2012.

Gabriel Altmann. Prolegomena to menzerath's law. *Glottometrika*, 2(2):1–10, 1980.

Gabriel Altmann, Michael H. Schwibbe, and Werner Kaumanns. *Das Menzerathsche Gesetz in informationsverarbeitenden Systemen*. G. Olms, Hildesheim, 1989. ISBN 978-3-487-09144-0.

Jan Andres and Martina Benešová. Fractal analysis of poe's raven, ii*. *Journal of Quantitative Linguistics*, 19(4):301–324, 2012. URL https://doi.org/10.1080/09296174.2012.714538.

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, page 119–130, 2016.

Peter Bakker. Pidgins versus creoles and pidgincreoles. *The handbook of pidgin and creole studies*, pages 130–157, 2008.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *LAW@ACL*, 2013. URL https://api.semanticscholar.org/CorpusID:7771402.

Frederick Augustus Porter Barnard. *Analytic grammar, with symbolic illustration, French, New York, 1836.* New York, 1836a.

Frederick Augustus Porter Barnard. *Analytic Grammar, with symbolic illustration*. E. French, 1836b.

Nicolas Beauzée. *Régime*, volume 14, page 511. 1765.

Martina Benešová and Radek Čech. *Menzerath-Altmann law versus random model*, pages 57–70. De Gruyter Mouton, Berlin, München, Boston, 2015. ISBN 978-3-11-036287-9. URL https://doi.org/10.1515/9783110362879-005.

Johann Gustav Friedrich Billroth. *Lateinische Syntax für die obern Klassen gelehrter Schulen*. Weidmann, 1832.

Johann Gustav Friedrich Billroth and Gustav Billroth. *Lateinische Syntax für die obern Klassen gelehrter Schulen*. Weidmann, 1832.

Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In Chu-Ren Huang and Dan Jurafsky, editors, *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://aclanthology.org/C10-1011.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. *Application of Graph Rewriting to Natural Language Processing*. Wiley Online Library, 2018.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. The groningen meaning bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer, 2017.

Cristina Bosco, Manuela Sanguinetti, and Leonardo Lesmo. The parallel-TUT: a multilingual and multiformat treebank. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1932–1938, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/209_Paper.pdf.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, volume 168, pages 24–41, 2002.

Claude Buffier. *Grammaire françoise sur un plan nouveau*. Le Clerc-Brunet-Leconte Montalant, Paris, 1709.

Solomija Buk and Andrij Rovenchak. Menzerath–altmann law for syntactic structures in ukrainian. *Glottotheory*, 1(1):10–17, 2008. doi: doi:10.1515/glot-2008-0002. URL https://doi.org/10.1515/glot-2008-0002.

Roberto Busa. The annals of humanities computing: The index thomisticus. *Computers and the Humanities*, pages 83–90, 1980.

Marie Candito, Benoît Crabbé, and Pascal Denis. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/392_Paper.pdf.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Rst discourse treebank, Feb 2002. URL https://catalog.ldc.upenn.edu/LDC2002T07.

Bernard Caron. Naijasyncor : A corpus-based macro-syntactic study of naija (nigerian pidgin), 2017. URL naijasyncor.huma-num.fr.

Bernard Caron. *Methodological and technical challenges of a corpus-based study of Naija*, page 57–75. Warsaw: University of warsaw press edition, 2020. ISBN 978-83-235-4631-3. doi: https://doi.org/10.31338/uw.9788323546313. URL https://wuw.pl/data/include/cms//West_African_Pawlak_Nina_Will_Izabela_2020.pdf?v=1612354546850.

Bernard Caron. Clefts in naija, a nigerian pidgincreole. *Faits de langues*, 52(1):159, 2021. doi: 10.1163/19589514-05201008.

Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. A surface-syntactic ud treebank for naija. In *TLT 2019 - 18th International Workshop on Treebanks and Linguistic Theories, Aug 2019, Paris, France*, 2019.

Heng Chen and Haitao Liu. A quantitative probe into the hierarchical structure of written Chinese. In Xinying Chen and Ramon Ferrer-i Cancho, editors, *Proceedings of the First Workshop on Quantitative Syntax (Quasy, SyntaxFest 2019)*, pages 25–32, Paris, France, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-7904. URL https://aclanthology.org/W19-7904.

Xinying Chen and Kim Gerdes. Classifying languages by dependency structure. typologies of delexicalized universal dependency treebanks. In *Proceedings of the Fourth*

International Conference on Dependency Linguistics (Depling 2017), September 18-20, 2017, Università di Pisa, Italy, page 54–63. Linköping University Electronic Press, 2017.

Xinying Chen, Kim Gerdes, Sylvain Kahane, and Marine Courtin. *The co-effect of Menzerath-Altmann law and heavy constituent shift in natural languages*, pages 11–24. De Gruyter Mouton, Berlin, Boston, 2022. ISBN 9783110763560. doi: doi:10.1515/9783110763560-002. URL https://doi.org/10.1515/9783110763560-002.

Hee-Soo Choi, Bruno Guillaume, Karën Fort, and Guy Perrier. Investigating Dominant Word Order on Universal Dependencies with Graph Rewriting. In *RANLP 2021 - Recent Advances in Natural Language Processing*, Online, Bulgaria, September 2021. URL https://hal.inria.fr/hal-03322613.

Noam Chomsky. *Aspects of the theory of syntax*. MIT Press, Cambridge, Massachusetts, 1965. ISBN Massachusetts.

Christian Chanard. Elan-corpa-v4.7.3. URL http://llacan.vjf.cnrs.fr/res_ELAN-CorpA.php.

Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA, February 1988. Association for Computational Linguistics. doi: 10.3115/974235.974260. URL https://aclanthology.org/A88-1019.

Maud Ciekanski. Les corpus: de nouvelles perspectives pour l'apprentissage des langues en autonomie? *Recherches en didactique des langues et des cultures. Les cahiers de l'Acedle*, 11(11-1), 2014.

Stephen Watkins Clark. *The science of the English grammar: A practical grammar in which words, phrases, and sentences are classified to their offices, and their relation to each other, illustrated by a complete system of diagrams*. H. W. Barnes Company, Cincinnati, 1847.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. Multiword Expression Processing: A Survey. *Computational Linguistics*, 43(4):837–892, 12 2017. ISSN 0891-2017. doi: 10.1162/COLI_a_00302. URL https://doi.org/10.1162/COLI_a_00302.

Pierre Corbin. *De la production des données en linguistique introspective*. Presses Universitaires de Lille, 1980. URL https://hal.science/hal-03562337.

Brown Corpus. The standard corpus of present-day edited american english., 1964.

Brown Corpus. A standard corpus of present-day edited american english, for use with digital computers (brown), 1979.

Marine Courtin. Extraction de fragments syntaxiques en français à partir d'une mesure d'autonomie basée sur l'entropie (mining French syntactic fragments using an entropy-based autonomy measure). In Pascal Denis, Natalia Grabar, Amel Fraisse, Rémi Cardon, Bernard Jacquemin, Eric Kergosien, and Antonio Balvet, editors, *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL)*, pages 15–27, Lille, France, 6 2021. ATALA. URL https://aclanthology.org/2021.jeptalnrecital-recital.2.

Marine Courtin and Chunxiao Yan. What can we learn from natural and artificial dependency trees. In Xinying Chen and Ramon Ferrer-i Cancho, editors, *Proceedings of the First Workshop on Quantitative Syntax (Quasy, SyntaxFest 2019)*, pages 125–135, Paris, France, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-7915. URL https://aclanthology.org/W19-7915.

Irene Cramer. The parameters of the altmann-menzerath law. *Journal of Quantitative Linguistics*, 12(1):41–52, April 2005a. ISSN 0929-6174. doi: 10.1080/09296170500055301.

Irene M. Cramer. *Das Menzerathsche Gesetz*, page 659–688. De Gruyter, Berlin / New York, 2005b.

Emanuela Cresti and Massimo Moneglia. *C-ORAL-ROM: integrated reference corpora for spoken romance languages*. John Benjamins Publishing, 2005.

William Croft, Dawn Nordquist, Katherine Looney, and Michael Regan. Linguistic typology meets universal dependencies. In *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories (TLT15), Bloomington, IN, USA*, pages 63–75, 2017.

Richard Eckart de Castilho, Eva Mujdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, 2016.

Miryam de Lhoneux. *Linguistically Informed Neural Dependency Parsing for Typologically Diverse Languages*. PhD thesis, Uppsala University, 2019. URL http://www.diva-portal.org/smash/get/diva2:1357373/FULLTEXT01.pdf.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07 2021. ISSN 0891-2017. doi: 10.1162/coli_a_00402. URL https://doi.org/10.1162/coli_a_00402.

Jeanne-Marie Debaisieux, Christophe Benzitoun, and Henri-José Deulofeu. Le projet ORFEO: Un corpus d'études pour le français contemporain. *Corpus*, 15:91–114, June 2016. doi: 10.4000/corpus.2936. URL https://hal.archives-ouvertes.fr/hal-01449600.

Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, pages 110–119, Hong Kong, December 2009. City University of Hong Kong. URL https://aclanthology.org/Y09-1013.

José Deulofeu, Lucie Duffort, Kim Gerdes, Sylvain Kahane, and Paola Pietrandrea. Depends on what the French say - spoken corpus annotation with and beyond syntactic functions. In Nianwen Xue and Massimo Poesio, editors, *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 274–281, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/W10-1843.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North*, page 1129–1141, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1116. URL http://aclweb.org/anthology/N19-1116.

Matthew S. Dryer and Martin Haspelmath, editors. *WALS Online (v2020.3)*. Zenodo, 2013. doi: 10.5281/zenodo.7385533. URL https://doi.org/10.5281/zenodo.7385533.

César Chesneau Dumarsais. *Construction*, volume 4, page 73–92. 1754.

Jay Earley. *An Efficient Context-Free Parsing Algorithm*. PhD thesis, Carnegie-Mellon, 1968.

Jan Einarsson. Talbankens talspråkskonkordans, 1976.

Jason M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL https://aclanthology.org/C96-1058.

Franti ek Čermák. Czech national corpus: A case in many contexts. *International Journal of Corpus Linguistics*, 2(2):181–197, 1997.

Alvar Ellegård. The syntactic structure of english texts: A computer-based study of four kinds of text in the brown university corpus. *(No Title)*, 1978.

Ben O Elugbe and Augusta P Omamor. Nigerian pidgin: background and prospects. ibadan. *Heine mann Educational Books*, 1991.

Nicholas Gregory Faraclas. *A grammar of Nigerian Pidgin*. University of California, Berkeley, 1989.

Meghdad Farahmand, Aaron Smith, and Joakim Nivre. A multiword expression data set: Annotating non-compositionality and conventionalization for english noun compounds. In *Proceedings of the 11th Workshop on Multiword Expressions*, pages 29–33, 2015.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English Universal Dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 22–33, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2002. URL https://aclanthology.org/K18-2002.

Ramon Ferrer-i Cancho and Carlos Gómez-Rodríguez. Anti dependency distance minimization in short sequences. a graph theoretic approach. *Journal of Quantitative Linguistics*, 28(1):50–76, January 2021. ISSN 0929-6174. doi: 10.1080/09296174.2019.1645547.

Ramon Ferrer-i Cancho and Haitao Liu. The risks of mixing dependency lengths from sequences of different length. *Glottotheory*, 5(2), January 2014. ISSN 2196-6907, 1337-7892. doi: 10.1515/glot-2014-0014. URL https://www.degruyter.com/document/doi/10.1515/glot-2014-0014/html.

Ramon Ferrer-I-Cancho and Núria Forns. The self-organization of genomes. *Complexity*, 15(5):34–36, 2009. doi: 10.1002/cplx.20296.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3): 501–538, 1988. ISSN 00978507, 15350665. URL http://www.jstor.org/stable/414531.

Jonathan G Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354. IEEE, 1997.

Ernest F Fiske. *The American heritage dictionary of the English language*. Houghton Mifflin Company, 1969.

Serge Fleury and Maria Zimina. Trameur: A framework for annotated text corpora exploration. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 57–61, 2014.

Karën Fort. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. Wiley-ISTE, July 2016. URL https://hal.science/hal-01324322.

Karën Fort and Benoît Sagot. Influence of pre-annotation on POS-tagged corpus development. In Nianwen Xue and Massimo Poesio, editors, *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/W10-1807.

Karën Fort, Adeline Nazarenko, and Sophie Rosset. Modeling the complexity of manual annotation tasks: a grid of analysis. In *Proceedings of COLING 2012*, pages 895–910, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1055.

Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. Because size does matter: The Hamburg dependency treebank. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2326–2333, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/860_Paper.pdf.

W Nelson Francis and Henry Kučera. Brown corpus manual. *Letters to the Editor*, 5 (2):7, 1979.

Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341, Aug 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1502134112.

Richard Futrell, Peng Qian, Edward Gibson, Evelina Fedorenko, and Idan Blank. Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, page 3–13, Paris, France, Aug 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-7703. URL https://www.aclweb.org/anthology/W19-7703.

Julia R. Galliers and Karen Sparck Jones. *Evaluating natural language processing system*. Technical Report TR-291. 1993.

Louis Gaultier. *Atlas de grammaire, ou tables propres à exciter et à soutenir l'attention des enfans dans l'étude de cette science.* Jules Renouard, Paris, 1817.

Kim Gerdes. Collaborative dependency annotation. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, page 88–97, Prague, Czech Republic, Aug 2013. Charles University in Prague, Matfyzpress, Prague, Czech Republic. URL https://aclanthology.org/W13-3711.

Kim Gerdes. *Same Same but Different: Paradigms in Syntax.* PhD thesis, Université Paris Nanterre, 2018.

Kim Gerdes and Sylvain Kahane. Defining dependencies (and constituents). In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, page 12, 2011. Depling.

Kim Gerdes and Sylvain Kahane. Dependency annotation choices: Assessing theoretical and practical issues of universal dependencies. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 131–140, 2016.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6008. URL https://aclanthology.org/W18-6008.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. Improving surface-syntactic universal dependencies (sud): Surface-syntactic relations and deep syntactic features. In *Proceedings of the Universal Dependencies Workshop (UDW), SyntaxFest, Paris.*, 2019a.

Kim Gerdes, Sylvain Kahane, and Xinying Chen. Rediscovering greenberg's word order universals in ud. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 124–131, 2019b.

Slavko Geršić and Gabriel Altmann. Laut – silbe – wort und das menzerathsche gesetz. In *Frankfurter phonetische Beiträge 3*, pages 115–123, Buske, Hamburg, 1980.

Hussein Ghaly. *Computational Approaches to the Syntax-Prosody Interface: Using Prosody to Improve Parsing*. PhD thesis, City University of New York, USA, 2020.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

Daniel Gildea and David Temperley. Do grammars minimize dependency length? *Cognitive Science*, 34(2):286–310, Mar 2010. ISSN 1551-6709. doi: 10.1111/j.1551-6709.2009.01073.x.

Barbara B. Greene and Gerald M. Rubin. Automatic grammatical tagging of English. Department of Linguistics, Brown University, Providence, Rhode Island, 1971.

Ralph Grishman, Catherine Macleod, and Adam Meyers. Comlex syntax: Building a computational lexicon. In *International Conference on Computational Linguistics*, 1994.

Loïc Grobol. *Coreference resolution for spoken French*. Theses, Université Sorbonne Nouvelle - Paris 3, July 2020. URL https://hal.archives-ouvertes.fr/tel-02928209.

Peter Grzybek. *History and methodology of word length studies*, page 15–90. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-4068-9. URL https://doi.org/10.1007/978-1-4020-4068-9_2.

Peter Grzybek. Close and distant relatives of the sentence: Some results from russian. In Ivan Obradović and Emmerich Kelih, editors, *Methods and applications of quantitative linguistics: selected papers of the 8th International Conference on Quantitative Linguistics (QUALICO)*, pages 44–59, Belgrade, Serbia, 2013. University of Belgrade and Academic Mind. ISBN 978-86-7466-465-0.

Gaël Guibon, Isabelle Tellier, Sophie Prévost, Mathieu Constant, and Kim Gerdes. Searching for Discriminative Metadata of Heterogenous Corpora. In Markus Dickinson, Erhard Hinrichs, Agnieszka Patejuk, and Adam Przepiórkowski, editors, *Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14)*, Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14), pages 72–82, Varsovie, Poland, December 2015. URL https://hal.archives-ouvertes.fr/hal-01250981.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, page 10, Marseille, France, 2020. URL https://www.aclweb.org/anthology/2020.lrec-1.651. LREC.

Bruno Guillaume. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-demos.21. URL https://aclanthology.org/2021.eacl-demos.21.

Bruno Guillaume and Karen Fort. Expériences de formalisation d'un guide d'annotation : vers l'annotation agile assistée. In *20e conférence sur le Traitement Automatique des Langues Naturelles*, page 628–635, Les Sables d'Olonne, France, Jun 2013. URL https://hal.archives-ouvertes.fr/hal-00840895.

Bruno Guillaume, Guillame Bonfante, Paul Masson, Mathieu Morey, and Guy Perrier. Grew : un outil de réécriture de graphes pour le TAL (Grew: a graph rewriting tool for NLP) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 5: Software Demonstrations*, pages 1–2, Grenoble, France, June 2012a. ATALA/AFCP. URL https://www.aclweb.org/anthology/F12-5001.

Bruno Guillaume, Guillame Bonfante, Paul Masson, Mathieu Morey, and Guy Perrier. Grew: un outil de réécriture de graphes pour le tal. In *Proceedings of the Joint*

Conference JEP-TALN-RECITAL 2012, volume 5: Software Demonstrations, pages 1–2, 2012b.

Bruno Guillaume, Karën Fort, and Nicolas Lefèbvre. Crowdsourcing Complex Language Resources: Playing to Annotate Dependency Syntax. In *International Conference on Computational Linguistics (COLING)*, Proceedings of the 26th International Conference on Computational Linguistics (COLING), Osaka, Japan, December 2016. URL https://hal.inria.fr/hal-01378980.

Kirian Guiller. *Analyse syntaxique automatique du pidgin-créole du Nigeria à l'aide d'un transformer (BERT): Méthodes et Résultats*. Mémoire de master, Sorbonne Nouvelle, Paris, 2020. URL http://www.tal.univ-paris3.fr/plurital/memoires/kirian_GUILLER-RD-1920.pdf.

Kristina Gulordava and Paola Merlo. Multi-lingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *TACL*, 4(0): 343–356, Jul 2016. ISSN 2307-387X. TACL.

Morgan L. Gustison, Stuart Semple, Ramon Ferrer-i Cancho, and Thore J. Bergman. Gelada vocal sequences follow menzerath's linguistic law. *Proceedings of the National Academy of Sciences*, 113(19):E2750–E2758, May 2016. doi: 10.1073/pnas.1522072113.

Jan Hajič. Building a syntactically annotated corpus: The prague dependency treebank. *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, pages 106–132, 1998.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, et al. Announcing prague czech-english dependency treebank 2.0. In *LREC*, pages 3153–3160, 2012.

Eva Hajičová. Computational linguistics without linguistics? view from prague. *Linguistic Issues in Language Technology*, 6, 2011. URL http://ufal.mff.cuni.cz/biblio/attachments/2011-hajicova-m3899101404408682562.pdf.

Zellig S Harris. From morpheme to phoneme. *Language*, 31(2):190–222, 1955.

Martin Haspelmath. The serial verb construction: Comparative concept and cross-linguistic generalizations. *Language and Linguistics*, 17(3):291–319, 2016.

Katri Haverinen, Filip Ginter, Veronika Laippala, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. A dependency-based analysis of treebank annotation errors. page 10, 2011.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.201. URL https://aclanthology.org/2021.naacl-main.201.

Johannes Heinecke. Conllueditor: a fully graphical editor for universal dependencies treebank files. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 87–93, 2019.

Gabriela Heups. Untersuchungen zum verhältnis von satzlänge zu clauselänge am beispiel deutscher texte verschiedener textklassen. *Glottometrika*, 5:113–133, 1983.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. 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)*, page 4129–4138, Minneapolis, Minnesota, Jun 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419. NAACL.

Julia Hockenmaier and Mark Steedman. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computa-*

tional Linguistics, 33(3):355–396, 2007. doi: 10.1162/coli.2007.33.3.355. URL https://aclanthology.org/J07-3004.

Yu-Siang Hong, Chen-Yu Chiang, Yih-Ru Wang, and Sin-Horng Chen. An approach to constructing prosodic grammar for mandarin read speech. *JASA*, 153(4):20, 2023.

Renkui Hou, Chu-Ren Huang, Kathleen Ahrens, and Yat-Mei Sophia Lee. Linguistic characteristics of Chinese register based on the Menzerath—Altmann law and text clustering. *Digital Scholarship in the Humanities*, 35(1):54–66, 02 2019a. ISSN 2055-7671. doi: 10.1093/llc/fqz005. URL https://doi.org/10.1093/llc/fqz005.

Renkui Hou, Chu-Ren Huang, Mi Zhou, and Menghan Jiang. Distance between chinese registers based on the menzerath-altmann law and regression analysis. *Glottometrics*, 45:24–56, 2019b. URL https://www.ram-verlag.eu/wp-content/uploads/2019/04/g45zeit-1.pdf.

Kristen Howell. *Inferring Grammars from Interlinear Glossed Text: Extracting Typological and Lexical Properties for the Automatic Generation of HPSG Grammars*. PhD thesis, 2020. URL https://digital.lib.washington.edu:443/researchworks/handle/1773/46080. Accepted: 2020-08-14T03:32:08Z.

Rodnry Huddleston and Geqffrry Pullum. The cambridge grammar of the english language. *Zeitschrift für Anglistik und Amerikanistik*, 53(2):193–194, 2005.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(03):311, Sep 2005. ISSN 1351-3249, 1469-8110. doi: 10.1017/S1351324905003840.

Luděk Hřebíček. *Text levels: language constructs, constituents and the Menzerath-Altmann law*. Wissenschaftlicher Verlag Trier, Trier, 1995. ISBN 978-3-88476-179-3.

Tomás Jelínek. Fictree: A manually annotated treebank of czech fiction. In Jaroslava Hlaváčová, editor, *Proceedings of the 17th Conference on Information Technologies –*

Applications and Theory (ITAP 2017), pages 181–185, 2017. URL http://ceur-ws.org/Vol-1885/181.pdf.

Otto Jespersen. *Analytic syntax*. Allend Unwin, Londres, 1937.

Jingyang Jiang and Haitao Liu. The effects of sentence length on dependency distance, dependency direction and the implications–based on a parallel english–chinese dependency treebank. *Language Sciences*, 50:93–104, 2015. ISSN 0388-0001. doi: https://doi.org/10.1016/j.langsci.2015.04.002.

Anders Johannsen, Dirk Hovy, and Anders Søgaard. Cross-lingual syntactic variation over age and gender. 01 2015. doi: 10.18653/v1/K15-1011.

Aravind Krishna Joshi. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? 1985.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560.

Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*, 2008.

Jan Chromý Ján Mačutek and Michaela Koščová. Menzerath-altmann law and prothetic /v/ in spoken czech. *Journal of Quantitative Linguistics*, 26(1):66–80, 2019. URL https://doi.org/10.1080/09296174.2018.1424493.

Sylvain Kahane. *How dependency syntax appeared in the French Encyclopedia: from Buffier (1709) to Beauzée (1765)*, volume 212, page 86. John Benjamins Publishing Company, February 2020. doi: 10.1075/slcs.212.04kah. URL https://hal.parisnanterre.fr/hal-04067395.

Sylvain Kahane and Kim Gerdes. *Syntaxe théorique et formelle: Volume 1: Modélisation, unités, structures*. Language Science Press, September 2022. ISBN 978-3-9855403-7-2. URL https://langsci-press.org/catalog/view/241/1923/2532-1.

Sylvain Kahane and Nicolas Mazziotta. Les corpus arborés avant et après le numérique. *Revue TAL : traitement automatique des langues*, 63(3):63–88, 2022. URL https://hal.parisnanterre.fr/hal-04074851.

Sylvain Kahane, Marine Courtin, and Kim Gerdes. Multi-word annotation in syntactic treebanks - propositions for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, page 181–189, Prague, Czech Republic, 2017a. URL https://www.aclweb.org/anthology/W17-7622.

Sylvain Kahane, Chunxiao Yan, and Marie-Amélie Botalla. What are the limitations on the flux of syntactic dependencies? evidence from ud treebanks. page 11, 2017b.

Sylvain Kahane, Martine Vanhove, Rayan Ziane, and Bruno Guillaume. A morph-based and a word-based treebank for Beja. In Association for Computational Linguistics, editor, *TLT 2021 - 20th International Workshop on Treebanks and Linguistic Theories. 21-25 March 2021, Sofia, Bulgaria*, pages 48–60, Sofia, Bulgaria, March 2022. Association for Computational Linguistics. URL https://hal.science/hal-03494462.

E. Kelih. Parameter interpretation of the menzerath law: evidence from serbian. 2010a. URL https://www.semanticscholar.org/paper/Parameter-interpretation-of-the-Menzerath-law%3A-from-Kelih/72bd3e2a86da4ffb80971b0286cc08aa5a3a1dfb.

Emmerich Kelih. *Parameter interpretation of Menzerath law: Evidence from Serbian*. Praesens Verlag, Wien, 2010b. ISBN 978-3-7069-0625-8. URL http://www.praesens.at/praesens2013/?p=1978.

Emmerich Kelih. Systematic interrelations between grapheme frequencies and word length: Empirical evidence from slovene. *J. Quant. Linguistics*, 19(3):205–231, 2012. doi: 10.1080/09296174.2012.685304. URL https://doi.org/10.1080/09296174.2012.685304.

Emmerich Kelih, Peter Grzybek, Gordana Antić, and Ernst Stadlober. Quantitative text typology: The impact of sentence length. In Myra Spiliopoulou, Rudolf Kruse, Christian Borgelt, Andreas Nürnberger, and Wolfgang Gaul, editors, *From Data and Information Analysis to Knowledge Engineering*, Studies in Classification, Data Analysis, and Knowledge Organization, page 382–389, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-31314-4. doi: 10.1007/3-540-31314-1_46.

Václava Kettnerová and Markéta Lopatková. Reflexives in Czech from a Dependency Perspective. In Kim Gerdes and Sylvain Kahane, editors, *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, Syntaxfest 2019)*, pages 14–25, Paris, France, 2019. Association for Computational Linguistics. ISBN 978-1-950737-63-5.

Chérifa Ben Khelil, Archna Bhatia, Claire Bonial, Marie Candito, Fabienne Cap, Silvio Cordeiro, Vassiliki Foufi, Polona Gantar, Voula Giouli, Najet Hadj Mohamed, Carlos Herrero, Uxoa Iñurrieta, Mihaela Ionescu, Iskandar Keskes, Alfredo Maldonado, Verginica Mititelu, Johanna Monti, Joakim Nivre, Mihaela Onofrei, Viola Ow, Carla Parra Escartín, Manfred Sailer, Carlos Ramisch, Renata Ramisch, Monica-Mihaela Rizea, Agata Savary, Nathan Schneider, Ivelina Stonayova, Sara Stymne, Ashwini Vaidya, Veronika Vincze, Abigail Walsh, and Hongzhi Xu. Parseme shared task 1.2 - annotation guidelines. https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.2/?page=040_Annotation_process_-_decision_tree, 2022. Accessed: 2023-01-27.

Philipp Koehn, Steven Abney, Julia Hirschberg, and Michael Collins. Improving intonational phrasing with syntactic information. In *2000 IEEE International*

Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100), volume 3, pages 1289–1290. IEEE, 2000.

Thomas Krause, Anke Lüdeling, Carolin Odebrecht, and Amir Zeldes. Multiple tokenizations in a diachronic corpus. In *Exploring Ancient Languages through Corpora Conference (EALC)*, volume 14, 2012.

Alexander Krotov, Mark Hepple, Robert Gaizauskas, and Yorick Wilks. Compacting the Penn Treebank grammar. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998. URL https://aclanthology.org/C98-1111.

Henry Kučera. The phonology of czech. *(No Title)*, 1961.

Agnieszka Kułacka. The coefficients in the formula for the menzerath-altmann law. *Journal of Quantitative Linguistics*, 17(4):257–268, 2010. URL https://doi.org/10.1080/09296174.2010.512160.

Agnieszka Kułacka and Ján Mačutek. A discrete formula for the menzerath-altmann law. *Journal of Quantitative Linguistics*, 14(1):23–32, 2007. URL https://doi.org/10.1080/09296170600850585.

Reinhard Köhler. Das menzeratsche gesetz auf satzebene. *Glottometrika*, (4):103–113, 1982.

Reinhard Köhler. *Das Menzerathsche Gesetz als Resultat des Sprachverarbeitungsmechanismus*, page 108–112. Georg Olms Verlag, Hildesheim, 1989. ISBN 3487091445.

Anne Lacheret-Dujour, Sylvain Kahane, and Paola Pietrandrea, editors. *Rhapsodie - A prosodic and syntactic treebank for spoken French*. John Benjamins Publishing Company, 2019.

Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Apprentissage d'analyseur en dépendances cross-lingue par projection partielle de

dépendances. In *Actes de la 23e conférence sur le Traitement Automatique des Langues Naturelles*, page 1–14, 2016.

Knud Lambrecht. A framework for the analysis of cleft constructions. *Linguistics*, 39 (3):463–516, 2001.

Heather Lent, Emanuele Bugliarello, Miryam de Lhoneux, Chen Qiu, and Anders Søgaard. On language models for creoles. In Arianna Bisazza and Omri Abend, editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 58–71, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.5. URL https://aclanthology.org/2021.conll-1.5.

Wentian Li. Menzerath's law at the gene-exon level in the human genome. *Complexity*, 17(4):49–53, March 2012. ISSN 1076-2787. doi: 10.1002/cplx.20398.

Yixuan Li, Gerdes Kim, and Dong Chuanming. Character-level annotation for chinese surface-syntactic universal dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, page 216–226, Paris, France, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-7726. URL https://www.aclweb.org/anthology/W19-7726.

Tal Linzen and Marco Baroni. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212, Jan 2021. ISSN 2333-9683, 2333-9691. doi: 10.1146/annurev-linguistics-032020-051035.

Haitao Liu. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191, Dec 2008. ISSN 1598-2327. doi: 10.17791/jcs.2008.9.2.159.

Haitao Liu. Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6):1567–1578, Jun 2010. ISSN 0024-3841. doi: 10.1016/j.lingua.2009.10.001.

Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. Structural embedding of syntactic trees for machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 815–824, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1085. URL https://aclanthology.org/D17-1085.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. dep_search: Efficient search tool for large dependency parsebanks. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 255–258, 2017.

Ján Mačutek and Andrij Rovenchak. Canonical word forms: Menzerath-altmann law, phonemic length and syllabic length. *Issues in quantitative linguistics*, 2:136–147, 2011.

Ján Mačutek, Radek Čech, and Jiří Milička. Menzerath-altmann law in syntactic dependency structure. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 100–107, Pisa,Italy, September 2017. Linköping University Electronic Press. URL https://aclanthology.org/W17-6513.

Ján Mačutek, Radek Čech, and Marine Courtin. The menzerath-altmann law in syntactic structure revisited. In *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, pages 65–73, Sofia, Bulgaria, December 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.quasy-1.6.

Pierre Magistry. *Unsupervised Word Segmentation and Wordhood Assessment*. PhD thesis, Paris Diderot ; Inria, Dec 2013. URL https://hal.archives-ouvertes.fr/tel-01573561.

Pierre Magistry and Benoît Sagot. Unsupervized word segmentation: the case for Mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–387, Jeju Island,

Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-2075.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL https://aclanthology.org/H94-1020.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2):313–330, 1993. URL https://aclanthology.org/J93-2004.

Scott Martens. Tündra: A web application for treebank search and visualization. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, page 133, 2013.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. From noisy questions to Minecraft texts: Annotation challenges in extreme syntax scenario. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 13–23, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/W16-3905.

Ján Mačutek and Gejza Wimmer. Evaluating goodness-of-fit of discrete distribution models in quantitative linguistics. *Journal of Quantitative Linguistics*, 20(3):227–240, 2013. URL https://doi.org/10.1080/09296174.2013.799912.

Arya D McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian J Mielke, Jeffrey Heinz, et al. The sigmorphon 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, 2019.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2017.

Igor Melčuk. *Dependency syntax: theory and practice*. SUNY press, 1988.

Igor Mel'čuk. Collocations and lexical functions. *Phraseology. Theory, analysis, and applications*, pages 23–53, 1998.

Paul Menzerath. *Die architektonik des deutschen wortschatzes*, volume 3. F. Dümmler, 1954.

Georgios Mikros and Jiří Milička. *Distribution of the Menzerath's law on the syllable level in Greek texts*, page 181–189. RAM-Verlag, Lüdenscheid, 2014.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6301. URL https://www.aclweb.org/anthology/D19-6301.

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956. ISSN 1939-1471. doi: 10.1037/h0043158.

Friedrich Max Müller. *The first book of the Hitopadeśa: containing the Sanskrit text, with interlinear transliteration, grammatical analysis, and English translation*, volume 1. Longman, Green, Longman, Roberts, & Green, 1864.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. CorefUD 1.0: Coreference meets Universal Dependencies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.520.

Jens Nilsson, Johan Hall, and Joakim Nivre. Mamba meets tiger: Reconstructing a treebank from antiquity. page 15, 2005.

Joakim Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France, April 2003. URL https://aclanthology.org/W03-3017.

Joakim Nivre. *Treebanks*, page 225–241. Mouton de Gruyter, 2008a. URL http://stp.lingfil.uu.se/~nivre/docs/hsk.pdf.

Joakim Nivre. Treebanks. In Merja Kytö and Anke Lüdeling, editors, *Corpus Linguistics: An International Handbook*, pages 225–241. Mouton de Gruyter, 2008b. URL http://stp.lingfil.uu.se/~nivre/docs/hsk.pdf.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, and Natalia Silveira. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane

Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne,

Lng Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. Universal dependencies 2.3, 2018. URL http://hdl.handle.net/11234/1-2895. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Pavel Novak and Petr Sgall. On the prague functional approach. *Travaux linguistiques*

de Prague, 3:291–297, 1968.

Stephan Oepen, Lilja Øvrelid, Jari Björne, Richard Johansson, Emanuele Lapponi, Filip Ginter, and Erik Velldal. The 2017 shared task on extrinsic parser evaluation. towards a reusable community infrastructure. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies. Pisa, Italy*, pages 1–16, 2017.

S Olaiju and Akeem A Taiwo. Steps problem: the link between combinatoric and k-bonacci sequences. *European Journal of Statistics and Probability*, 3(4):10–19, 2015.

Mai Omura, Aya Wakasa, and Masayuki Asahara. Word delimitation issues in ud japanese. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 142–150, 2021.

Timothy Osborne and Kim Gerdes. The status of function words in dependency grammar: A critique of universal dependencies (ud). *Glossa: a journal of general linguistics*, 4(11), January 2019a. ISSN 2397-1835. doi: 10.5334/gjgl.537. URL https://www.glossa-journal.org/article/id/5124/.

Timothy Osborne and Kim Gerdes. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: a journal of general linguistics (2016-2021)*, January 2019b. doi: 10.5334/gjgl.537. URL https://hal.inria.fr/hal-02450313.

Timothy Osborne, Michael Putnam, and Thomas Groß. Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, 15(4):354–396, 2012. ISSN 1467-9612. doi: https://doi.org/10.1111/j.1467-9612.2012.00172.x.

Jarmila Panevová. On verbal frames in functional generative description. *Prague Bulletin of Mathematical Linguistics*, 22(3-40):6–3, 1974.

Thiago Alexandre Salgueiro Pardo, Magali Sanches Duran, Lucelene Lopes, Ariani Di Felippo, Norton Trevisan Roman, and Maria das Graças Volpe Nunes. Porttinari - a large multi-genre treebank for brazilian portuguese. In *Anais do Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)*, page 1–10. SBC, Nov 2021. doi: 10.5753/stil.2021.17778. URL https://sol.sbc.org.br/index.php/stil/article/view/17778.

Patrick Paroubek, Stéphane Chaudiron, and Lynette Hirschman. Principles of Evaluation in Natural Language Processing. *Revue TAL*, 48(1):7–31, May 2007. URL https://hal.archives-ouvertes.fr/hal-00502700.

Kateřina Pelegrinová, Ján Mačutek, and Radek Čech. The menzerath-altmann law as the relation between lengths of words and morphemes in czech. *Journal of Linguistics/Jazykovedný casopis*, 72(2):405–414, 2021. URL https://doi.org/10.2478/jazcas-2021-0037.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf.

Paola Pietrandrea and Sylvain Kahane. *Macrosyntactic annotation*, volume 89, page 97–126. John Benjamins Publishing Company, June 2019. doi: 10.1075/scl.89.07pie. URL https://hal.parisnanterre.fr/hal-04088576.

Barbara Plank, Dirk Hovy, and Anders Søgaard. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2083. URL https://aclanthology.org/P14-2083.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2067. URL https://aclanthology.org/P16-2067.

Ioan-Iovitz Popescu. *Word Frequency Studies*. De Gruyter Mouton, Berlin, New York, 2009. ISBN 9783110218534. URL https://doi.org/10.1515/9783110218534.

Marta Recasens and Eduard Hovy. A deeper look into features for coreference resolution. In Sobha Lalitha Devi, António Branco, and Ruslan Mitkov, editors, *Anaphora Processing and Applications*, pages 29–42, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04975-0.

Alonzo Reed and Brainerd Kellogg. *Graded lessons in English. An elementary English grammar [...]*. Clark and Maynard, New York, 1876.

Alonzo Reed and Brainerd Kellogg. *Higher lessons in English. A work on grammar and composition [...]*. Clark and Maynard, New York, 1877.

Ines Rehbein, Julius Steen, Bich-Ngoc Do, and Anette Frank. Universal Dependencies are hard to parse – or are they? In Simonetta Montemagni and Joakim Nivre, editors, *Proceedings of the Fourth International Conference on Dependency Linguistics*

(Depling 2017), pages 218–228, Pisa, Italy, September 2017. Linköping University Electronic Press. URL https://aclanthology.org/W17-6525.

Hue San Do Renkui Hou, Chu-Ren Huang and Hongchao Liu. A study on correlation between chinese sentence and constituting clauses based on the menzerath-altmann law. *Journal of Quantitative Linguistics*, 24(4):350–366, 2017. URL https://doi.org/10.1080/09296174.2017.1314411.

Douglas LT Rohde. Tgrep2 user manual. *Unpublished manuscript*, 2005. URL http://www.cs.cmu.edu/afs/cs.cmu.edu/project/cmt-55/OldFiles/lti/Courses/722/Spring-08/Penn-tbank/Tgrep2/tgrep2_manual.pdf.

John Robert Ross. *Constraints on variables in syntax.* Thesis, Massachusetts Institute of Technology, 1967.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, page 1–15, Berlin, Heidelberg, 2002. Springer. ISBN 978-3-540-45715-2. doi: 10.1007/3-540-45715-1_1.

Gözde Gül Şahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1545. URL https://aclanthology.org/D18-1545.

Geoffrey Sampson. *Thoughts on Two Decades of Drawing Trees*, volume 20 of *Text, Speech and Language Technology*, page 23–41. Springer Netherlands, Dordrecht, 2003. ISBN 978-1-4020-1335-5. doi: 10.1007/978-94-010-0201-1_2. URL http://link.springer.com/10.1007/978-94-010-0201-1_2.

Felix Sasaki, Andreas Witt, and Dieter Metzing. Declarations of relations, differences and transformations between theory-specific treebanks: A new methodology. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden, 2003. Växjö University Press. URL https://ids-pub.bsz-bw.de/frontdoor/deliver/index/docId/4544/file/Sasaki_Witt_Metzing_Declarations_of_Relations_Differences_and_Transformations_2003.pdf.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, and Ivelina Stoyanova. The parseme shared task on automatic identification of verbal multiword expressions. In *MWE2017 - Proceedings of the 13th Workshop on Multiword Expressions , Apr 2017, Valencia, Spain.*, 2017.

Nathan Schneider. What I've learned about annotating informal text (and why you shouldn't take my word for it). In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 152–157, Denver, Colorado, USA, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-1618. URL https://aclanthology.org/W15-1618.

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, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1376.

Michael H. Schwibbe. Text- und wortstatistische untersuchungen zur validität der menzerath'schen regel. *Glottometrika*, (6):127–138, 1984.

Djamé Seddah, Farah Essaidi, Amal Fethi, Matthieu Futeral, Benjamin Muller, Pedro Javier Ortiz Suárez, Benoît Sagot, and Abhishek Srivastava. Building a user-generated content north-african arabizi treebank: Tackling hell. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguis-*

tics, page 1139–1150, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.107. URL https://www.aclweb.org/anthology/2020.acl-main.107.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Fifth Conference on Applied Natural Language Processing*, pages 88–95, Washington, DC, USA, March 1997. Association for Computational Linguistics. doi: 10.3115/974557.974571. URL https://aclanthology.org/A97-1014.

Han Sloetjes and Peter Wittenburg. Annotation by category: ELAN and ISO DCR. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/208_paper.pdf.

Vladimír Šmilauer. Novočeská skladba (syntax of modern czech). *Prague: Academia*, 1947.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Lynne M. Stallings, Maryellen C. MacDonald, and Padraig G. O'Seaghdha. Phrasal ordering constraints in sentence production: Phrase length and verb disposition in heavy-np shift. *Journal of Memory and Language*, 39(3):392–417, October 1998. ISSN 0749596X. doi: 10.1006/jmla.1998.2586.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

James Strong. *The exhaustive concordance of the Bible : showing every word of the text of the common English version of the canonical books, and every occurrence of each word in regular order : together with A comparative concordance of the Authorized and Revised versions, including the American variations : also brief dictionaries of the Hebrew and Greek words of the original, with references to the English words.* Jennings & Graham, 1890.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, page 5027–5038. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1548. URL http://aclweb.org/anthology/D18-1548.

Maria Helena Svensson. A very complex criterion of fixedness: Non-compositionality. *Phraseology: An interdisciplinary perspective. John Benjamins Publishing Company*, 2008.

Anders Søgaard. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, page 682–686. Association for Computational Linguistics, 2011.

Sali A Tagliamonte. *Variationist sociolinguistics: Change, observation, interpretation*, volume 40. Wiley-Blackwell, 2012.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In Chengqing

Zong and Michael Strube, editors, *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 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://aclanthology.org/P15-1150.

Ulf Teleman. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, 1974.

Isabelle Tellier, Iris Eshkol, Samer Taalab, and Jean-Philippe Prost. Pos-tagging for oral texts with crf and category decomposition. *Research in Computing Science*, 46: 79–90, 2010.

David Temperley. Dependency-length minimization in natural and artificial languages. *Journal of Quantitative Linguistics*, 15(3):256–282, Aug 2008. ISSN 0929-6174. doi: 10.1080/09296170802159512.

Lucien Tesnière. Comment construire une syntaxe. *Bulletin de la Faculté des Lettres de Strasbourg*, (7):219–229, 1934.

Lucien Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1959.

Regina Teupenhayn and Gabriel Altmann. Clause length and menzerath's law. *Glottometrika*, (6):152–176, 1984.

John Tinsley, Mary Hearne, and Andy Way. Exploiting parallel treebanks to improve phrase-based statistical machine translation. In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009, Mexico City, Mexico, March 1-7, 2009. Proceedings*, volume 5449 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2009. doi: 10.1007/978-3-642-00382-0_26. URL https://doi.org/10.1007/978-3-642-00382-0_26.

Eric Tjong Kim Sang. Np bracketing. URL https://www.clips.uantwerpen.be/conll99/npb/.

Masaru Tomita, editor. *Proceedings of the First International Workshop on Parsing Technologies*, Pittsburgh, Pennsylvania, USA, August 1989. Carnegy Mellon University. URL https://aclanthology.org/W89-0200.

Iván G. Torre, Bartolo Luque, Lucas Lacasa, Christopher T. Kello, and Antoni Hernández-Fernández. On the physical origin of linguistic laws and lognormality in speech. *Royal Society Open Science*, 6(8):191023, 2019. doi: 10.1098/rsos.191023. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsos.191023.

Ke Tran and Arianna Bisazza. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In Colin Cherry, Greg Durrett, George Foster, Reza Haffari, Shahram Khadivi, Nanyun Peng, Xiang Ren, and Swabha Swayamdipta, editors, *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6132. URL https://aclanthology.org/D19-6132.

Francis M. Tyers, Maria Sheyanova, and Jonathan North Washington. Ud annotatrix: an annotation tool for universal dependencies. In *Proceedings Of The 16th International Workshop On Treebanks And Linguistic Theories.*, 2017.

Zdenka Uresova, Ondřej Dušek, Eva Fucíková, Jan Hajič, and Jana Šindlerová. Bilingual english-czech valency lexicon linked to a parallel corpus. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 124–128, 2015.

Giulia Venturi, Tommaso Bellandi, Felice Dell'Orletta, and Simonetta Montemagni. NLP–based readability assessment of health–related texts: a case study on Italian informed consent forms. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 131–141, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-2618. URL https://aclanthology.org/W15-2618.

Ngoc Phuoc An Vo and Octavian Popescu. Learning the impact and behavior of syntactic structure: A case study in semantic textual similarity. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 688–696, Hissar, Bulgaria, September 2015. INCOMA Ltd. Shoumen, BULGARIA. URL https://aclanthology.org/R15-1088.

Ilaine Wang. *Syntactic Similarity Measures in Annotated Corpora for Language Learning: Application to Korean Grammar*. PhD thesis, 2017. URL http://www.theses.fr/2017PA100092/document.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. Cross-lingual BERT transformation for zero-shot dependency parsing. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1575. URL https://aclanthology.org/D19-1575.

Henri Weil. *De l'ordre des mots dans les langues anciennes comparées aux langues modernes*. Crapelet, Paris, 1844.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitchell P Marcus. Developing guidelines and ensuring consistency for chinese text annotation. In *LREC*, 2000.

Lirong Xu and Lianzhen He. Is the menzerath-altmann law specific to certain languages in certain registers? *Journal of Quantitative Linguistics*, 27(3):187–203, 2020a. doi: 10.1080/09296174.2018.1532158. URL https://doi.org/10.1080/09296174.2018.1532158.

Lirong Xu and Lianzhen He. Is the menzerath-altmann law specific to certain languages in certain registers? *Journal of Quantitative Linguistics*, 27(3):187–203, 2020b. URL https://doi.org/10.1080/09296174.2018.1532158.

Victor H. Yngve. A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104(5):444–466, 1960. ISSN 0003-049X.

Amir Zeldes. The gum corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612, September 2017a. ISSN 1574-0218. doi: 10.1007/s10579-016-9343-x.

Amir Zeldes. The gum corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612, 2017b.

Amir Zeldes, Anke Lüdeling, Julia Ritz, and Christian Chiarcos. Annis: A search tool for multi-layer annotated corpora. In *In: Proceedings of Corpus Linguistics 2009, Liverpool, July 20-23, 2009.* Humboldt-Universität zu Berlin, Philosophische Fakultät II, 2009.

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 Hajič, 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 Daniel Zeman and Jan Hajič, editors, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2001. URL https://aclanthology.org/K18-2001.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana

Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Ethan Chi, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva,

Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravis-

hankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Steinþór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. Universal dependencies 2.7, 2020. URL http://hdl.handle.net/11234/1-3424. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Hongxin Zhang and Haitao Liu. *Interrelations among Dependency Tree Widths,*

Heights and Sentence Lengths, pages 31–52. De Gruyter Mouton, Berlin, Boston, 2018. ISBN 9783110573565. doi: doi:10.1515/9783110573565-002. URL https://doi.org/10.1515/9783110573565-002.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China. URL https://aclanthology.org/2021.ccl-1.108.

Titre Des corpus arborés à l'induction de structures syntaxiques partielles.

Résumé
Nos travaux portent sur les *treebanks*, ces corpus de textes dotés d'annotations de structures syntaxiques. Ils sont très utiles dans de nombreux domaines, de la linguistique au traitement automatique de la langue. Après une introduction portant sur leur rôle dans des domaines variés, nous plongeons dans l'histoire de leur création, depuis les pratiques d'annotation manuelle de textes vers les treebanks modernes avec l'avènement des technologiques. Le chapitre 3 montre les méthodes de création de ces treebanks. Le chapitre 4 discute des problématiques liées à la constitution des guides d'annotation, et mets en évidences certaines de ces problématiques au travers de deux études, la première portant sur traitement des expressions multi-mots, la seconde sur la constitution d'un treebank dans une langue peu pourvue en ressources, le Naija langue parlée au Nigéria étudiée dans le cadre du projet ANR NaijaSynCor. Le chapitre 5 présente l'outil Arborator-Grew, conçu pour faciliter l'annotation collaborative des treebanks. Le chapitre 6 étudie comment des lois linguistiques fondamentales comme la loi de *Menzerath-Altmann* et le *Heavy Constituent Shift* interagissent. Il propose également plusieurs procédures pour générer des arbres artificiels, permettant de contraster leurs propriété avec celles des arbres syntaxiques. Enfin, le chapitre 7 vise à utiliser des techniques statistiques pour découvrir la structure sous-jacente des phrases dans un texte. En résumé, ce travail montre l'importance des treebanks dans notre compréhension des langues, et leur rôle dans le développement des technologies linguistiques en soulignant l'innovation continue dans ce domaine.

Mots-clés: Traitement Automatique de la Langue, corpus arborés, syntaxe de dépendance, annotation syntaxique.

Title From treebanks to partial syntactic structure induction.

Abstract
This document focuses on *treebanks*, textual corpora with syntactic annotations. These treebanks are invaluable in numerous fields, ranging from linguistic studies to natural language processing. First, we explore how these treebanks aid researchers in multiple domains. Next, we dive into the history of treebank development. Before the computer age, researchers began to manually create collections of annotated texts, which evolved with the advent of computers into modern treebanks. Chapter 3 focuses on the challenges and methods of creating these treebanks. Chapter 4 addresses challenges relating to developing annotation guidelines. These discussions bring us to two case studies, the first relating to how to best handle complex multi-word expressions, the second retracing the development of a treebank for a low-resource language, the Naija pidgin-creole spoken spoken in Nigeria and its analysis as part of the ANR NaijaSynCor project. Chapter 5 introduces a new tool, Arborator-Grew, designed to facilitate collaborative annotation of treebanks. Chapter 6 studies how linguistic laws such as the *Menzerath-Altmann* law and the *Heavy Constituent Shift* interact. It also introduces several tree generation algorithms, which we use to contrast the properties of syntactic and artificial trees. Finally, Chapter 7 aims to use statistical techniques to latent structure of sentences in a text. In summary, this work highlights the importance of treebanks in our understanding of languages and their significant role in the development of language technologies. It also emphasizes continuous innovation in this field, opening new avenues for the study and analysis of languages.

Key-words: Natural Language Processing, treebanks, dependency syntax, syntactic annotation.

Université Sorbonne Nouvelle

ED 622 *622 Sciences du langage* - ed622@sorbonne-nouvelle.fr