Towards the automatic detection of syntactic differences

Kroon, M.S.

Towards the Automatic Detection of Syntactic Differences

Published by

LOT
Binnengasthuisstraat 9
1012 ZA Amsterdam
The Netherlands

phone: +31 20 525 2461
e-mail: lot@uva.nl
http://www.lotschool.nl

Cover illustration: Kalle Wolters

ISBN: 978-94-6093-414-8
DOI: https://dx.medra.org/10.48273/LOT0629
NUR: 616

Towards the Automatic Detection of Syntactic Differences

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof. dr. ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op donderdag 10 november 2022
klokke 10.00 uur

door

Martin Siebren Kroon

geboren 6 november 1993
te Groningen, Nederland

Promotores:	Prof. dr. L.C.J. Barbiers
	Prof. dr. J.E.J.M. Odijk (Universiteit Utrecht)
Co-promotor:	Dr. S.L. van der Pas (Amsterdam UMC)
Promotiecommissie:	Prof. dr. S.A. Raaijmakers
	Prof. dr. ir. J. Nerbonne (Rijksuniversiteit Groningen)
	Prof. dr. P.D. Grünwald
	Prof. dr. G.J.M. van Noord (Rijksuniversiteit Groningen)
	Dr. J. Prokic

The research reported in this thesis was funded by the Data Science Research Programme and the Faculty of Humanities of Leiden University.

Dedicated to my dear parents.

Contents

Acknowledgements

There are many people I owe thanks to, either for their direct involvement in my research endeavours during my PhD, for supporting me outside of my work, or even by simply existing. In fact, everyone that I had the pleasure of talking to ever since I moved to Leiden in 2017, I am to some extent indebted to. However, a few people I am particularly thankful to.

First and foremost I want to thank my three brilliant supervisors Sjef, Jan and Stéphanie, who have advised and guided me throughout this sometimes difficult project. You have taught me many things, and I could not have been luckier with you as my supervisors. Our collaboration was very enjoyable to me and I hope to be working together again on a new project at some point in the future.

Then, I want to thank my sweet parents, who have supported and loved me unconditionally – not only during my time in Leiden, but all my life. I count myself lucky with you as my parents and I treasure the way you have raised me. As my great-grandfather would have said in his native Bolserters: *kist op un minder like*.

I also want to thank my big brother Peter, who has often had the time to distract me with fun games as well as to stimulate and motivate me to perform research and to always stay curious. You have often been an example to me, and I am proud to be your little brother.

A great deal of thanks I owe to my Pascale. Your love and support seem unending and make me a very happy man indeed, even during the dark days of the pandemic.

A big thank you also goes out to my friends, who have proven to be fantastic people by keeping me sane during my time off. In particular Max, with whom I have learned to check whether people actually left before taking their cheese platter; Laura odB., for introducing me to Leiden and making me feel at home in a city I didn't know; Xander, Lotte, Laura D., and Vera, with whom I have developed a nearly cultist fascination with a type of Cypriot cheese; Hanjo and Froos Bommee, who have led my linguistic creativity into a wanton subversion

of reality; and Timo, Susan and Janoël, who even after all those years remain among my closest of friends.

I was also kept relatively sane during my time at work when the country wasn't in lockdown, so I am also very grateful to all of my colleagues at LUCL, my colleagues of the DSO and my colleagues of the LUCDH, chief among which are Xander and Lis (who I am also honoured and grateful to call my paranymphs), Laura odB., Manolis and all the data-science PhDs. Special thanks go out to Astrid and Anikó, who helped me analyse some of my data, and to the staff of ÚFAL of the Charles University in Prague, who have so graciously hosted me in the autumn of 2019.

Finally, I want to thank Toneelgroep Imperium – especially those with whom I've worked together closely during productions. Thank you for accepting me into your theatrical family with open arms. I thoroughly enjoy spending time and sharing the spotlight with you, and hope to continue to do so for many years to come.

Glossary of abbreviations, acronyms and notations

AAA	Affix-Attribute Associator
ADJ	adjective (UD tag)
ADP	adposition (UD tag)
ADV	adverb (UD tag)
`advmod`	adverbial modifier (UD relation)
AL list	the list containing characteristic differences between Hungarian and English as compiled by dr. Lipták
`amod`	adjectival modifier (UD relation)
Art	article (UD feature value)
AUC	area under the (ROC) curve
`aux`	auxiliary verb (UD relation)
AUX	auxiliary verb (UD tag)
`cc`	coordinating conjunction (UD relation)
`ccomp`	clausal complement (UD relation)
CCONJ	coordinating conjunction (UD tag)
`conj`	conjunct (UD relation)
CoNLL-U	a specific annotation format; short for *Conference on Computational Natural Language Learning–Universal Dependencies*
CS	Czech
DE	German
Def	definite (UD feature value)
Definite	definiteness (UD feature)
deprel	dependency relation

DeSDA	Detecting Syntactic Differences Automatically
`det`	determiner (UD relation)
DET	determiner (UD tag)
DGAE	Data Grouper for Attribute Exploration
EN	English
EU	European Union
Fin	finite (UD feature value)
GED	graph-edit distance
Gen	genitive (UD feature value)
GTI	Generalization Tree Inducer
Hn	hypothesis number n, referring to Table 4.3
HU	Hungarian
Ind	1. indefinite (UD feature value); 2. indicative (UD feature value)
Inf	infinite (UD feature value)
Intrans	intransitive
KJB	King James Bible
Lev.	Levenshtein distance
MDL	Minimum Description Length
Mn	missed difference number n, referring to Table 4.4
N	noun
n-gram	a contiguous sequence of n items
NL	Dutch
NLP	natural language processing
`nmod`	nominal modifier (UD relation)
NP	noun phrase
`nsubj`	nominal subject (UD relation)
`nsubj:pass`	passive nominal subject (UD relation)
Num	number (UD feature)
NUM	numeral (UD tag)
`obj`	object (UD relation)
`obl`	oblique nominal (UD relation)
OSV	object-subject-verb order
OVS	object-verb-subject order
Part	participle (UD feature value)
PART	particle (UD tag)

Plur	plural number (UD feature value)
pmi	pointwise mutual information
POS	part of speech
PRON	pronoun (UD tag)
PronType	pronoun type (UD feature)
PROPN	proper noun (UD tag)
Prs	personal or possessive personal pronoun or determiner (UD feature value)
PRS	present tense
PST	past tense
PUNCT	punctuation (UD tag)
ROC curve	receiver operating characteristic curve
S	sentence
SCONJ	subordination conjunction (UD tag)
Sent. length	sentence length
Sing	singular number (UD feature value)
sg.	singular number
SOV	subject-object-verb order
SQS	'Summarising event seQuenceS'; Tatti and Vreeken (2012)
SV	subject-verb order
SVO	subject-verb-object order
Trans	transitive
UD	Universal Dependencies; Nivre et al. (2016)
V	verb
V2	verb-second word order
VB	Vizsoly Bible
VBP	verb, non-3rd person singular present (Penn Treebank tag)
VO	verb-object order
VP	verb phrase
WEB	World English Bible
wh	question-, interrogative
X	other (UD tag)
<…>	used to identify an individual grapheme or character
$x\rangle\ldots\langle y$	used to mark off circumfixes

CHAPTER 1

Introduction

Natural language syntax is the system of combinatorial rules that builds complex hierarchical structures, i.e. phrases and clauses, out of individual words and morphemes.[1] The insight that the words of a sentence are organized both linearly and hierarchically, i.e. as phrases that contain phrases that contain phrases, is central in modern linguistics (cf., e.g., Berwick and Chomsky 2016).

Superficial comparison of different languages suggests that their syntax may differ immensely: for instance, variation in word order (e.g. English *the intelligent girl* vs. French *la fille intelligente*), variation in the absence or presence of a morpheme, word or phrase (e.g. English *I eat* vs. Italian *mangio*), doubling of grammatical features (e.g. English *big houses* vs. Finnish *isot talot*),[2] or variation in the morpho-syntactic expression of grammatical relations such as agreement between the subject and the finite verb (e.g. English *I walk, we walk* vs. Dutch *ik loop, wij lopen*). Yet, the syntactic literature gives numerous arguments to support the hypothesis that all human languages share the same abstract set of syntactic principles. The main aim of theoretical comparative syntactic research is to identify the range, limits and locus of syntactic variation

[1] A morpheme is the smallest lexical unit that bears meaning in a language. For example, the word *bears* is built from the morphemes *bear* and *-s*, which can itself not be divided into smaller, meaningful parts.

[2] For clarity: the plural marker *-t* is present on both the adjective and the noun in Finnish, whereas the English plural marker *-s* is present only on the noun.

between natural languages by comparing their structures and describing the syntactic similarities and differences, and to capture them in a cross-linguistic formal theory (Cinque and Kayne 2005). The research field endeavours to find answers to questions such as: what is an (im)possible natural language, which syntactic properties are universal and which are language-specific, and is the syntactic variation a property of the component of the mental grammar that builds hierarchical structures, or is it possible to reduce the syntactic variation to other modules of the grammar such as the lexicon and the module that takes care of phonological spell-out and linearization?

It is common practice for syntacticians to compare their native language with other languages by referencing detailed grammars and other linguistic literature, as well as consulting with fellow linguists. With the enormous number of natural languages and dialects (estimates commonly arrive at around 5000 to 7000 spoken languages, excluding their often numerous dialects (cf., e.g., Eberhard, Simons and Fennig 2021)), the very high level of variation they exhibit between one another (even between closely related languages or dialects; cf. Barbiers et al. 2005/2008, who describe more than 100 syntactic differences within Dutch dialects alone, which are generally very similar and closely related to one another), and the technically infinite number of possible sentences per language or dialect of which the linguist needs to make a selection to be investigated, systematic comparison is a hugely daunting task.

As a result of this, syntacticians may leave many differences and associations between them undetected, and formal descriptions of language incomplete. The field would therefore significantly benefit from the (partial) automatization of the process, as it would increase the scale, speed, systematicity and reproducibility of research. The computer can process and analyse much more material on many more languages in a much more systematic way, which makes it more likely that new variation will be discovered, including correlations between variables that may be reducible to more abstract underlying syntactic properties. However, the question remains: **can syntactic differences between languages be detected automatically, and if so, how?**

1.1 Background

There has not been much research into the automatic detection of syntactic differences, but all researches have in common that they rely on the availability of sophisticated Natural Language Processing (NLP) tools. An important, early contribution was made by Nerbonne and Wiersma (2006) and Wiersma, Nerbonne and Lauttamus (2011), who devised a method based on word-category labels, called part-of-speech tags, or POS tags, to select on statistical grounds hypotheses about related dialects and language varieties for further investigation. In general, these POS tags can be as simple as N for nouns and A for adjectives, or be more detailed such as VBP for non-3rd person singular present verbs (Taylor, Marcus and Santorini 2003), with POS tag sets usually

consisting of 15 to 25 predefined distinct tags. The method devised by Nerbonne and Wiersma (2006) consists of taking POS-tag sequences of varying lengths (called POS n-grams, in which the n stands for the length of the sequence, i.e. the number of tags in it) from two comparable corpora from the same language. After that, the relative frequencies of the POS n-grams are compared using a permutation test and the statistically significant ones are sorted by degree of difference. In their paper, they demonstrated the utility of their approach by detecting syntactic differences between the English of two generations of Finnish immigrants to Australia (Nerbonne and Wiersma 2006). The method proposed by Nerbonne and Wiersma (2006) requires the user to commit to a specific length of POS n-grams, which limits the number of types of differences that can be found. It is furthermore designed to compare variants (mostly sociolects) of the same language.

Nerbonne et al.'s (2006) method was extended by Sanders (2007), who added syntactic hierarchy to the analysis, using the leaf-ancestor path representation of syntactic parse trees developed by Sampson (2000) instead of POS n-grams.[3] Syntactic trees reflect the hierarchical structure of a sentence, and are typically constructed either by (recursively) grouping words that form constituents or phrases (leading to a constituency tree), or by connecting every word to its dependents (leading to a dependency tree). Sampson's (2000) leaf-ancestor path represents the structure of a constituency tree by deriving the path from the root of the tree (usually a node labelled with S) to each word in the sentence. Let us for example consider the sentence *The dog barks*, of which a parsed version can be found in Figure 1.1. From it, one can extract the leaf-ancestor paths *S-NP-Det-The*, *S-NP-N-dog* and *S-VP-V-barks*. Sanders (2007) applies this method to find dialectical variation between several British regions, and reports that his method is successful in detecting differences between corpora divided on geographical area (although not showing specific types of differences that can be detected with it, and only reporting on statistical significance between regions), rather than on language proficiency as Nerbonne and Wiersma (2006) do. Still, it is designed to work on variants of the same language. Apart from that, the leaf-ancestor nodes do not take into account the strict (linear or structural) contexts of words within each sentence, nor do they indicate the syntactic function or relation of a phrase within a sentence, making the method unable to detect certain types of differences.

In his PhD dissertation Sanders (2010) adapts Nerbonne and Wiersma's (2006) work for syntactic dialectometric research. Other works mainly focus on measuring the syntactic distance between language varieties and dialects, and do not particularly aim to extract the actual syntactic differences in question. For instance, Spruit (2008) relates binary syntactic features to geographical distance as given by Barbiers et al. (2005/2008) in order to measure the distances between Dutch dialects.

[3] Strictly speaking, the term *leaf-ancestor path* is not entirely correct, because the paths represent the path from the ancestor to the leaf; *ancestor-leaf path* would perhaps have been more appropriate.

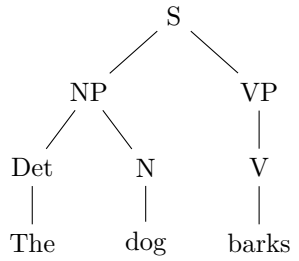


Figure 1.1: A syntactic (constituency) tree of the sentence *The dog barks* after Sanders (2007). From it, Sanders extracts leaf-ancestor paths *S-NP-Det-The*, *S-NP-N-dog* and *S-VP-V-barks*.

The introduction of *semantic maps* by Haspelmath (1997; 2003), and later adaptation by Cysouw (2010), allows to graphically represent different uses and meanings of grammatical constructions throughout languages, illustrating how they relate to one another. For instance, a semantic map of the dative construction in English and French illustrates straightforwardly that English *to* and French *à* are not used in the exact same situations (whereas *to* can be used to express purpose, *à* cannot; conversely, *à* can be used to express predicative possessors, which *to* cannot; Haspelmath 2003). With these semantic maps usually being driven by predefined sets of usages and instantiations, van der Klis, Le Bruyn and De Swart (2017) introduced *translation mining*, a method to automatically construct semantic maps from the data, and mapped the different uses of the perfect tense between German, English, Spanish, French and Dutch, by aligning instances of the perfect tense across the languages and calculating a dissimilarity matrix based on the verb forms used in the translations. This allowed them to investigate the use of the perfect between those languages more closely, and to reproduce previous research as well as to report on new findings with respect to the tense/aspect role of the perfect. Their method, however, requires the user to manually correct or handpick the data used, and makes no use of automatic annotation of the data.

Another recent contribution was made by Wong et al. (2017), who present a method to quantitatively investigate a parallel treebank. A treebank is a corpus of parsed sentences (i.e. syntactic trees) and plays a crucial role in NLP. A parallel treebank is then a treebank in which every parsed sentence is aligned to a parsed translation in another language. Using a treebank with dependency parses, Wong et al. (2017) extract some differences between Mandarin and Cantonese by calculating which POS tags or syntactic-relation labels are under- or overrepresented in either language; they for instance find that punctuation marks and particles (or rather, words labelled as such) are overrepresented in Cantonese, while the opposite is true for adverbs and adpositions. They also investigate very local structural contexts by calculating the under- or overrepresentation of parts of the trees, such as a POS tag with the de-

pendency relation to their head (i.e. the syntactic relation that they have to their mother node in the tree) and two POS tags with the dependency relation between them. These local structural contexts e.g. suggest that subject pronouns are more prevalent in Mandarin than in Cantonese. However, their method does not explicitly leverage the parallellity of the treebank to identify in which contexts differences might occur, and only aggregates the data and the POS frequencies by not looking at each sentence pair individually. The lack of any representation of the linear order of words to each other furthermore makes it impossible for Wong et al. (2017) to detect simple word order differences. Apart from that, only considering POS tags and syntactic-relation labels limits the access to relevant morphological features and differences regarding them.

To the best of the author's knowledge, it seems that most relevant work can be characterized as

A detecting specific differences but without syntactic relations and without parallel data (Nerbonne and Wiersma 2006; Sanders 2007);

B using parallel data, but manually corrected or handpicked (Spruit 2008; Sanders 2010; Cysouw 2010; Wälchli 2010; van der Klis, Le Bruyn and De Swart 2017), all of which also fall under C;

C mainly focusing on visualizing or quantitatively summarizing linguistic variation without the distinct intent of identifying the syntactic differences or the contexts in which they occur (also Wong et al. 2017);

D measuring syntactic distance between language variations or dialects for dialectometric purposes (among others Sanders 2010; Spruit 2008).

The research put forth in this dissertation differs from the works discussed above in that it has the express goal of detecting syntactic differences in as wide a range as possible (i.e. aiming for generality, not being constrained to a specific difference) and that it presumes parallel corpora. A parallel corpus is a collection of texts in multiple languages where each sentence is aligned to its translation(s), resulting in sentence pairs (or sometimes triples or more, depending on the number of languages included in the corpus) that have the same meaning throughout the languages. These parallel corpora provide excellent data for automatic syntactic comparison, mimicking the case of manual comparative syntactic research, where syntacticians often compare a sentence with its translations. Massively parallel texts are an important addition to the kinds of data used in linguistic typology, such as reference grammars, dictionaries and field work or questionnaires (among others Cysouw and Wälchli 2007; Dahl 2007; Wälchli 2007).

Wälchli (2007) discusses several advantages and disadvantages of using parallel corpora. The most important advantage is that using parallel corpora allows for the direct comparison of concrete examples across languages, because every sentence and its translation are instantiations in the same textual context, with the same emphasis, and in the same register. In a parallel corpus the

researcher can also identify in which structural or syntactic contexts the differences occur, whereas in a non-parallel corpus only quantitative differences can be measured. Wälchli notices that it is, in general, much easier to work with one parallel corpus, as opposed to two non-parallel corpora, because the meaning is the same throughout both texts and because the structure of the text allows the linguist to investigate a small number of sentences or fragments selectively that are directly relevant to the research question. Wälchli furthermore argues that parallel texts are very good for lexical domains or research questions that have not been at the focus of the linguistic research field, and that are therefore underrepresented in reference grammars. In other words, using parallel corpora, as opposed to reference grammars, can lead to findings that were hitherto unknown simply because they happen to be extant in the corpus. All of these makes parallel corpora ideal for the studying of differences in language use, and therefore for the purposes of my research.

The advantages however only hold true if the quality of the parallel corpus is good enough. Wälchli (2007) rightly comments on the danger of free translations or translations that are plainly wrong, though wrong translations are a problem for typology in general and are therefore not a problem of parallel corpora in themselves. However as it is a problem nonetheless, I try to tackle this in Chapter 2 of this dissertation (see for the outline of the dissertation Section 1.3 below). Wälchli also considers the genre or domain of the corpus a concern, because structures in the corpus may not be as frequent as they are in normal language. Over- or underrepresentation of a structure in a corpus can lead to wrong conclusions, but the domain-sensitivity of linguistic research and NLP tools is a known problem in computational linguistics. Nevertheless, over- or underrepresentation of a structure in a corpus can also tie in with the problem that certain constructions cannot be translated well into another language, which relates with the question of when two sentences are syntactically comparable, which I discuss in Chapter 2 of this dissertation as well.

In relation to wrong translations, though not mentioned by Wälchli (2007), parallel corpora are nowadays often (partially) compiled through the use of automatic translation models. Automatic translation has a reputation of producing wrong translations on a regular basis, with all kinds of errors and translation biases stemming from the structure of the input language, which may be innocent at first glance but may have repercussions for the conclusions drawn by the linguist. Automatic translation therefore poses a problem when working with parallel corpora, and the researcher should always know if the parallel corpus was compiled with machine translation.

Another disadvantage of using parallel corpora as mentioned by Wälchli is that of diversity: available parallel corpora cover much less genealogical and areal diversity than available reference grammars, i.e. while there are many parallel corpora available, there is not one for every language, and certain language families and certain regions are overrepresented. This is certainly true, though over the past few years there have been many endeavours to compile parallel corpora for under-resourced languages (e.g. the JW300 corpus, which

contains over 300 languages and over 54,000 language pairs, including those that are generally under-resourced; Agić and Vulić 2019). Despite the diversity of parallel corpora still not being on par with reference grammars, I believe it should not hinder the linguist to use parallel corpora that are available.

Lastly, Wälchli (2007) discusses that analysis is a sore point of the use of parallel corpora, because of possible differences in script, complicated orthographies, complex morphonological processes and possibly staggering numbers of affixes or function words. He concludes this point with the express wish that some steps of analysis be automated, as it may make the analysis of parallel corpora more appealing in the future, which plays a large role in the research put forth in this dissertation.

1.2 Data

In general, the data at hand, then, is a collection of sentences and their translations: a parallel corpus. The way a computer processes this, is as a sequence of characters, or a string, but in the linguistic reality every sentence is a collection of words that are in a hierarchical relation to one another, have a linear order to each other and have their own morpho-syntactic properties. In order to detect any syntactic differences or to analyse the structural or syntactic context, the data needs to be enriched with syntactic annotations first.[4] It is common practice in computational linguistics to automatically tag every word (and punctuation mark) with a POS tag, as was already mentioned in Section 1.1. With POS tag sets usually consisting of 15 to 25 predefined distinct tags, POS tags identify the category of a word, with every category behaving similarly syntactically. Tagging words with a POS tag allows the linguist to analyse the morpho-syntactic properties of a word and of its context. POS taggers can be stand-alone tagging tools, but are nowadays often integrated parts of an NLP pipeline. Some better-known stand-alone taggers are the Brill Tagger (Brill 1992), the Stanford Tagger (Toutanova et al. 2003)[5] and Frog (for Dutch; van den Bosch et al. 2007).[6]

the	dog	barks	loudly
DET	NOUN	VERB	ADV

Figure 1.2: An example of a POS-tagged sentence.

In order to represent the hierarchical structure of a sentence, linguists often resort to parsing, representing the sentence as a tree. While there are multiple

[4] Technically, it needs to be tokenized first, i.e. segmenting the string into (meaningful) chunks of characters, which correspond to words, particles (such as *'s*) and punctuation.

[5] `https://nlp.stanford.edu/software/tagger.shtml`

[6] `http://languagemachines.github.io/frog/`

ways, theories and conventions of analysing sentences syntactically, they are always useful tools to describe language and to identify language variation. While Figure 1.1 shows an example of a constituency-based parse tree, the use of dependency trees has become more popular in computational linguistics over the past few years. In this dissertation, too, I will represent hierarchical structure in sentences using dependency trees, specifically following the Universal Dependencies programme (UD; Nivre et al. 2016), which aims at cross-linguistically consistent tagging and annotation of dependency trees.[7] The sentence *The dog barks*, for instance, would be represented as in Figure 1.3, containing the POS tag, as well as the syntactic, or dependency, relations between words. These dependency relations make it easy to analyse the syntactic function of a word or a phrase within a sentence cross-linguistically, which is not possible with constituency based trees. There has been much research into automatic parsing, too, and throughout this dissertation I make use of UDPipe, an NLP pipeline specifically developed for Universal Dependencies that tokenizes, tags and parses the data (Straka and Straková 2017).[8]

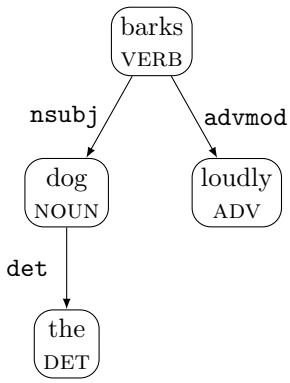


Figure 1.3: A dependency tree of *The dog barks loudly.*

Perhaps the largest and most famous parallel corpus available is the *OPUS* collection (Tiedemann 2012). It contains up to 90 languages, 3800 language pairs and a total of over 40 billion tokens in 2.7 billion aligned sentences or sentence fragments. Two famous subcorpora include the OpenSubtitles corpus (Lison and Tiedemann 2016) and the Europarl corpus (Koehn 2005). The Europarl corpus consists of the proceedings of the European Parliament, a wealth of parallel data as everything is translated into all languages of the EU, and contains around 60 million words per language. In this dissertation I use (parts of) the Europarl corpus for research purposes in Chapters 2 and 3.

Another, perhaps even more famous parallel corpus is the Bible, the text that has been translated into the largest number of languages. For several

[7] `universaldependencies.org`
[8] `ufal.mff.cuni.cz/udpipe`

reasons, the Bible is an attractive corpus for comparative linguistic research, among which is its diversity, availability, heterogeneous nature, fair number of representations of spoken language, and its structure of books, chapters and verses, which make parallelization straightforward (Dahl 2007). However, the Bible is much smaller than the previously mentioned OPUS corpus or the OpenSubtitles and Europarl subcorpora with "only'' 800,000 words in the English version, the King James Bible. In Chapter 4 I make use of the Bible and its parallellity (Christodoulopoulos and Steedman 2015).

The size of the corpus used is certainly something to consider. Computational linguistic research stereotypically requires *big data*, a term that gained traction over the last decade, but the question of how big the data need to be is hard to answer. While it is safe to assume that more data give more opportunities (despite giving rise to other complications with regards to computing power and algorithmic architecture), the Bible is considered to be a relatively small corpus. Nevertheless, I obtained good results in Chapter 4 using the Bible as corpus.

1.3 Outline of the dissertation

Relating this back to the research question of this dissertation, the goal is to detect syntactic differences by automatically comparing vast quantities of parallel sentence pairs that have been syntactically annotated with POS tags, morphological information and parses, and to answer the question of whether and how this is possible.

In Chapter 2 the issue of syntactically incomparable sentence pairs is addressed. In parallel corpora it is not a given that sentences that are aligned to one another are syntactically comparable, exhibiting vastly different constructions or a free translation. To illustrate this, let us consider an example of an aligned sentence fragment triple from the Europarl corpus (Koehn 2005):

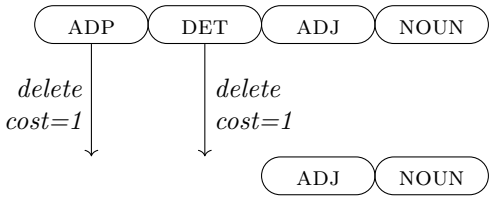
EN: "On the subject at hand, I think that the people of Europe must … "

DE: "Zum Thema: Ich denke, die Bürger Europas müssen…"

NL: "Dan nu het eigenlijke onderwerp: ik geloof dat de burgers van Europa … moeten…"

For the human observer it is immediately obvious that the first part of these fragments (*On the subject at hand* etc.), although being each other's rough translations, are not syntactically equivalent or even comparable. When one would use this instance for the detection of syntactic differences, one would find many that are in fact not informative. "Free'' translations, such as these, must be removed from the dataset. However, using corpora as large as the Europarl corpus makes it impossible to handpick syntactically comparable sentence pairs. A method and measure is needed to filter out sentence pairs that are syntactically too different, while the notion of syntactic comparability is hard

to define. Four ways to automatically filter out parallel sentence pairs that are not sufficiently similar syntactically are explored and evaluated on datasets of English, Dutch and German parallel sentences taken from the Europarl corpus manually labelled for syntactic comparability. The first filter is based on the Levenshtein distance on POS tags (Levenshtein 1966), a well-established algorithm that calculates the minimum number of edit operations that need to be performed in order to turn one sequence into the other. Consider for example Figure 1.4, in which the POS sequence ADP DET ADJ NOUN can be turned into ADJ NOUN most cheaply by deleting ADP (adposition) and DET (determiner), arriving at a Levenshtein distance of 2. In addition to deletion, the Levenshtein distance algorithm also considers the operations insertion and substitution. Adaptations to it can also consider transposition, such as the Damerau-Levenshtein distance (Bard 2007).



Total cost: 2

Figure 1.4: An example of the Levenshtein distance between the sequences ADP DET ADJ NOUN and ADJ NOUN. ADP and DET are deleted, arriving at a Levenshtein distance of 2. This example could appear, for instance, when comparing a language that has prepositions and articles with a language that has case (which is not visible on the tag NOUN) and no articles, such as the pair English-Finnish.

The second filter is based on the sentence-length ratio, built on the presumption that a sentence that is significantly longer or shorter than its translation is likely to be wrongly aligned. The third filter is based on the graph-edit distance (GED) between dependency parses. The GED is equivalent to the Levenshtein distance, albeit on graphs instead of linear sequences. It calculates the minimal number of edit operations that need to be performed in order to turn one graph into the other. The final filter combines the other three in a logistic regression model.

The results of Chapter 2 suggest chiefly that filtering for syntactic comparability is a hard task, in part because syntactic comparability is hard to define. Nevertheless, the filters presented are useful tools for automatizing the selection of syntactically comparable sentences from a parallel corpus. The best results were achieved with the combination filter, while the filter based on the Levenshtein distance or the GED filter can be used to achieve reasonable results. However, the GED filter was suggested to be the most stable throughout

language pairs. The sentence-length based filter did not achieve satisfying results.

In Chapter 3 I present a systematic approach to detect and rank hypotheses about possible syntactic differences for further investigation by leveraging parallel data and using the Minimum Description Length (MDL) principle, building on Wiersma, Nerbonne and Lauttamus (2011). MDL provides an elegant paradigm to find structure in data, formalizing the idea that any regularity in the data can be used to compress the data (among others Grünwald 2007; Barron, Rissanen and Yu 1998). These regularities can then be considered characteristic building blocks underlying the data. The SQS-algorithm ('Summarising event seQuenceS'; Tatti and Vreeken 2012) – an MDL-based algorithm that finds patterns in sequential data – is deployed to mine 'typical' sequences of POS tags for each language under investigation. An important innovation is that these POS sequences are not n-grams, as SQS allows for gaps within the patterns, intuitively making the patterns more flexible and making mapping differences in the use of discontinuous patterns with interfering material easier. When run on English, for example, one would expect SQS to detect patterns such as a preposition followed by an article followed by a noun (e.g. *in the house*), a determiner followed by a noun (e.g. *that (big) tree*, in which the adjective can be gapped over by the algorithm) and a particle followed by a verb (e.g. *to write*), all of which can be considered characteristic building blocks of the English syntax. SQS retrieves lists such as in Table 1.1 ranked by how much they compress the data, indeed containing expected patterns for English, Dutch and Czech. From the Table one can already tentatively conclude that Czech does not use articles as frequently as English or Dutch do, seeing as DET is not as prominently represented in the Czech patterns as in the English and Dutch ones.

	English	**Dutch**	**Czech**
1.	ADP DET NOUN	ADP DET NOUN	ADJ NOUN
2.	DET ADJ NOUN	DET NOUN	ADP NOUN
3.	PART VERB	ADP NOUN	ADP DET NOUN
4.	DET NOUN	ADP DET ADJ NOUN	AUX ADJ
5.	PRON AUX VERB	DET ADJ NOUN	PUNCT SCONJ

Table 1.1: An example of characteristic POS-tag patterns ranked on how much they compress the data found for English, Dutch and Czech.

From these characteristic patterns, a shortlist of potential syntactic differences is created based on the number of parallel sentences with a mismatch in pattern occurrence. The patterns are then ranked on a χ^2 value calculated from these mismatch frequencies, generating hypotheses on where syntactic differences may be found within the language pair. The method is applied to parallel

corpora of English, Dutch and Czech sentences from the Europarl v7 corpus (Koehn 2005), and I experiment with the application of the filter developed in Chapter 2. The approach proved useful in both retrieving POS building blocks of a language as well as pointing to meaningful syntactic differences between languages. The effect of the use of the filter were somewhat minimal, but nevertheless reduced some noise in the results. Despite a clear sensitivity to tagging accuracy, the results and approach are promising.

The method proposed in Chapter 3 assumes the availability of POS taggers for both languages under investigation, and assumes that both languages are annotated using the same tag set and conventions. However, this is not always the case. In fact, although aiming for universality and homogeneous annotation conventions throughout languages, the UD guidelines can differ significantly from language to language (for which there always is a good reason), which was observed in Chapter 3. In Chapter 4 a different approach is explored to detect morpho-syntactic differences that is not dependent on the availability of NLP tools for both languages under investigation. The key question of Chapter 4 is whether it is possible to use fully annotated text in language A (called the source language) to detect grammatical properties of a different, less well-described language B (called the target language), and differences between the two languages, in parallel text. To this end, word alignment is used to map source language words to target language words with the aim of detecting syntactic features of the target language and differences between source and target language by semi-automatically analysing this mapping. Word alignment is the task of automatically identifying translations among words in a parallel text, i.e. identifying which words are each other's translation. Consider Figure 1.5, in which every word in the English sentence is aligned to its translation in the Latin sentence, and vice versa, if there is one. Many alignment algorithms exist, such as the IBM Models (Brown et al. 1993), GIZA++ (Och and Ney 2003), `fast_align` (Dyer, Chahuneau and Smith 2013), and `eflomal` (Östling and Tiedemann 2016), of which I use the last.[9]

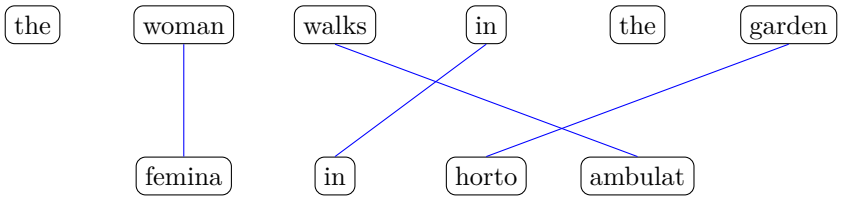


Figure 1.5: An example of word alignment in an English-Latin sentence pair. Word alignments are indicated in blue.

Three tools were developed to detect syntactic properties and differences from parallel data aligned on a word level. The first is the Data Grouper for Attribute Exploration (DGAE), a tool that groups the observed words based

[9] See for the motivation of this choice Section 4.2.1.

on values of attributes (we call an attribute any annotation that was assigned to a word by UDPipe, such as POS tags and dependency relations) and gives useful breakdowns of attribute frequencies within the groups for straightforward exploration. For instance, grouping the data by POS tag quickly shows that articles are very likely not to be aligned to a word in the target language if the target language does not have articles; consider Figure 1.5 once more, in which all English articles are not aligned.

The second tool is the Generalization Tree Inducer (GTI), a tool that structures the data based on the entropy of attributes in an attempt to generalize. GTI aims at iteratively partitioning the data based on the least distinctive feature, with the goal to obtain groups with homogenous attributes. The expected behaviour of this algorithm is that it would detect "stable" attributes that show little variation. For instance, it can be expected that it would partition the data on POS tag very early. With the help of GTI, one can expect to find groups of words with many common attributes, which helps to structure the data, and to detect grammatical properties of the target language.

The third tool is the Affix-Attribute Associator (AAA), which aims to generate hypotheses about which character sequences, or strings, could be affixes in the target language, and to associate them to morpho-syntactic attributes in the source language. Using word alignments, the algorithm looks at the attributes of a word in the source language and tries to associate them to substrings in the word in the target language to which it was aligned. If a certain substring in the target language co-occurs very often with a set of attributes in the source language, a linguist could come to the conclusion that that substring is an affix. For instance, the AAA is expected to find that English verbs with a past-tense attribute are very often aligned to a Dutch word that contains the substring *-te* or *-de*, which is the past tense suffix in Dutch.

The three tools were evaluated on the language pair English-Hungarian. Having no prior knowledge about Hungarian in order not to be biased in my interpretation of the data, I used the tools to generate 43 hypotheses on morpho-syntactic features of Hungarian or differences between it and English. The hypotheses were independently checked by a native speaker and expert of Hungarian and its syntax, and cross-checked with a list of characteristic differences between Hungarian and English independently compiled by said expert. It was concluded that the tools can be used very effectively to form many correct hypotheses on differences between the languages in several syntactic domains. With the help of the tools, I even generated two hypotheses of which the correctness is yet to be investigated, highlighting the power of the tools in the search for syntactic differences between languages.

In Chapter 5 all conclusions and discussions from the previous Chapters are reiterated and related to each other, leading to new observations and conclusions.

Finally, all tools developed and data compiled for this dissertation will be uploaded to `https://github.com/mskroon/DeSDA`, along with relevant output. Additionally, an overview of links to referenced tools or datasets can be

found in the Appendix on page 127.

CHAPTER 2

A filter for syntactically incomparable parallel sentences

A version of this chapter was published as:

Kroon, M., Barbiers, S., Odijk, J., & van der Pas, S. (2019). A filter for syntactically incomparable parallel sentences. *Linguistics in the Netherlands, 36*, 147-161. https://doi.org/10.1075/avt.00029.kro

Author contributions: MK, SB, JO and SvdP conceptualized the research; MK designed and wrote the tools; MK and SvdP labelled the data; MK analyzed the data and wrote the paper; SB, JO and SvdP supervised and critically reviewed the research.

Abstract

Massive automatic comparison of languages in parallel corpora will greatly speed up and enhance comparative syntactic research. Automatically extracting and mining syntactic differences from parallel corpora requires a pre-processing step that filters out sentence pairs that cannot be compared syntactically, for example because they involve "free" translations. In this paper we explore four possible filters: the Damerau-Levenshtein distance between POS-tags, the sentence-length ratio, the graph-edit distance between dependency parses, and a combination of the three in a logistic regression model. Results suggest that the dependency-parse filter is the most stable throughout language pairs, while the combination filter achieves the best results.

2.1 Introduction

An important goal of comparative syntactic research is to identify the syntactic differences between languages and the correlations between these differences. This should lead to an explanation of the locus and limits of syntactic variation (cf. Barbiers 2009). Massive automatic syntactic comparison of languages will greatly speed up and enhance this research. This is necessary given the enormous number of language varieties and syntactic variables involved. Parallel corpora such as Europarl (Koehn 2005), containing sentence aligned versions of the proceedings of the European Parliament in 21 languages, provide excellent data for automatic syntactic comparison.[1] The advantage of using a parallel corpus over a non-parallel corpus for this goal is that in a parallel corpus one can also identify in which contexts the differences occur, whereas in a non-parallel corpus one can only identify quantitative differences (cf. Wiersma, Nerbonne and Lauttamus 2011 for an example of the latter).

We are developing a pipeline to make automatic syntactic comparison of parallel sentences possible, as part of the *DeSDA* project.[2] The first step in this pipeline is to filter out syntactically incomparable parallel sentences. Steps two and three include the extraction of syntactic differences from the remaining sentences and the application of data mining techniques to discover possible correlations. This paper describes and evaluates the first step.

When extracting syntactic differences from parallel corpora, it is essential only to compare sentence pairs that are syntactically sufficiently similar. A method and measure is needed to filter out sentence pairs that are syntactically too different, such as "free" translations. Any extracted differences from too dissimilar sentence pairs will lead to noisy and uninterpretable results. In other works, researchers manually discard incorrect translations or those that are too free (among others, van der Klis, Le Bruyn and De Swart 2017; Abzianidze et al. 2017), whereas we aim for the automatization of the task, which, to the best of our knowledge, has not been attempted before.

In this paper we present four ways to automatically filter out parallel sentence pairs that are not sufficiently similar syntactically,[3] and report on experiments using these filters on manually labelled datasets of English, Dutch and German parallel sentences taken from the Europarl corpus – one filter based on the Levenshtein distance (Levenshtein 1966), one on the sentence-length ratio, one on the graph-edit distance between dependency parses and one that combines the other three in a logistic regression model.

All four filters use a threshold value beyond which a sentence pair is filtered out. In the case of the combination filter, this threshold is automatically derived from the logistic regression model, which is trained in a supervised manner, therefore requiring a gold standard dataset of labelled sentence pairs. The

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

[2] `https://www.universiteitleiden.nl/en/humanities/centre-for-digital-humanities/projects/past-lucdh-research-projects#outline-of-the-desda-project`

[3] The code is made available on `https://github.com/mskroon/DeSDA`

other three filters can either take a manually set threshold, or find an optimal threshold value through supervised learning (see Section 2.4.5).

Syntactic comparability is hard to define and also depends on the research goals. We shall therefore first discuss this concept in more detail. We then describe the data and the filters, and how they were evaluated. Results are presented thereafter. Finally, we discuss and conclude.

2.2 Syntactic Comparability

It is difficult to define syntactic comparability. The sentence pair (1a,b), from the Europarl parallel corpus, involves "free" translation and is clearly not syntactically comparable.

(1) a. That is what will make us strong.

b. Dan zijn wij sterk.
then are we strong
'Then we are strong.' (Koehn 2005)

A similar problem arises with idiomatic expressions, as in (2). Such cases must be filtered out.

(2) a. ... I hope that this report will not be allowed **to bite the dust** on account of this...

b. ... hoffe ich, dass dieser Bericht nicht deswegen **zu Fall gebracht**
hope I that this report not therefore to fall brought
wird...
is
(Koehn 2005)

Switching a sentence's voice may also cause problems, as in the pair (3a,b). The English fragment shows an active construction, the Dutch fragment a passive one. This example should preferably be filtered out for syntactic research, as there is no reason other than (e.g.) a stylistic preference for changing the voice of the verb: (3b′) shows an equally natural yet active translation.

(3) a. This can double the available resources...

b. Hierdoor kunnen de beschikbare middelen worden verdubbeld...
by.this can the available resources be doubled
Through this, the available resources can be doubled. (Koehn 2005)

b′. Dit kan de beschikbare middelen verdubbelen...
this can the available resources double
This can double the available resources...

In this paper, we consider these kinds of examples to be syntactically incomparable. We realize, however, that other researchers may want to use different constraints and definitions in selecting comparable material. The filters can then still be used by manually or automatically setting a threshold that better suits their wishes.

2.3 Data

In order to evaluate the filters, we compiled a dataset of 400 randomly selected English-German-Dutch sentence triples from the Europarl corpus and labelled for each pair whether its sentences are syntactically comparable. This gave us three datasets of 400 sentence pairs, containing the same sentences so as to ensure that the results would be comparable between different language pairs.

These datasets were all labelled by several annotators. The English-Dutch dataset was annotated by three people, in which the inter-annotator agreement (Fleiss' κ (Fleiss and Cohen 1973) was 0.61). The judgement of the majority was taken as truth. The German datasets were annotated by two people each, with a Cohen's κ (Cohen 1960) of 0.55 and 0.26 for the German-Dutch and the German-English datasets, respectively. For the German sets, labelling as done by the first author was taken as truth.

Before annotation, annotators were given the following rule of thumb, which is in line with our definition of syntactic comparability but contains some language-specific examples:

> Two parallel sentences are considered syntactically comparable if: All content words in sentence A have an alignment with a word in sentence B and all content words in sentence B have an alignment with a word in sentence A, ignoring word order, and there is no voice shift, such as active to passive, an idiomatic construction in one language or a (pseudo-)cleft in one language.

The datasets were somewhat imbalanced. The exact distribution of labels ('Y' for syntactically comparable, 'N' for incomparable) can be found in Table 2.1.

2.4 Filters

In this section, we describe the filters in more detail. Each filter calculates a specific value for every sentence pair. The filters determine on the basis of this

	Y	N
German-English	131	269
German-Dutch	106	294
English-Dutch	173	227

Table 2.1: The distributions of the labels in the three datasets.

value whether to keep the sentence pair or to discard it. They are supervised learners, and learn a threshold value, above which sentence pairs are filtered out, from training data. They can also, however, use manually set thresholds or a pre-trained model.

Given that languages differ particularly in the domain of function words and the goal of comparative syntax is to identify syntactic variation, we give users of the filters the option to automatically ignore specific functional material, as based on the words' POS tags.

2.4.1 Levenshtein distance on POS-tags

Using the Levenshtein distance (Levenshtein 1966) on POS tags, which represent morphosyntactic properties of word tokens in context, is a simple approach to filtering for syntactic comparability. The Levenshtein distance is the minimum number of edit operations (in terms of insertion, deletion and substitution) needed to change one sequence into the other, e.g. DET NOUN to ADJ NOUN requires one substitution of DET to ADJ (hence the Levenshtein distance is 1). Intuitively, if the Levenshtein distance between two sentences is low, the sentences are probably syntactically comparable – if it is high, they probably are not.

Users can use the Damerau-Levenshtein distance (Bard 2007) as opposed to the classic Levenshtein distance, adding transpositions to the allowable operations. This will yield lower edit distances between a language where adjectives are prenominal and a language where they are postnominal, for example.

A weakness of the (Damerau-)Levenshtein distance is its sensitivity to whole constituents or phrases moving around. In a comparison of an SVO and an SOV language, the threshold will likely have to be very high to find syntactically comparable sentences, but at the same time having a high threshold will lead to many undesirable sentence pairs not being filtered out. For example, the sentence pair in Figure 2.1 yields a (Damerau-)Levenshtein distance of 4, while if it knew that the object phrase and the verbal cluster were transposed as a whole, the Damerau-Levenshtein distance would only be 2 (1 transposition of the phrases and 1 transposition between AUX VERB–VERB AUX).

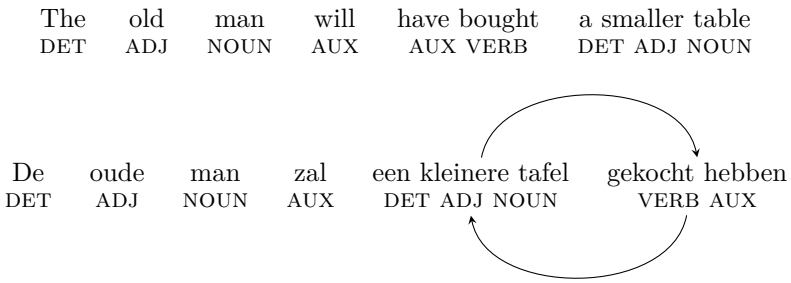


Figure 2.1: The Levenshtein distance is very sensitive to transposing phrases.

2.4.2 Sentence-length ratio

Filtering based on sentence-length ratio is another simple approach: if the sentence-length ratio of a sentence pair is very high or very low, they probably are not syntactically comparable. The sentence-length ratio is defined as the number of words in the source sentence divided by the number of words in the target sentence.

However, some languages use fewer words, for example because they are highly inflectional or do not have articles. Therefore, a language-pair specific threshold is defined in terms of percentiles, where the $n\%$ most extreme sentence-length ratios (relative to the median sentence-length ratio of a language pair) are considered syntactically incomparable – i.e. the left and right tails of the histogram in Figure 2.2 are filtered out. Note that the percentile-based cut-off allows for asymmetric decision rules, where e.g. a sentence is incomparable if it is twice as long in A as in B or three times as long in B as in A.

A sentence-length ratio-based filter is computationally cheap but it does not use any syntactic information, making it very coarse-grained: e.g., the English-Dutch pair *The next item is the vote.–Wij gaan over tot de stemming.* is syntactically incomparable, but the sentences have the same number of words and will not be filtered out, since syntactic information is not taken into account.

Very short sentences are another concern. The pair in (4) is syntactically comparable, but the Dutch and Italian sentences have a sentence-length ratio of 2. A ratio of 2 probably indicates syntactic incomparability when comparing two sentences with 12 and 6 words. This potential issue can be remedied by ignoring function words such as pronouns.

(4) a. ik eet
b. mangio
'I eat.'

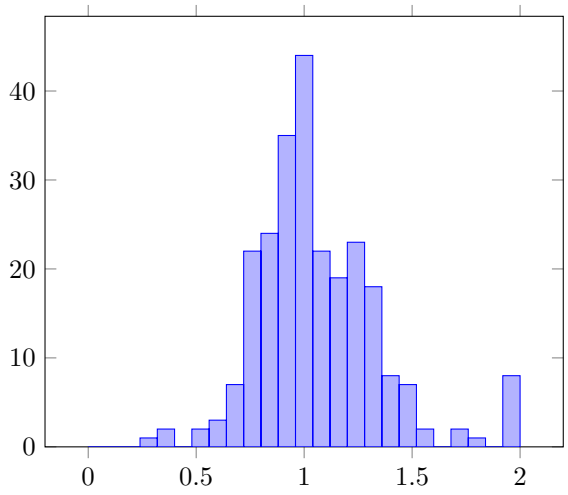


Figure 2.2: A histogram for sentence-length ratios for English and Dutch. The left and right tail are filtered out. If the threshold is, e.g., 10%, the cut-offs would be 0.74 and 1.38: sentence pairs where the English sentence is more than 1.38 times as long or less than 0.74 times as long as the Dutch counterpart will be discarded.

2.4.3 Graph edit distance on dependency trees

Whereas the Levenshtein distance calculates an edit distance between two linear sequences, a graph edit distance (GED) can be applied to hierarchically structured graphs. This has the benefit that it is insensitive to phrases or constituents transposing.

The filter applies Abu-Aisheh et al.'s (2015) exact GED algorithm on dependency parses, where the parses are represented as unordered directed trees (as implemented in `networkx`, Hagberg, Schult and Swart 2008) with labelled edges from heads to dependencies (cf. Figure 2.3). Importantly, both languages should use the same tag set. Nodes, i.e. words, are considered equal if they have the same POS tag; edges, i.e. syntactic relations, are considered equal if they have the same label. Node and edge insertion, deletion and substitution are all defined as 1.

The fact that graphs are unordered should make the algorithm more robust between different languages and language families, as it ignores the linear order between any two words, irrespective of whether these have a grammatical relation between them. The linear order between a word, phrase or constituent and its head is also not represented. Consequently, it is unimportant for the GED whether a direct object is on the left or right of its (head) verb, nor is it important what the linear order of the subject and the object is: transposition costs between sister nodes are therefore 0. This makes it easier to correctly clas-

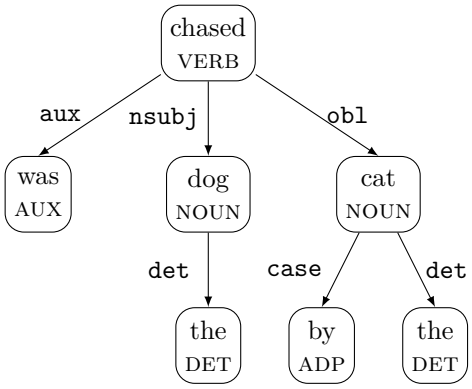


Figure 2.3: An example of a dependency parse (in Universal Dependencies) as an unordered directed tree. Every edge is labelled and directed and the surface order of the words is not represented in the graph.

sify sentence pairs between SOV and SVO (or other) languages as syntactically comparable.[4]

We provide users with the option to ignore function words, similar to the other filters. Importantly, to-be-ignored POS tags are removed from the graph. If a node has children nodes, the edge leading to it is contracted. Any edge leading from the removed POS tag now leads directly from its head to its children (cf. Figure 2.4). The root of the sentence is never removed, sentences always remain one connected component.

Although this approach uses syntactic structure to filter for syntactic comparability and is insensitive to phrase transposition, it is sensitive to parse accuracy. If a dependency parse is erroneous or even slightly off, it will influence the filter as it will yield noisy GED values. Apart from that, it requires the existence of a dependency parser for both languages, which must use the same annotation guidelines and tag set. In our experiments the filter tags and parses sentence pairs using UDPipe (Straka and Straková 2017).

2.4.4 Combination filter

Filtering is essentially a binary classification task. The final filter therefore combines the other filters by fitting a logistic regression model on a pre-labelled dataset of parallel sentence pairs. Each pair is binarily labelled as syntactically (in)comparable. The values calculated by the other filters are then the *features*. Given a pre-labelled dataset all sentence pairs are passed to the other filters,

[4] The linear order of nodes in the tree is only important when discovering syntactic differences, but since this filter is designed only to select sentence pairs from which to extract syntactic differences in a later stage, the linear order can (and should) be ignored by the filter.

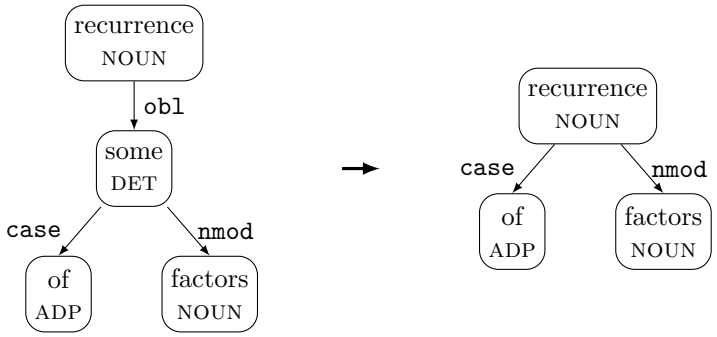


Figure 2.4: An example of contracting edges when DETs are removed. The graphs represent a fragment of *the recurrence of some of these factors*, as found in the Europarl corpus and parsed in UDPipe (Straka and Straková 2017).

which calculate a value. These values are then passed back to a logistic regressor, combined with the labels, to fit a model. This model can be used to calculate the probability of a sentence pair either being or not being syntactically comparable, and to predict a sentence pair's syntactic comparability – if the calculated probability that the sentence pair is syntactically comparable is too low, it will be filtered out. This filter has a clear drawback in that it must have a pre-labelled dataset, which is not always available.

The user can choose which other filters to use for the feature calculation. For every filter, users can select the same options described above. Importantly, ignored functional material need not be the same for each filter.

2.4.5 Automatically setting a threshold

Threshold values can be automatically set with a pre-labelled dataset of sentence pairs using a receiver operating characteristic curve (ROC curve), which plots the true positive rate against the false positive rate at various threshold values – true positive are sentence pairs correctly labelled as syntactically comparable; false positive are those incorrectly not filtered out. The threshold value is found where there is a compromise between the false positive rate and the true positive rate, which can be calculated with Youden's J statistic (Youden 1950).

2.5 Evaluation of the filters

We evaluate the filters on all three datasets, and use them as test sets in order to assess the filters' performance. The sentences used for evaluation were POS tagged and parsed in Universal Dependencies (UD)[5] – a programme that aims

[5] `http://universaldependencies.org/`

at cross-linguistically consistent tagging and annotation of dependency trees (Nivre et al. 2016) – so that all languages used the same tag set. Tagging and parsing was done automatically using UDPipe (Straka and Straková 2017).[6] Aiming for cross-linguistic consistency, UD defines a handful of coarse-grained POS tags that only capture a word's category; morphological information is not included in these tags.[7]

The filters were tested in all possible set-ups and compared to a baseline, which was a bare Levenshtein distance on POS tags: not ignoring any functional material, no transpositions allowed.

When ignoring functional material, we tested all combinations of closed set POS tags of the UD programme,[8] to see which subset of functional POS tags would render the best results when ignored, because the subset of POS tags that are to be ignored is dependent on the language pair in question.

The combination filter was tested by fitting models for all possible combinations of two or three filters in every set-up.

The filters were evaluated in terms of the area under the ROC curve (AUC). For the combination filters, a ROC curve was plotted with the calculated probabilities as threshold values in order to make the results comparable.

2.6 Results

The baselines performed with an AUC of 0.74 on the German-English and English-Dutch sets and with an AUC of 0.76 on the German-Dutch set. Its best thresholds (as found with Youden's J statistic) were 10, 7 and 5, respectively.

Runs with the non-bare Levenshtein distance filters don't clearly outperform the baseline, achieving only slightly higher AUCs. The best runs were also rather divergent in their parameter settings and thresholds.

In general it was observed that the sentence-length filter performed significantly worse, with AUCs of on average about 0.05 lower than the baseline. Ignored functional material differed greatly between the datasets.

The GED filter also did not clearly outperform the baseline in case of German-English and German-Dutch. However, on the English-Dutch dataset it performed somewhat better, with an AUC of 0.77.

More striking, however, is that it is more consistent in its parameter settings throughout the datasets. All best runs included all functional material. Also the threshold was consistent, discarding all sentence pairs with more than 4 edits.

The AUC and parameter settings are summarized in Table 2.3.

[6] `https://github.com/ufal/udpipe`

[7] Although UDPipe also does morphological tagging in the form of attributes, we only used the coarse-grained POS tags in our evaluation. The set of morphological features used by the three languages was too heterogeneous to achieve satisfying results.

[8] UD defines eight POS-tags as being in the closed set: ADP (adpositions), AUX (auxiliaries), CCONJ (coordinating conjunctions), DET (determiners), NUM (numerals), PART (particles), PRON (pronouns) and SCONJ (subordinating conjunctions).

	German-English			German-Dutch			English-Dutch		
	Lev.	**Sent. length**	**GED**	**Lev.**	**Sent. length**	**GED**	**Lev.**	**Sent. length**	**GED**
AUC	0.75	0.66	0.75	0.77	0.68	0.75	0.74	0.73	0.77
Threshold	9	24%	4	5	24.375%	4	7	20.625%	4
Ignored functional material	AUX, CCONJ, NUM	ADP, NUM	–	ADP, AUX, CCONJ, NUM, PART	SCONJ	–	AUX, CCONJ, NUM	AUX, NUM	–
Transpositions	No	–	–	No	–	–	Yes	–	–

Table 2.3: Overview of the results of the filters: AUC, and parameters per language pair.

The combination filter more clearly outperformed the baseline, with AUCs of on average about 0.06 higher than the baseline. It benefited from using all other filters, all best runs using all three single filters, though interestingly with different parameters.

In the best German-English run, the Levenshtein filter did not use transpositions, and ignored CCONJ, NUM, and PART instead of AUX. The sentence-length filter ignored ADP, but no NUM. The GED filter ignored NUM and SCONJ. In this setup, it achieved an AUC of 0.79.

As for the German-Dutch dataset, the best run, with an AUC of 0.80, was achieved by combining a Levenshtein filter that ignores ADP, CCONJ, NUM, PART and SCONJ and allows transpositions, a sentence-length filter that ignores all functional material but NUM, and a GED filter that ignores CCONJ.

Finally, the best run for the English-Dutch dataset achieved an AUC of 0.81, combining a transposing, CCONJ ignoring Levenshtein filter, a sentence-length filter that ignores nothing and a GED filter that ignores ADP, DET, NUM and SCONJ.

2.7 Discussion

The results suggest chiefly that filtering for syntactic comparability is a hard task, as corroborated by the annotations' relatively low κ values. Nevertheless, we believe that the presented filters are useful tools for automatizing the selection of syntactically comparable sentences from a parallel corpus, especially since it allows users to manually set thresholds and parameters and to work

with other definitions of syntactic comparability.

The results further suggest that German, English and Dutch are rather similar syntactically. If not, we would have expected a larger performance gap between the GED and Levenshtein filters, due to the Levenshtein distance's sensitivity to constituents or phrases transposing. On the other hand, the difference in parameters between the Levenshtein runs does point towards syntactic dissimilarity of the languages, since, if the languages were more similar, the sets of ignored function words would have been smaller.

The filters' sensitivity weakly suggest that there is a syntactic difference to be found in the use of auxiliaries, adpositions and conjunctions. The fact that numerals are often ignored can be explained by the difference in how numerals are tagged by UDPipe: in English and German ordinal numerals are often tagged as adjectives (e.g. *second*) or adverbs (e.g. *thirdly*) – which are both rather frequent in the Europarl corpus – whereas in Dutch they are always tagged as numerals. This emphasizes the importance of uniform tagging conventions between compared languages. Although UD aims for consistent tagging, there are subtle differences from language to language. These differences, however subtle, lead to issues for our filters.

Overall, the best filter is the combination filter. It necessitates, though, the existence of a pre-labelled dataset – and if such a dataset is available, doing a grid search to find which parameters yield the best results is computationally expensive. Also, the risk of overfitting on the dataset is high.

If a pre-labelled dataset is not available, the other filters can still be used with reasonable results by setting thresholds manually. While the baseline is not clearly outperformed by the other filters, the GED filter's robustness in its parameters, thresholds and performance throughout the different language pairs suggests that it is most stable in all aspects. Its parameter robustness even suggests that the settings found could be used as a default for other language pairs. We also expect it to outperform other approaches more clearly when supplied with more accurate parses.

The Levenshtein filter performs similarly, but has the advantage of not requiring a parser model. If such a model is not available, the Levenshtein filter could still be used, but it is likely to perform well on closely related languages only and requires more parameter fine-tuning.

The sentence-length filter did not give satisfying results, as expected since it does not use any syntactic information. Interestingly, the combination filter did use sentence-length ratio. This makes sense, as using the sentence-length ratio to filter out the most extreme sentence pairs allows for the model to more finely tune the weights for the Levenshtein distance and GED, yielding more informed decisions.

A point of improvement could be the edge contraction conventions when nodes are removed in the GED; in our current design, the `obl` relation in Figure 2.4 is lost entirely, which may be undesirable.

It will be most useful to improve the filter such that it also selects sentence *fragments* that are syntactically comparable. Now an entire sentence can

be filtered out despite being almost completely syntactically comparable. A possible design of such a filter is top-down: if a sentence-pair exceeds the edit threshold, the algorithm can search for two pairs of maximally large subtrees that do not exceed the threshold.

The way the combination filter operates could perhaps be improved upon, too. Now the other filters are combined in parallel. A sequential fashion may yield better results, discarding sentence pairs that exceed some sentence-length ratio before optimizing a threshold for the GED filter, for example.

2.8 Conclusion

Automatic extraction of cross-linguistic syntactic differences from parallel corpora will greatly speed up comparative syntactic research. Automatic extraction requires a pre-processing step to filter out syntactically incomparable sentence pairs, e.g., because they involve "free" translations. In this paper we evaluated four possible filters. The best results were achieved with a filter that combines the other three in a regression model, but it has the downside of requiring a pre-labelled training set, more so than the others which allow for manual tuning. Alternatively our filter based on the Levenshtein distance or our GED filter can be used to achieve reasonable results, but both have their own weaknesses. Our last filter, based on sentence length, did not achieve satisfying results in itself, as expected.

CHAPTER 3

Detecting syntactic differences automatically using the minimum description length principle

A version of this chapter was published as:

Kroon, M., Barbiers, S., Odijk, J., & van der Pas, S. (2020). Detecting syntactic differences automatically using the Minimum Description Length principle. *Computational Linguistics in the Netherlands Journal, 10*, 109-127.

For typographical reasons, the large Tables 3.3 through 3.5 have been adapted and split over two pages.

Author contributions: MK, SB, JO and SvdP conceptualized the research; MK designed and wrote the tools; MK, SB and JO analyzed the data; MK wrote the paper; SB, JO and SvdP supervised and critically reviewed the research.

Abstract

In this paper we present a systematic approach to detect and rank hypotheses about possible syntactic differences for further investigation by leveraging parallel data and using the Minimum Description Length (MDL) principle. We deploy the SQS-algorithm ('Summarising event seQuenceS'; Tatti and Vreeken 2012) – an MDL-based algorithm – to mine 'typical' sequences of Part of Speech (POS) tags for each language under investigation. We create a shortlist of potential syntactic differences based on the number of parallel sentences with a mismatch in pattern occurrence. We applied our method to parallel corpora of English, Dutch and Czech sentences from the Europarl v7 corpus (Koehn 2005).

The approach proved useful in both retrieving POS building blocks of a language as well as pointing to meaningful syntactic differences between languages. Despite a clear sensitivity to tagging accuracy, our results and approach are promising.

3.1 Introduction

The central question of theoretical comparative syntactic research is: What is an (im)possible natural language? As an answer to this question, a formal theoretical model needs to be developed that captures all syntactic structures that are possible in natural language and excludes all impossible structures.

This research program requires massive and detailed comparison of syntactic structures in a large number of languages, in order to discover the (abstract) syntactic principles that all languages have in common and that determine the range and limits of variation. This systematic comparison is a daunting task in view of the large number of distinct syntactic structures, the high degree of variation and the large number of language varieties in the world and therefore proceeds too slowly if carried out by humans alone. Also, the human observer may be biased by expectations of what will be found.

We therefore need the help of the computer to scale up and enhance the systematic cross-linguistic comparison of syntactic structures. In this paper we propose a method for automatic detection of syntactic differences in huge parallel corpora.[1] We present a systematic approach to detect and rank hypotheses about possible syntactic differences for further investigation by leveraging parallel data and comparing frequencies of Part of Speech (POS) tag sequences. To delineate our contribution, a diagram may be helpful; the process of discovery of syntactic variation is conceptualized as a three-step-process in Figure 3.1. Our contribution is towards the second step, guiding the linguist to interesting hypotheses in a data-driven way. We will come back to the other two steps in the discussion.

[1] The code is made available on `https://github.com/mskroon/DeSDA`

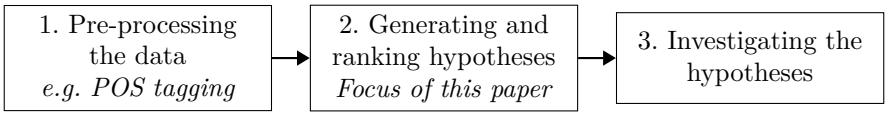


Figure 3.1: Schematic overview of the process of discovery of syntactic variation.

Ideally, to capture the enormous variety in syntactic differences, the algorithm should be without bias, and would not be limited in the kind of patterns to consider. However, without any limitations the number of patterns to search over rapidly exceeds current computing capacity. In this paper, we make use of the Minimum Description Length (MDL) principle (see e.g. Barron, Rissanen and Yu 1998; Grünwald 2007) in order to circumvent this problem. MDL translates the problem of pattern finding to a compressibility problem, prioritizing patterns for which an encoding leads to the shortest possible description of the corpus, and has been used in syntactic research before (among others: Osborne 1999a; Osborne 1999b; Wong et al. 2017).[2] Compressing with MDL yields a shortlist of patterns that can be considered 'building blocks' of the corpus. More specifically, we deploy the SQS-algorithm ('Summarising event seQuenceS' Tatti and Vreeken 2012) – an MDL-based algorithm – to mine 'typical' sequences of POS tags that vary in length as well as allow for gaps, pushing the boundaries of allowed flexibility in the patterns considered by an algorithm.

We apply our method to parallel corpora of English, Dutch and Czech sentences from the Europarl v7 corpus (Koehn 2005). The comparison of English and Dutch will serve as a sanity check of sorts, since many syntactic differences between the two have been described exhaustively in the past (see e.g. Donaldson 2008; Aarts and Wekker 1987). While domain-specific differences between Czech and English have been described (see e.g. Dušková 1991; Babická et al. 2008; Malá 2014) and Czech grammars have been written from the perspective of an English speaker (see e.g. Naughton 2005), to the best of our knowledge, a dedicated work systematically describing syntactic differences or a contrastive grammar of Czech with Dutch or English does not exist. The comparison of Czech to English and Dutch will therefore showcase the potential of our proposed method and deliver a basic fragment of a contrastive grammar.

First we shall discuss some previous work on the automatic detection of syntactic differences. After that, in Section 3.3, we shall describe our proposed method (i.e. step 2 in Figure 3.1) in more detail. In Section 3.4 we describe our experiments with English, Dutch and Czech and discuss their results for each step. We end with a general discussion in Section 3.5 and conclude in Section

[2] Using MDL in learning linguistic patterns from a corpus, may raise questions on the cognitive aspects of MDL and on the role of MDL in human language acquisition. This, however, is not in the scope of this research.

3.6.

3.2 Background

An early contribution to automatic detection of syntactic variation was made by Nerbonne and Wiersma (2006) and Wiersma, Nerbonne and Lauttamus (2011), who devised a method based on POS n-grams to select on statistical grounds hypotheses about related dialects and language varieties for further investigation. Their method consists of taking POS n-grams ($1 \leq n \leq 5$) from two comparable, non-parallel corpora from the same language. After that, they compare the relative frequencies of the POS n-grams using a permutation test[3] and sort the significant ones by degree of difference. In their paper, they demonstrated the utility of their approach by detecting syntactic differences between the English of two generations of Finnish immigrants to Australia (Nerbonne and Wiersma 2006). In this experiment they opted for using trigrams with a frequency of 5 or higher only for statistical reasons. This method was extended by Sanders (2007), who used the leaf-ancestor path representation[4] of parse trees developed by Sampson (2000) instead of n-grams, and applied this method to find dialectical variation between several British regions.

We further extend this approach in two directions. The main innovation is that we search over all possible n-grams for any value of n, with no need to commit to a fixed n. We also include the possibility for the POS n-grams to contain gaps. Allowing for n-grams with gaps intuitively makes the patterns more flexible, and makes mapping differences in the use of discontinuous patterns with interfering material easier. For example, gapping over the adjective in an article-adjective-noun sequence allows us to identify the sequence as being an occurrence of article-noun, too, in turn allowing us to identify a syntactic difference in the use of articles more easily. As mentioned, we use SQS (Tatti and Vreeken 2012), which applies the Minimum Description Length (MDL) principle to mine for characteristic POS-tag patterns. Applying the MDL principle in this task furthermore circumvents complex normalization or ranking techniques to select relevant patterns; while using all n-grams brings the risk of having many irrelevant patterns, SQS automatically selects POS-tag patterns typical of the data due to the principle on which the algorithm was built. This will be explained in more detail in Section 3.3.1.

The second extension is that we compare different languages. The major underlying goal of this extension is to contribute to the question which syntactic

[3] A permutation test is a type of statistical test in which the data from both languages are pooled and repeatedly reshuffled into two new data sets. Some measure, such as the difference in frequency of a particular n-gram, is then computed on these reshuffled data sets and then compared to the measure based on the original data set. See Wiersma, Nerbonne and Lauttamus (2011) for more details.

[4] Sanders' (2007) leaf-ancestor path representation records each word (i.e. leaf in a tree) as a path from the root of the tree to the leaf. For example *S-NP-Det-The*, *S-NP-N-dog* and *S-VP-V-barks* from the sentence *The dog barks*.

properties are universal, which are language specific, and how these properties interact. A search for cross-linguistic differences removes the need for some of the statistical tests employed by Wiersma, Nerbonne and Lauttamus (2011) and Sanders (2007). For example, Wiersma, Nerbonne and Lauttamus (2011) first formally test whether there are syntactic differences at all between the English of the two generations of immigrants, while in cross-linguistic comparison as in the present paper, the existence of syntactic differences is presumed and requires no formal test. To ensure comparability and improve interpretability of results across languages, we furthermore use a parallel corpus in our research. The method can be adapted for use with non-parallel corpora, too, a possibility we will come back to in the discussion.

3.3 Generating hypotheses with the minimum description length principle

We propose a two step process. In the first step, typical patterns per language are mined using SQS, taking POS-tags as the input. In the second step, a search and filtering method based on distributional differences is used, resulting in a ranked shortlist of potential sources of syntactic variation. This means that step two, as pointed out in Figure 3.1, will in itself encompass two sub-steps – 2a and 2b – as in Figure 3.2. In this process, steps 2a and 2b both yield useful results, and for some purposes step 2a alone may suffice.

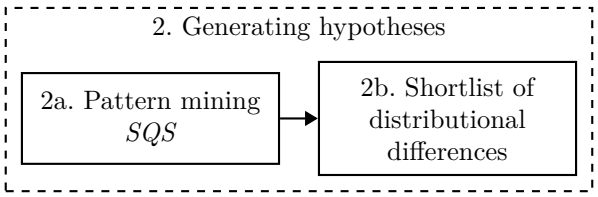


Figure 3.2: Schematic overview of hypothesis generating mechanism.

3.3.1 Step 2a: Pattern mining with SQS

Ideally, as few limits as possible are set on the combinations of POS-tags that are considered as potential patterns. The cost of allowing increasingly flexible patterns is an increase in the number of patterns to search over, making the ranking process more complicated and computationally expensive. A balance between flexibility and feasibility needs to be struck, and the minimum description length principle-based SQS algorithm offers an appealing compromise.

The minimum description length principle provides an elegant paradigm to find structure in data, formalizing the idea that any regularity in the data can

be used to compress the data (among others Grünwald 2007; Barron, Rissanen and Yu 1998). These regularities can then be considered characteristic building blocks underlying the data. For example, if our data consists of POS-tagged sentences,[5] as follows:

PRON AUX DET ADJ NOUN
DET NOUN VERB ADP DET NOUN
PRON VERB PRON ADP DET NOUN
DET NOUN AUX ADV VERB PRON
DET ADJ NOUN VERB DET NOUN
DET NOUN ADP DET NOUN AUX VERB PRON
DET NOUN AUX VERB PRON ADP DET NOUN
DET NOUN VERB PRON

we could compress[6] these into

Codebook	**Coded data**
ADP DET NOUN ↦ A	E F D
DET NOUN ↦ B	B G A
VERB PRON ↦ C	E C A
DET ADJ NOUN ↦ D	B F H C
PRON ↦ E	D G B
AUX ↦ F	B A F C
VERB ↦ G	B F C A
ADV ↦ H	B C

using the 'codebook' on the left. If a pattern leads to a substantial reduction in the number of tokens required to describe the data set, DET NOUN, VERB PRON and ADP DET NOUN in this example, we may consider it a typical pattern.

The main question is which codebook to use. In the minimum description length paradigm, the optimal encoding C_{opt} is codebook C that achieves the ideal balance between $L(C)$, the length of the codebook itself, and $L(D|C)$, the length of the data D as compressed using the codebook, expressed mathematically as:

[5] Using the Universal Dependencies tagset (Nivre et al. 2016).

[6] It must be stressed that this example is a toy example, in which the difference in size between the original data and the compressed data is very small. When performed on larger data, the compression rate will be much more substantial.

$$C_{opt} = \arg\min_{C} \left(L(C) + L(D|C) \right).$$

This is generally a difficult optimization problem, since the number of possible codebooks is 2^n, where n is the number of possible codes or patterns to consider putting in the codebook (which is a very large number in itself, especially when considering gaps). Given that this number of codebooks grows exponentially with the number of codes, an approach that approximates the optimal solution is necessary. The difficulty of finding the optimal encoding also depends on the type of codes that are allowed. More flexibility in these codes leads to a harder problem, e.g. finding the optimal codebook when only 3-grams (i.e. codes of length 3) are allowed is substantially easier than finding the optimal codebook when all possible n-grams are considered.

The SQS-algorithm ('Summarising event seQuenceS'; Tatti and Vreeken 2012) is based on the minimum description length principle and finds patterns in sequential data. In their paper Tatti and Vreeken show that SQS is able to mine typical phrases in several texts successfully. In our proposed approach, SQS is deployed to detect patterns in POS-tags. The main innovation of SQS is that it allows the possibility to leave gaps in the pattern. In our POS-tagged example, this means that in addition to all possible n-grams, SQS will also consider e.g. DET NOUN as a possible pattern in the data DET ADJ NOUN, gapping over the ADJ. To limit the number of patterns under consideration, however, SQS limits the number of gaps that can occur in a pattern to be strictly less than the length of the pattern itself; in the case of DET NOUN, SQS can gap over one element, while in the case of DET ADJ NOUN, it can gap over at most two elements.[7]

The main appeal of this approach is the enormous flexibility. With SQS, we can find patterns of variable length, without the need to commit to a specific value of n for n-grams; the codebook returned by SQS can contain uni-, bi- and e.g. 7-grams alike, and the composition of the codebook is chosen such that the data can be compressed (more or less) optimally with it. Moreover, the possibility of having a gap allows us to identify patterns that can take optional material that would interfere in an approach where no gaps are considered.

The main disadvantage is that the possibility of a gap can make interpretation difficult. Consider for example that the pattern DET NOUN ends up in the codetable. It is then unknown whether this pattern was ever attested with other material between the two words, i.e. with a gap. Although in the case of DET NOUN it may still be relatively easy to interpret, interpretation becomes increasingly difficult the longer the patterns become due to the possible gap configurations. As a result of this, longer patterns can still be a characteristic

[7] Where these gaps occur inside the pattern, does not matter, as long as the number of gaps does not exceed the length of the pattern. DET ADJ NOUN therefore matches DET ADJ GAP NOUN, DET GAP ADJ NOUN, DET GAP ADJ GAP NOUN, DET GAP GAP ADJ NOUN and DET ADJ GAP GAP NOUN, in which GAP can be any POS tag.

POS-tag pattern of a language but it may be unclear what they mean syntactically and whether they do not just happen to compress the data well without bearing any linguistic relevance. Examples of this interpretation difficulty will be discussed in Section 3.4.

3.3.2 Step 2b: Creating a shortlist of distributional differences.

Based on the assumption that the distribution of a pattern must be the same in both languages if there is no syntactic difference, we extract potential syntactic differences from the pattern lists obtained through SQS. We leverage the parallelism of our corpus by considering whether a pattern is present in both translations of a sentence.

In more detail, we take two lists of patterns as obtained through SQS. Because SQS does not explicitly return unigram patterns,[8] we add all unigrams to the pattern lists. For each pattern we then count in the textual data how often it occurs in language A while not occurring in its translation in language B and how often it occurs in language B while not occurring in its translation in language A; mismatching frequencies, so to say. From these frequencies we calculate a χ^2-value as

$$\chi^2 = \frac{(b-c)^2}{b+c}$$

where b and c are the mismatching frequencies. The motivation behind this is that this is the test statistic of the McNemar test (McNemar 1947), which was designed to be used with paired nominal data. Seeing as we want to create a ranked list of potential syntactic differences, to be investigated by a linguist, statistical significance is not of much importance, and we therefore do not propose a certain cut-off point, threshold value or α-level. In our case the χ^2-value is a practical, one-dimensional summary of the extent of difference in distribution of a pattern between two languages on which we sort the patterns: the higher the χ^2-value, the more strongly a distributional difference and therefore syntactic difference is suggested. Apart from sorting on χ^2-value, we also report on mismatching frequencies in order to make interpretation easier.

We must, however, consider the case of 'subpatterns', that are contained by other patterns.[9] For if we, e.g., find a distributional difference for the pattern DET ADJ NOUN, we will also find a difference for pattern ADJ NOUN, because all occurrences of DET ADJ NOUN also count towards occurrences of ADJ NOUN. Since this is not informative per se, we also experimented with subtracting the occurrences of DET ADJ NOUN, i.e. their superpattern, when

[8] This is because implicitly a codebook minimally must contain all unigrams, otherwise the data cannot be fully encoded. From an algorithmical point of view, SQS does not add unigrams to its output because unigrams do not compress the data.

[9] To avoid confusion: we say XY is contained by XYZ: all singletons in XY are in XYZ and the gap configuration allows for an alignment. As such, YZ and even XZ are also contained by XYZ.

counting occurrences of ADJ NOUN; if we then find a difference again, there is a difference with ADJ NOUN proper. We therefore sort the patterns on length and start with the longest pattern, because subpatterns must by definition be shorter than a pattern containing them.

To summarize, we mine for potential syntactic differences by running SQS on two parallel POS-tagged corpora (using the same tagset), taking all patterns and counting their mismatching occurrences, from which we calculate a χ^2-value. Having sorted on this, this yields a ranked list of POS-tag patterns sorted by extent of distributional difference. The bigger the difference, the more strongly a syntactic difference between the languages pertaining to that pattern is suggested. Similar to Wiersma, Nerbonne and Lauttamus (2011), a linguist should then investigate these patterns.

It is important to note, however, that other linguists may opt to divert from our approach after step 2a, for example when the patterns from SQS prove interesting enough or if they desire to shortlist differences differently, employing a different ranking technique, to better suit their needs. If a user of our method does want to use a cut-off point, threshold value or α-level, we strongly recommend correcting for multiple testing, for example using a Bonferroni correction (Bonferroni 1936) or the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995).

3.4 Example: Europarl

To illustrate the effectiveness of our proposed technique, we report on three runs on the Europarl parallel corpus (Koehn 2005): English-Dutch, English-Czech and Dutch-Czech. Since the language pair English-Dutch has been described extensively in literature (among others Donaldson 2008; Aarts and Wekker 1987), the first run will function as a sanity check as well as a proof of concept. The runs involving Czech show the method's effectiveness on less well described language pairs. Specifically the data used consisted of 10000 sentences of the corpus that were available in all three languages so as to ensure comparable results between the three runs. This resulted in 219781 tokens for English, 224622 tokens for Dutch and 193482 tokens for Czech.

There are various complications, however, with using the Europarl corpus. One of which is that a substantial amount of the data consists of headlinese: section titles, such as *Agreement between the EC and Australia on certain aspects of air services*, section numbering, and notes (such as *Closure of sitting* and *Written statements (Rule 116): see Minutes*). This could potentially be a problem, as it is unknown how much of the data really is headlinese. If the proportion of headlinese sentences is high, it could influence results, since it has been shown that headlinese grammar significantly differs from standard grammar (among others Mårdh 1980; de Lange 2004; Weir 2009). For example, article drop is very common in English and Dutch headlines, and if the proportion of headlines where this occurs is very large, our method may be unable

to detect a syntactic difference with Czech which lacks articles altogether. The same holds for formulaic utterances used in Parliament, such as *I put to the vote the proposal*, which have high frequency and can influence results. A remedy to this would be to remove headlines and formulaic utterances, but this poses a entirely different problem which lies beyond the scope of this research. We therefore decided to leave the data as it is, also because it would only underline the usefulness of the proposed method if it still found meaningful differences in real data.

3.4.1 Step 1: data pre-processing

For preprocessing, step 1 in Figure 3.1, we are using POS tags from the Universal Dependencies (UD) framework for consistent annotation of grammatical properties (parts of speech, morphological features and syntactic dependencies) across different human languages (Nivre et al. 2016). For this we used UDPipe (Straka and Straková 2017), a pipeline for tagging and parsing in UD, using the latest models pertaining to UD 2.3.[10] UD uses 17 different POS tags, which were all used in the tagging of our data.

We noticed however that there was an (easily solvable) inconsistency in tagging between English and Dutch. While English verbal particle *to* was consistently tagged as a particle (PART), its Dutch counterpart *te* was consistently tagged as a preposition (ADP). This was remedied by manually changing all occurrences of *te* to a PART when it was directly followed by a verb or auxiliary,[11] because such an inconsistency results in syntactic differences found that are actually spurious. Similar preprocessing was also done for Czech.

Furthermore, we investigated the effect of using Kroon et al.'s filter for syntactic incomparability (see also Chapter 2; i.e. Kroon et al. 2019) on the results, since in principle step 2b requires sentences to be syntactically comparable.[12] The filter was designed to remove noise from the data (such as too free translations) by selecting sentence pairs that are syntactically comparable and suitable for syntactic research, and by removing those that are syntactically incomparable based on a threshold setting. We therefore experimented with and without filtering the data before counting mismatching occurrences

[10] Specifically, the English EWT model, the Dutch Alpino model and the Czech PDT model, all from November 15, 2018. Available at `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2898`.

On pre-tokenized data, the POS-tag accuracy of the models are reported as respectively 94.4%, 94.4% and 98.3%.

[11] In other positions the ADP tag was kept, because *te* can also function as a preposition ('in') or even as a degree morpheme ('too').

[12] The term *syntactic comparability* is hard to define, and filtering out sentence pairs that are too different syntactically in order to detect syntactic differences seems circular. However, in order to find differences between the syntactic potentials of two languages rather than their syntactic preference, noisy sentence pairs, that show incomparable structures for no other reason than a preference, must be removed from the data. For a more detailed discussion on this, we refer to Kroon et al. (2019).

of patterns. Specifically, we used the graph edit distance[13] based filter with threshold 4, which was proposed by Kroon et al. (2019) to be a default setting if a training set was lacking, meaning that if the graph edit distance between the dependency graphs on the two sentences as parsed with UDPipe exceeded 4, the sentence pair would be removed. In this we opted to ignore function words, a class we defined based on the closed set POS tags in UD, because syntactic variation often occurs in the domain of function words. After filtering out incomparable sentence pairs, about one fifth or one sixth of the sentences remained in the data (English-Dutch: 2197 (15628 and 15478 tokens); English-Czech: 2096 (16677 and 14324 tokens); Dutch-Czech: 1665 (10481 and 9228 tokens)).

3.4.2 Step 2a: characteristic patterns per language

Running SQS on the data yielded 302 POS-tag patterns in the data for English, 199 for Dutch and 89 for Czech. The top-10 most characteristic, i.e. compressing the data most, patterns for the three languages are presented in Table 3.2. Notice that many patterns are somehow permutations or subpatterns of each other. Also notice that English and Dutch exhibit more similar pattern lists than Czech; the fact that Dutch and English are more closely related to each other than to Czech is therefore nicely corroborated by these lists.

These codetables with POS-tag patterns are already insightful for many linguistic purposes, as they reflect the syntactic building blocks of a language, despite not directly reflecting the hierarchical structure that characterizes human language. For example, this top 10 already suggests strongly that English has mostly prepositions (as suggested by pattern 1, an adposition followed by a determiner and a noun),[14] possibly few grammatical cases because of the abundance of patterns with adpositions, a verbal particle that occurs often, and a V-NP word order by virtue of pattern 6 (sentences or phrases ending in a noun).

As an important side note: we investigated the stability of SQS's output patterns between different datasets by running it on 10000 different sentences

[13] The graph edit distance, or GED, is the minimal number of edit operations needed to transform graph A into graph B. One can compare it to the Levenshtein distance (Levenshtein 1966), only for hierarchical trees or graphs instead of linear sequences. It has the advantage of not being sensitive to the directionality of two sister nodes, or even between a node and its mother or head, making it more reliable in its filtering between less closely related languages.

[14] While it is the case that prepositions are both most likely preceded and followed by a noun (taking into account the possible gap, just like SQS does), the entropy for following material is much lower, meaning that the certainty of what follows is higher. That is to say, it is more unlikely that something other than a noun follows a preposition, than it is unlikely that something other than a noun precedes it. It is therefore better for SQS to add the pattern ADP NOUN to the codebook than to add NOUN ADP (in which the ADP stands for a preposition), because it more efficiently compresses the data. For Japanese, a strict head-final language with postpositions, the entropy is lower for preceding material, resulting in the adding of NOUN ADP to the codebook, instead of ADP NOUN. Therefore, the presence of ADP NOUN in the codebook suggests that a language has prepositions.

	English	Dutch	Czech
1.	ADP DET NOUN	ADP DET NOUN	ADJ NOUN
2.	DET ADJ NOUN	DET NOUN	ADP NOUN
3.	PART VERB	ADP NOUN	ADP DET NOUN
4.	DET NOUN	ADP DET ADJ NOUN	AUX ADJ
5.	PRON AUX VERB	DET ADJ NOUN	PUNCT SCONJ
6.	NOUN PUNCT	AUX VERB PUNCT	ADJ NOUN PUNCT
7.	PRON VERB	ADP ADJ NOUN	NOUN PUNCT DET VERB
8.	ADP DET ADJ NOUN	SCONJ PRON	ADP DET PUNCT SCONJ
9.	ADP NOUN	SCONJ DET NOUN	AUX ADV ADJ
10.	ADP ADJ NOUN PUNCT	ADP PRON NOUN	PRON ADV VERB

Table 3.2: The top-10 most characteristic POS-tag patterns found in the data for English, Dutch and Czech.

from the Europarl corpus for English and Dutch. We noticed that the output was very comparable between the different parts of the corpus, although the order of the patterns differs slightly. This suggests that the patterns found really reflect true properties of the languages and are not a result of strong overfitting on the input data. We did not check for stability across genres, however.

3.4.3 Step 2b: distributional differences

Based on syntactic literature (e.g. Radford 2004; Zwart 2011) and the authors' knowledge of English and Dutch, we should expect the algorithm to especially find differences in the verbal domain. Whereas English is strictly SVO, Dutch has V1 or V2 if the verb is finite and no complementizer is present and SOV otherwise. This should for example lead to our method finding that patterns with a verb sequence (i.e. one or more verbs or auxiliaries) followed by a noun phrase are more frequent in English than in Dutch, because in English the object must follow the verb(s) while in Dutch it is only preceded by the finite verb if there is no complementizer.

As mentioned, we investigated the effect of subtracting occurrences of superpatterns on the results, as well as the effect of using a filter for syntactic incomparability (Kroon et al. 2019) before counting mismatching occurrences. This led to 4 distinct runs for each language pair, yielding varying numbers

of differences per run, per language pair. The top 10 highest ranking differences are reported in Tables 3.3 to 3.5, along with the total frequencies of each pattern per language, the mismatching frequencies, written as $x : y$, and the χ^2-value, by which the list is ranked.

What can be noticed from the results in Tables 3.3 to 3.5 is that the average lengths of the differences found is shorter when superpattern occurrences are not subtracted. This is due to the fact that the algorithm starts out with the longest patterns, the occurrences of which will then not count towards the calculation of the χ^2-value for shorter patterns. This leads for example to the fact that DET NOUN is not found to be a top-10 difference when subtracting superpatterns between Czech and English at all, because DET NOUN was included in many other patterns.[15] At first sight this may seem problematic, however the superpattern subtraction method returns more detailed differences by including specific contexts in which the syntactic difference occurs, while the runs without superpattern subtraction return more general patterns. We therefore give users of this algorithm the option to subtract superpatterns or not, because both approaches have their strengths, as will be further exemplified in Section 3.4.4.

Relating the results to the expectation of finding differences between Dutch and English in the verbal domain, we see several patterns with verbs and auxiliaries across the four experimental setups. Although we do not find a pattern with a verb sequence followed by a noun phrase, we do find the opposite, which is, in line with our expectation, more often unmatched in Dutch (i.e. there are more occurrences of DET NOUN VERB in Dutch that do not have an occurrence of said pattern in the English translation). Additionally, in general, we see many patterns in which an auxiliary is followed by a verb in English to be more often unmatched in Dutch; this is also in line with our expectations, since in Dutch the auxiliary and the verb are often split by other material due to the V2 word order.

It is important to note that the differences found by this step are not by definition a syntactic difference. The patterns for which it finds a large distributional difference (i.e. a large χ^2-value) are therefore returned as possible syntactic differences, giving rise to hypotheses which then have to be investigated and tested by linguists. While the results of steps 2a and 2b are already insightful, our proposed method is in essence meant for guiding linguists in their search for syntactic differences.

3.4.4 Step 3: investigating hypotheses

While the findings concerning the patterns in the verbal domain already underline the potential of our proposed method, the third step would be to investigate the hypotheses, as in Figure 3.1. Although step 3 is not necessarily

[15] There actually is a syntactic difference between Czech and English; whereas English has articles, Czech does not. For every occurrence of an English article, there structurally is no article in the Czech translation.

pattern	**total**	**mismatch**	
(total: 388)	EN : NL	EN : NL	χ^2
PROPN	11410 : 5196	6680 : 466	5404
DET NOUN	17730 : 24322	1533 : 8125	4499
ADP DET NOUN	9134 : 14760	1180 : 6806	3963
DET	21947 : 27534	1832 : 7419	3374
ADP DET	10655 : 15549	1383 : 6277	3127
ADP	24336 : 29547	2808 : 8019	2508
PROPN PROPN	3478 : 1015	2597 : 134	2221
AUX PART	1865 : 127	1814 : 76	1598
AUX PART VERB	1824 : 186	1729 : 91	1474
PART	5891 : 3422	3434 : 965	1386

(a) No filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 371)	EN : NL	EN : NL	χ^2
DET NOUN VERB	2764 : 5224	1111 : 3571	1293
ADV	3351 : 5959	1362 : 3970	1276
PRON	715 : 2505	396 : 2186	1241
ADJ VERB PUNCT	245 : 1525	166 : 1446	1016
ADP PART VERB PUNCT	107 : 933	69 : 895	708
PRON NOUN VERB PUNCT	150 : 973	100 : 923	662
ADP DET NOUN ADP DET NOUN PUNCT	998 : 2040	434 : 1476	568
ADP DET VERB	357 : 1253	267 : 1163	561
NOUN	3265 : 1854	2487 : 1076	559
VERB	1816 : 3235	1127 : 2546	548

(b) No filter, with superpattern subtraction.

Table 3.3: (Continued on next page.)

pattern	**total**	**mismatch**	
(total: 188)	EN : NL	EN : NL	χ^2
X PUNCT	326 : 3	326 : 3	317
X	347 : 22	344 : 19	291
PROPN	608 : 296	336 : 24	270
NUM	359 : 656	38 : 335	236
AUX VERB	554 : 261	363 : 70	198
AUX VERB ADP	306 : 87	237 : 18	188
AUX VERB ADP NOUN	256 : 69	198 : 11	167
DET NOUN	1190 : 1474	122 : 406	153
PART	297 : 117	208 : 28	137
DET	1356 : 1624	142 : 410	130

(c) With filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 154)	EN : NL	EN : NL	χ^2
X PUNCT	326 : 3	326 : 3	317
NUM PUNCT	132 : 444	17 : 329	281
DET NOUN VERB	204 : 425	75 : 296	132
ADP DET VERB	33 : 126	12 : 105	74
PUNCT DET NOUN AUX ADP NUM NOUN VERB PUNCT	0 : 73	0 : 73	73
PUNCT DET NOUN AUX VERB ADP NUM NOUN PUNCT	63 : 0	63 : 0	63
ADJ VERB PUNCT	16 : 93	11 : 88	60
ADP DET NOUN	108 : 199	34 : 125	52
SCONJ VERB	73 : 11	68 : 6	52
PRON NOUN VERB PUNCT	12 : 75	7 : 70	52

(d) With filter and superpattern subtraction.

Table 3.3: Top 10 highest ranking differences for English-Dutch. Reported are the four distinct runs, c.q. experiment setups, with the total attested frequencies per language, the mismatching frequencies, written as $x : y$, as well as the χ^2 value for each difference. A mismatch is when a pattern occurs in the one language while being absent in the translation in the other language.

pattern	**total**	**mismatch**	
(total: 340)	CS : EN	CS : EN	χ^2
DET NOUN	5834 : 17730	732 : 12628	10592
DET	9572 : 21947	1351 : 13726	10157
ADJ	25951 : 16772	10326 : 1147	7344
PROPN	4225 : 11410	546 : 7731	6237
PRON	5308 : 13063	972 : 8727	6201
ADJ NOUN	19315 : 12154	7957 : 796	5859
ADJ DET NOUN	2422 : 9134	645 : 7357	5630
PART	480 : 5891	191 : 5602	5054
PART VERB	91 : 4686	39 : 4634	4518
PRON AUX	427 : 5101	121 : 4795	4444

(a) No filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 332)	CS : EN	CS : EN	χ^2
VERB	7378 : 2455	5893 : 970	3531
NOUN ADJ PUNCT	4081 : 1028	3416 : 363	2466
PRON VERB DET NOUN	307 : 2474	182 : 2349	1855
NOUN	6679 : 3378	4945 : 1644	1654
ADJ NOUN ADP NOUN	4013 : 1606	3110 : 703	1519
ADJ ADJ NOUN PUNCT	2250 : 544	1931 : 225	1350
PRON AUX DET NOUN	67 : 1475	42 : 1450	1329
PART VERB DET NOUN	9 : 1293	8 : 1292	1268
ADP ADJ NOUN PUNCT	3182 : 1242	2516 : 576	1217
ADP DET NOUN ADP NOUN PUNCT	315 : 1731	203 : 1619	1100

(b) No filter, with superpattern subtraction.

Table 3.4: (Continued on next page.)

pattern	**total**	**mismatch**	
(total: 241)	CS : EN	CS : EN	χ^2
DET NOUN	464 : 1224	101 : 861	600
PRON	422 : 1131	87 : 796	569
PRON AUX	28 : 459	11 : 442	410
DET	751 : 1420	236 : 905	392
NUM PUNCT	532 : 175	361 : 4	349
X	0 : 346	0 : 346	346
AUX	597 : 1029	55 : 487	344
X PUNCT	0 : 324	0 : 324	324
NUM	586 : 243	355 : 12	321
PART	33 : 377	20 : 364	308

(c) With filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 206)	CS : EN	CS : EN	χ^2
VERB	781 : 258	641 : 118	360
X PUNCT	0 : 324	0 : 324	324
NUM PUNCT	452 : 123	332 : 3	323
NOUN	944 : 460	703 : 219	254
NOUN ADJ PUNCT	295 : 80	244 : 29	169
PRON AUX DET NOUN	3 : 124	3 : 124	115
PRON VERB DET NOUN	36 : 185	24 : 173	113
PRON VERB PRON	5 : 103	4 : 102	91
PUNCT PROPN PUNCT PROPN	91 : 2	89 : 0	89
ADJ NOUN PUNCT	258 : 116	193 : 51	83

(d) With filter and superpattern subtraction.

Table 3.4: Top 10 highest ranking differences for Czech-English. Reported are the four distinct runs, c.q. experiment setups, with the total attested frequencies per language, the mismatching frequencies, written as $x : y$, as well as the χ^2 value for each difference. A mismatch is when a pattern occurs in the one language while being absent in the translation in the other language.

pattern	**total**	**mismatch**	
(total: 254)	CS : NL	CS : NL	χ^2
DET NOUN	5834 : 24322	516 : 19004	17511
DET	9572 : 27534	1127 : 19089	15959
ADP DET	2928 : 15549	561 : 13182	11591
ADP DET NOUN	2422 : 14760	417 : 12755	11557
ADP	17609 : 29547	1611 : 13549	9401
PRON	5308 : 14212	955 : 9859	7331
ADJ NOUN	19315 : 11567	8614 : 866	6332
ADJ	25951 : 17825	10069 : 1943	5497
ADJ NOUN PUNCT	9739 : 4392	6000 : 653	4297
DET ADJ	2026 : 7432	739 : 6145	4245

(a) No filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 252)	CS : NL	CS : NL	χ^2
NOUN	10905 : 3086	8785 : 966	6270
ADP DET NOUN	1414 : 5257	642 : 4485	2881
NOUN ADJ PUNCT	4124 : 1055	3522 : 453	2369
ADP DET NOUN ADP DET NOUN PUNCT	102 : 2040	70 : 2008	1807
VERB	8813 : 5173	5655 : 2020	1722
ADJ ADJ NOUN PUNCT	2164 : 386	1978 : 200	1451
DET NOUN AUX VERB	352 : 2122	225 : 1995	1411
ADP ADJ NOUN PUNCT	3418 : 1286	2696 : 564	1394
PRON	1084 : 3012	703 : 2631	1115
ADP DET ADJ NOUN	311 : 1726	208 : 1623	1094

(b) No filter, with superpattern subtraction.

Table 3.5: (Continued on next page.)

pattern	**total**	**mismatch**	
(total: 121)	CS : NL	CS : NL	χ^2
DET NOUN	286 : 1040	57 : 811	655
DET	446 : 1125	138 : 817	483
ADP DET NOUN	120 : 518	49 : 447	319
ADP DET	139 : 544	62 : 467	310
ADP	562 : 975	102 : 515	276
PRON	274 : 650	81 : 457	263
PUNCT	2140 : 1917	261 : 38	166
PROPN	439 : 215	271 : 47	158
DET ADJ	79 : 290	36 : 247	157
DET ADJ NOUN	83 : 280	32 : 229	149

(c) With filter, no superpattern subtraction.

pattern	**total**	**mismatch**	
(total: 107)	CS : NL	CS : NL	χ^2
NOUN	701 : 437	439 : 175	114
ADP DET NOUN	88 : 261	50 : 223	110
NOUN ADJ PUNCT	187 : 63	145 : 21	93
PUNCT PROPN PUNCT PROPN	87 : 1	86 : 0	86
AUX ADJ	180 : 67	142 : 29	75
DET	103 : 15	99 : 11	70
DET NOUN AUX VERB	13 : 94	9 : 90	66
PRON AUX DET NOUN	3 : 61	3 : 61	53
ADP DET NOUN ADP DET NOUN PUNCT	3 : 56	3 : 56	48
PRON AUX PRON	1 : 50	1 : 50	47

(d) With filter and superpattern subtraction.

Table 3.5: Top 10 highest ranking differences for Czech-Dutch. Reported are the four distinct runs, c.q. experiment setups, with the total attested frequencies per language, the mismatching frequencies, written as $x : y$, as well as the χ^2 value for each difference. A mismatch is when a pattern occurs in the one language while being absent in the translation in the other language.

in the scope of this paper, we will discuss a few patterns to further showcase that this technique delivers useful hypotheses.

English-Dutch

The distributional difference for the pattern DET NOUN leads to the hypothesis that there is a difference between Dutch and English in their use of articles, a very significant one in fact. Inspection of the data suggests that there is indeed a difference in the conditioning of article use,[16] which is confirmed by Donaldson (2008: pp. 25–31) who describes several cases in which Dutch articles behave differently from English articles. However, these mismatches due to conditioning do not make up the largest proportion of the unmatched cases. On the one hand, these are caused by cases of headlinese, where the article is often dropped in English while it remains in Dutch. On the other hand, they are caused by a syntactic difference concerning the Saxon genitive,[17] which takes the position of determiners and is much less prevalent in Dutch, where a prepositional phrase is more common. So, despite the clear influence of headlinese, this pattern still suggests potential syntactic differences.

The patterns ADP DET NOUN and ADP DET hypothesize a difference in the use of prepositions, Dutch using more than English. The data however show, similar to DET NOUN, that a distributional difference is mainly caused by a difference in DET, so in the conditioning of articles, headlinese and the Saxon genitive. It also seems to be caused by a difference in ADP: occurrences in English are often unmatched due to the presence of R-pronouns in Dutch,[18] which are tagged as ADV (e.g. *waarvan* 'of which', in which the preposition *van* is affixed to *waar* 'where'; compare English *whereof*) or compound nouns (e.g. *kredietoverschrijvingen* 'transfers of appropriations'), and occurrences in Dutch are often unmatched due to many verbs having a prefix, which is often a preposition that can be separated from the verb, similar to German (e.g. *aannemen* 'accept', in which the preposition *aan* is separated when the verb is in V2-position: *Het Parlement* ***neemt*** *het mondelinge amendement* ***aan***. 'Parliament accepts the oral amendment.'). Despite several mismatches being caused by either free translations or tagging errors, these differences do point

[16] E.g. from the data:

(5) a. Human **rights** and legal **order** do not prevail.

b. **De mensenrechten** en **de rechtsstaat** worden niet gerespecteerd.
lit. 'The human rights and the legal order are not respected.'

[17] In English, a Saxon genitive is a possessive formed with the clitic *-'s*, e.g. *The king**'s** horse*.

[18] In Dutch, and some closely related languages, the pronominalization of an inanimate complement of a preposition results in an R-pronoun, which is a subtype of pronouns named for their recurring final letter *r*. These R-pronouns then precede the preposition, and are often attached to it in writing. For example, pronominalizing *de tafel* 'the table' in *op de tafel* 'on the table' does not result in **op het* but in *erop*, in which *er* is an R-pronoun. See e.g. Broekhuis (2020) for a more detailed explanation.

towards useful syntactic differences.

Furthermore based on patterns AUX PART, AUX PART VERB and PART, one might hypothesize that there is a syntactic difference with regards to the use of particles such as English *to* and Dutch *om* and *te*. While this is still true, the data do not overwhelmingly confirm this and suggest that the distributional difference is mainly caused by a tagging difference between Dutch and English: whereas Dutch *niet* 'not' is consistently tagged as an adverb (ADV) by UDPipe, English *not* is tagged as a particle (PART) instead. Because of this difference in tagging, PARTs are much more frequent in English than in Dutch (and, conversely, ADVs are more frequent in Dutch than in English; cf. the pattern ADV in Table 3.3), leading to a high χ^2-value. Although these patterns therefore primarily suggest a tagging inconsistency, tagging negation differently between Dutch and English was most likely a solidly justified choice by UD, because English *not* has different syntactic properties than Dutch *niet*. For example, while negation in English triggers do-support, it does not in Dutch, accounting for a major syntactic difference between Dutch and English.

Closer inspection of the highly significant pattern PROPN shows us that it is also caused by a tagging inconsistency. In the English data, (almost) all words with a capital letter are tagged as a proper noun, while their Dutch translations are tagged as nouns or adjectives, in line with their morpho-syntactic properties. The same holds for PROPN PROPN. These patterns therefore do not detect a syntactic difference, but they do point towards an important tagging inconsistency.

Other meaningful hypotheses and syntactic differences were found by nearly all patterns containing a verb or an auxiliary. While the majority of those detected a difference in SOV vs. SVO, the pattern ADP PART VERB PUNCT was caused by a difference in the infinitival complementizer and a difference in separable verbal prepositional prefixes (e.g. *om te handelen.* 'to act.' in which *om* is arguably wrongly tagged as ADP; and *... tegen te gaan.* 'to counter ...'), and the patterns AUX VERB, AUX VERB ADP and AUX VERB ADP NOUN furthermore appear to reflect a difference in auxiliary use, especially the obligatory use of an auxiliary in the future tense in English, where Dutch often uses a simple finite verb.

Other less meaningful candidate differences are suggested by X, NUM, X PUNCT, NUM PUNCT, NOUN PUNCT VERB PROPN and PRON, which were all caused by tagging inconsistencies; in fact, X (PUNCT) and NUM (PUNCT) almost exist in a complementary distribution. Also less useful are perhaps the longer patterns, such as ADP DET NOUN ADP DET NOUN PUNCT, as they are much harder to interpret due to gaps. Nevertheless, this particular distributional difference is mainly caused by the syntactic difference involving the Saxon genitive, as well as a difference in headlinese. The even longer patterns (PUNCT DET NOUN AUX ADP NUM NOUN VERB PUNCT and PUNCT DET NOUN AUX VERB ADP NUM NOUN PUNCT) are only useful because they come in a pair, also in an almost complementary distribution, exemplifying nicely the SOV-SVO word order difference between

the languages.

It appears that filtering the data for syntactically incomparable sentences somewhat influences the usefulness of the returned hypotheses. Although differences due to tagging issues are returned in either setup, they are slightly fewer when filtering. Interpretation of the results also becomes easier. Furthermore, superpattern subtraction influences results considerably, returning patterns in more specific contexts. Through this, patterns returned when subtracting superpatterns more clearly show word order differences, such as SOV vs. SVO. We therefore suggest to filter out syntactically incomparable sentences and to perform two runs; one with and one without superpattern subtraction.

Czech

As for Czech, there are some general conclusions that can be drawn from the comparison with English and Dutch. It turns out that mismatching unigrams are very informative, also because they are much easier to interpret for human observers than complex sequences of POS-tags. Three important syntactic differences could be discovered with unigrams: (i) as opposed to English and Dutch, Czech does not have indefinite or definite articles (as suggested by DET), (ii) Czech allows for pro-drop, i.e. silent subject pronouns when the subject is not stressed, while English and Dutch do not (PRON), and (iii) Czech participles are always adjectival where English and Dutch participles can be verbs or adjectives, showing no adjectival morphology except when used attributively in Dutch (ADJ). In the comparison with Dutch, unigrams additionally suggest that (iv) Czech often uses morphological case where Dutch, lacking such cases, has to use a preposition (ADP). English unigrams furthermore discover that (v) Czech uses verbal affixes for aspectual and temporal distinctions (e.g. perfective and imperfective) where English uses auxiliaries (AUX), and (vi) Czech does not have *to*-infinitivals and has a negative verbal prefix *ne-* instead of a separate negative adverb or particle (PART).

All these findings are confirmed by reference grammars such as Naughton (2005) that mention these features as salient grammatical properties of Czech. They are also supported by longer patterns in the top-10s. Overall, however, in the cases under consideration longer patterns do not seem to add much information to what we can derive from the unigrams alone, except for pattern ADJ ADJ NOUN PUNCT, that discovers that Dutch and English use compound nouns, whereas Czech often uses a noun phrase with adjectives (e.g. *unášené tenatové sítě* : *drijfnetten* : *drift nets*). Nevertheless, where English unigrams are unable to suggest difference (iv), it is discovered by the longer patterns ADP DET NOUN and ADP DET NOUN ADP NOUN PUNCT for English. Similarly, where Dutch unigrams are unable to suggest difference (v), it is discovered by the longer patterns DET NOUN AUX VERB and PRON AUX PRON for Dutch. While difference (vi) is an important difference between Czech and Dutch, too, our method seems to be unable to detect it for that language pair. Some other well-known differences, such as cliticization in Czech

but not in Dutch or English were not found (at least, do not appear in the top 10). It is not entirely clear why this difference was not found, but it is likely caused by tagging; the tagging conventions used may not be sufficiently rich to grasp fine-grained differences as these.

Furthermore, some patterns are less useful. The unigram patterns PROPN, NOUN, VERB, NUM and X detect tagging differences. Similar to the Dutch-English run, English uses more PROPNs while the Czech translations are tagged as nouns or adjectives. A result of this is also that NOUNs are more frequently mismatched in Czech, however closer inspection of NOUN does weakly suggest that Czech uses more nominalizations where Dutch and English use verbs. VERBs are more frequent in Czech, too, which is also due to a tagging difference. While Dutch and English modal verbs are tagged as AUX, they are consistently tagged as VERB in Czech, accounting for the high number of mismatches. NUM and X, similar to what was found in the comparison of English and Dutch, almost exist in a complementary distribution; in fact, the data show us that it often is the case that numerals are tagged as X in English, while being tagged as NUM in Czech. As for longer patterns, it is unclear which difference ADP ADJ NOUN PUNCT and ADJ NOUN ADP NOUN suggest.

It is not surprising that applying superpattern subtraction lowers the number of unigrams in the top 10. While this makes interpretation for the human researcher harder, superpattern subtraction does detect difference (vi) for Dutch, and makes the compounding and nominalization differences discoverable, which had otherwise gone unnoticed. However, we also found that the number of useful patterns goes down, meaning that more noise or irrelevant differences, such as due to tagging inconsistencies, are retrieved. The patterns that are retrieved, though, seem less repetitive, and without superpattern subtraction, patterns often just show that Czech has no articles.

Using the filter, however, yields somewhat worse results. While for Dutch the difference seems insignificant, for English the number of useful patterns interestingly goes down and it strikingly makes our approach unable to detect difference (i). Nevertheless, filtering the data makes the patterns easier to interpret.

3.5 Discussion

Our results show our approach to be effective. Step 2a, in which we run SQS on POS-tag sequences, retrieves POS building blocks of a language, representing each utterance as a sequence of POS tags, which can already be of use to detect broad typological characteristics. In step 2b and 3 we showed and argued that many differences it returns are meaningful and can be used for comparative linguistic research; researchers are pointed in the right direction of where to look for syntactic differences between languages. Apart from that, our approach is able to easily detect tagging inconsistencies between two languages.

Compared to Wiersma, Nerbonne and Lauttamus (2011), our approach is

not subject to a fixed n and can find differences in patterns of variable length, which makes our approach more flexible. Yet, despite our hypothesis that SQS's ability to allow for gaps in the patterns intuitively makes it easier to map differences in e.g. the use of articles, we noticed that gaps can make interpretation a tricky business. We are therefore not entirely certain whether gaps are truly beneficial to the results. While the effects of gaps require further investigation – by for example contrasting our method with a method in which patterns are obtained through and MDL-based, non-gapping pattern mining algorithm – we do believe our approach is promising.

Nevertheless, some caveats and possible points of concern need mentioning. First, tagging influences results. The fact that our approach has proved to be able to successfully identify tagging inconsistencies between two languages means that our approach is sensitive to them, too. If the two languages under investigation have even slightly different annotation guidelines, a NOUN tag in the one language may not fully correspond to a NOUN tag in the other, which will lead to more mismatching occurrences and consequently to patterns with a high χ^2 value that in fact may not indicate a syntactic difference. As pointed out, we found that in English many more words were tagged as PROPN than in Dutch and Czech, despite having clear nominal or adjectival morpho-syntactic properties and the direct translations in the latter two languages were often tagged as nouns or adjectives, capitalized or not. Although it may be true and solidly justified to have the words be tagged as proper nouns within a language's grammar, this inconsistency led to our approach finding many syntactic differences between English and the other two languages – noticing a statistically significant difference in distribution in proper nouns between the languages – that arguably do not signify true differences in the syntactic potential of the languages in question.

Additionally, the quality of the tags influences results down the line, as well. Tagging errors lead to less reliable patterns found by SQS, which in turn influence the usefulness of the differences found. Even if the languages use the same annotation guidelines and have no tagging inconsistencies, if one language has a low tagging accuracy,[19] the patterns found for that language represent syntactic building blocks less reliably. These less reliable patterns lead to less reliable frequencies and less reliable counts of mismatching patterns in step 2b, resulting in noisy χ^2 values. How large the effect of tagging errors on the results really is, however, remains a subject for future investigation.

Similarly, it is fairly straight-forward that the quality of the tags limits our method to finding differences in the information that is put into the tags. Any difference that is not reflected in the POS sequence cannot be detected. If the POS tags are too coarse-grained, it is (almost) impossible to find, for example, the differences in order in verbal clusters between Dutch and German, a difference in case marking, or even a difference in argument order between

[19] This may arise, for example, due to low amounts of data for the model to be trained on, or because the language is morphologically rich, which makes POS tagging more difficult in general.

OSV and SOV languages.

As a final note on tagging, it may be beneficial to remove punctuation from the analysis. Currently, many patterns with a punctuation tag are returned as a significant difference, which may be true between certain languages (e.g. in Czech the subordinating conjunction *že* 'that' is always preceded by a comma, while in Dutch and English its counterpart never is save a few rare exceptions), but it is not necessarily informative syntactically. Removing punctuation altogether, however, could result in unwanted patterns, as the probability of two non-constituent tags being adjacent grows, although this may not be an issue as SQS can already consider them as adjacent by skipping over the punctuation mark with a gap. Leaving PUNCT in the data can also prove useful in the interpretation and investigation of patterns, as it denotes a phrase ending.

Secondly, the statistical test used in our approach is not equipped to detect those cases where the distribution of the pattern is complementary. However, it is not obvious that this will cause serious problems and therefore it may not be necessary to use different (combinations of) statistical tests. An example of a case that at first sight might cause problems is that of Ancient Greek and Turkish articles: whereas Ancient Greek only has definite articles, Turkish only has indefinite articles. This means that in every case Ancient Greek has an article (tagged uniformly as DET in Universal Dependencies), Turkish will not have an article, and vice versa. However, definite and indefinite articles do not occur equally frequently in natural languages.[20] Additionally, the hypothetical problem of this particular example is easily remedied by tagging definite and indefinite articles separately, which underlines the importance of appropriate and consistent tagging.

Thirdly, our approach is not able to detect all patterns and syntactic differences between two languages. In general, some underlying structures or long-distance relations between words such as agreement will not be detected due to the nature of SQS's algorithm, and hence will not be returned as a syntactic difference. Although SQS does allow for gaps in the patterns, which makes the patterns more flexible, these gaps cannot be longer than the pattern itself, limiting the variation and distance over which they can occur.

In the case of our current experimental setup it became clear that some well-known differences between English, Dutch and Czech had gone unnoticed. These missed differences, acting as false negatives, contain for example the difference in cliticization, which occurs in Czech but not in Dutch or English. As mentioned, it is not entirely clear why this difference was not found, but it is likely caused by tagging. It is probably due to the fact that most clitic pronouns were tagged as PRON in Czech, but since many more unmatched PRONs were found in English and Dutch than in Czech (which we explained as being a result of pro-drop being extant in Czech), the difference in cliticization

[20] For example, English *the* occurs roughly 50 million times in the Corpus of Contemporary American English (Davies 2008), while *a* occurs "only" 21.9 million times. Similar numbers are found for Dutch in OpenSoNaR (Oostdijk et al. 2013): *de* and *het* 'the' occur 38 million times, *een* 'a' occurs 11 million times.

probably went unnoticed. This problem could easily be solved, by making the tag set differentiate between clitics and normal pronouns, though. Another difference that was missed, is that of scrambling, a syntactic phenomenon that causes non-canonical word and argument orders, which is possible in Dutch and Czech, but not in English; this was probably not identified in our experiments because syntactic relations between words were not reflected in the POS tags.

In this research we decided against using SQSNorm (Hinrichs and Vreeken 2017). Whereas SQS detects characteristic patterns in one sequential dataset, SQSNorm is designed to capture characteristics of each individual sequential dataset as well as to capture the shared characteristics of multiple datasets. This MDL-based algorithm therefore seems perfect for our task of detecting syntactic differences (as well as similarities) between multiple languages, however we found that SQSNorm was unable to find a difference for a pattern when it occurs in both languages but in different frequencies or distributions. For example, we noticed that SQSNorm detected the pattern DET NOUN to be shared by English and Swedish, implying that there is no syntactic difference. This is because DET NOUN occurs in both English and Swedish, and is frequent enough in both to compress the data well. Hence, SQSNorm fails to capture a significant distributional and syntactic difference, namely that Swedish denotes the definiteness of nouns primarily with suffixes: only when the noun is preceded by an adjective will there be an explicit definite article. For every DET NOUN in English, where there is no adjective and the article is definite, the DET is absent in Swedish. Even though this is a very basic and striking difference between English and Swedish, the nature of SQSNorm's algorithm made it unable to detect it.

As mentioned before, our method can be adapted for use with non-parallel corpora. While step 2a does not require parallel data since this step discovers characteristic patterns for both languages individually, step 2b in its current form does. Applying it to non-parallel data could for example be done by using a permutation test (as Wiersma, Nerbonne and Lauttamus 2011) instead of a McNemar test.

In the future it would be most interesting to enrich the patterns by using multivariate SQS (Ditto; Bertens, Vreeken and Siebes 2016), despite its computational expense. Bertens, Vreeken and Siebes present Ditto, which like SQS finds patterns in sequential data but uses multiple channels of sequential data instead of one. While Bertens, Vreeken and Siebes enrich their textual data with a POS channel to mine for more general patterns in Melville's Moby Dick such as *to:PART VERB a:DET NOUN* (i.e. *to* followed by any verb followed by the indefinite article *a* and any noun, e.g. *to get a broom, to buy (him) a coat*), our approach can benefit from a morphological channel. Using morphological tags and features alongside POS-tags can certainly improve results by being able to find more fine-grained differences, which for example only apply to finite verbs and not to all verbs alike. Note the distinction with running (univariate, i.e. normal) SQS on POS-tags with morphological features: if one would simply attach the feature to the POS-tag, there would be a difference between sin-

gular nouns (NOUN:Num=Sing) and plural nouns (NOUN:Num=Plur), and SQS would treat them as two separate symbols entirely, not knowing that they both underlyingly represent a subclass of nouns. In multivariate SQS, the algorithm would be aware of this fact, because the POS channel would be the same (NOUN) for both singular and plural nouns, while the morphological channel would specify the nouns' number.

Another interesting improvement could be to use hierarchical data instead of linear data. Whereas simple POS-tags are sequential in nature, trees should give more insight in the syntactic differences between languages. Especially when using a dependency grammar such as Universal Dependencies, results can be improved as syntactic relations become the subject of analysis, too. Apart from that, using hierarchical data would solve the problem that SQS also retrieves patterns that are not necessarily constituents. However, to the best of our knowledge an MDL-based pattern mining algorithm does not exist for hierarchical data, and we expect the task to be even more computationally expensive when involving trees instead of sequential data.

Although we do count mismatching occurrences in step 2b, in this approach we do not make use of alignment algorithms: an occurrence of a pattern is considered to be mismatching if there are not as many occurrences of the same pattern in the translation sentence. Effectively it counts the surplus or deficit of a pattern in a sentence pair. Therefore, there may be some noise: a pattern is not considered to be mismatching if there is an occurrence of that pattern in the translation even though they do not actually directly correspond. Consider (6), where the pattern NOUN AUX VERB is present in both English and Dutch.

(6) a. I know that my **neighbour has bought**
PRON VERB SCONJ PRON NOUN AUX VERB

a house.
DET NOUN

b. Ik weet dat mijn buurman een **huis**
PRON VERB SCONJ PRON NOUN DET NOUN

heeft gekocht.
AUX VERB
lit. 'I know that my neighbour a house has bought.'

Due to Dutch's SOV nature these two patterns are not translations of each other, but because the pattern is present in both sentences, it is not counted towards mismatching occurrences. Aligning the data before counting mismatches may solve this, however alignment errors could introduce more noise, as well, especially since alignment algorithms typically require large quantities of data in order to be reliable.

We expected that languages with freer word orders are harder to compress with SQS, showing fewer highly frequent patterns of POS-tags. We indeed noticed a clear tendency: Czech, with its famously free word order, was harder to compress (to 91% of its original size) than English or Dutch, with their stricter word orders (to 81% and 83% respectively, which also reflects Dutch's slightly freer word order). We did not further investigate a correlation between the compression rate and a language's free word order, but if such a correlation exists, we could use the minimum description length principle to quantify the freeness of a language's word order. This serendipitous find remains the subject of future research.

3.6 Conclusion

In this paper we have introduced a new approach to automatically detect syntactic differences between languages by using the Minimum Description Length principle. The approach proved useful in both retrieving POS building blocks of a language as well as pointing to meaningful syntactic differences and tagging inconsistencies. Apart from that, we believe MDL is widely applicable to natural language tasks, from translation studies to the quantification of word-order freeness in a language. Despite a clear sensitivity to tagging accuracy, our results and approach are promising.

CHAPTER 4

Detecting syntactic differences automatically using word alignment

Author contributions: MK, SB, JO and SvdP conceptualized the research; MK designed the algorithms, wrote the tools, analyzed the data, and wrote the paper; SB, JO and SvdP supervised and critically reviewed the research.

Abstract

The key question of this Chapter is whether extensive linguistic knowledge about a language can be leveraged in order to detect grammatical properties of a less well-described language and differences between the two languages. To this end, word alignment is used to map source language words to target language words with the aim of detecting syntactic features of the target language and differences between source and target language by semi-automatically analysing this mapping. Three tools are developed to detect syntactic properties and differences. The tools are evaluated on the language pair English-Hungarian. It is concluded that the tools can be used effectively to form many correct hypotheses on differences between the languages in several syntactic domains, though some room for improvement remains.

4.1 Introduction

In the previous Chapter the possibility of using the Minimum Description Length principle in the automatic detection of syntactic differences was invest-

igated. The key question of this Chapter is whether extensive linguistic knowledge about a language can be leveraged in order to detect morpho-syntactic features[1] of another, less well-described language and differences between the two languages. It is assumed in this research that knowledge about only the source language is available, while no knowledge about the language under investigation (the target language) is available and the utterances in a corpus are not enriched with grammatical information, reflecting an extreme case of investigating an under-resourced and under-researched language. By aligning the utterances in a parallel corpus on a word level, the knowledge about the source language can be analysed automatically and mapped onto the target language in order to arrive at conclusions about morpho-syntactic properties of the target language.

For the purpose of the detection of morpho-syntactic properties of the target language, as well as differences between it and the source language, a three-step process is proposed: Preprocessing, Attribute extraction and Discovering features; cf. Figure 4.1. For the last step, three distinct tools were developed: the Data Grouper for Attribute Exploration, the Generalization Tree Inducer, and the Affix-Attribute Associator.[2] Section 4.2 consists of an extensive description of the overall proposed method, as well as detailed descriptions of the workings of the developed tools.

The remainder of this Chapter consists of a description of the setup for the evaluation of the process and tools (Section 4.3), a detailed results section of said evaluation (Section 4.4), the discussion of the proposed method and its results (Section 4.5), and a concluding section (Section 4.6).

4.2 Method

The proposed approach assumes zero knowledge about the target language, while assuming the availability of linguistic knowledge about a different language, henceforth the source language, as well as the availability of natural language processing tools, such as parsers and taggers, for the source language. In order to be able to conclude anything about the morpho-syntactic nature of the target language or to be able to extract any morpho-syntactic differences between the source language and the target language, there must be a mapping between the two languages. In this approach this mapping is achieved by leveraging parallel data and using bitext word alignment. These alignments are combined with the linguistic annotation of the words in the source language, leading to suggestions for morpho-syntactic features of the target language for a linguist to investigate. In this section we describe the process of going from raw parallel text corpora to the extraction of morpho-syntactic features of the target language and differences between it and the source language.

[1] Recall that by morpho-syntactic features I mean all morphological and all syntactic properties of a language. This reading is used throughout the dissertation.

[2] The code is made available on `https://github.com/mskroon/DeSDA`

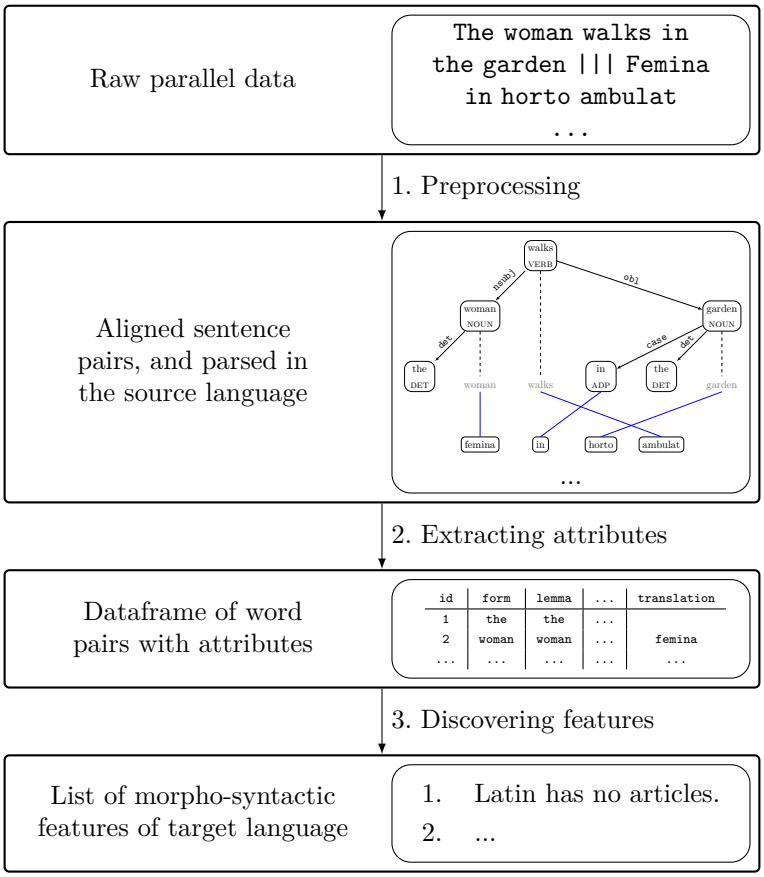


Figure 4.1: A list of morpho-syntactic features of the target language and differences between it and the source language is extracted from raw parallel data. The method consists of three steps: Preprocessing, Extracting attributes and Discovering features.

This process is divided into three steps, as illustrated in Figure 4.1. In the first step, Preprocessing, raw data is aligned on word level and, for the source language, parsed and tagged. In the second step word-internal and contextual morpho-syntactic attributes are extracted from the dependency parses to create a dataframe of words and attributes. In the third and last step three newly developed tools process the dataframe in order to detect morpho-syntactic features of the target language and differences between it and the source language. All of these steps will be described in detail below.

4.2.1 Preprocessing

First, the data of both the source and the target language need to be tokenized. In the current setup this is done with a language-independent tokenizer, that separates tokens based on whitespace, and splits punctuation symbols from tokens to treat them as separate tokens, and lower-cases words. However, a language-specific tokenizer may be more appropriate, depending on the language and the research goals.

Next, the parallel data are aligned on a word level. In principle any alignment algorithm or tool can be used – for the purpose of this research it was opted to deploy `eflomal` (Östling and Tiedemann 2016),[3] short for Efficient Low-Memory Aligner, a well-established statistical aligner that outperforms other popular statistical aligners such as `fast_align` (Dyer, Chahuneau and Smith 2013) and Giza++ (Och and Ney 2003) in both speed and alignment quality.

The task of word alignment can be defined as identifying which words in a translationally equivalent, parallel sentence pair correspond to each other. This is a notably hard problem, because it often involves word order differences, word omissions or insertions, and single words corresponding to multiple words, or a phrase. Due to this and a general danger for high computational complexity, there has been extensive research on the task (cf. Och and Ney 2003 and Tiedemann 2011, who give good overviews and descriptions of existing alignment models), in which three distinct families of approaches can be identified: heuristic, statistical and neural.

Heuristic models are the simplest, as they obtain word alignments through the 'similarity' between words of the two languages. One could think of applying the Dice coefficient (Dice 1945), which quantifies the similarity between two samples based on the intersection of the sample sets; in the task of word alignment, this straightforwardly constitutes the number of sentence pairs in which a word of the source language and a word of the target language occur together, relative to the total number of sentence pairs in which the words occur, whether alone or together. The higher this coefficient, the more often two words occur together, relatively, which indicates they may be each other's translations. While heuristic models are easy to implement and interpret, the problem with heuristic models is that the choice of similarity measure is arbitrary (Och and Ney 2003).

Statistical models, in comparison, have measures that are more soundly defined in probability theory, and often outperform simple heuristic models. They are distinguished by the fact that the alignments are the result of statistical estimation of a generative translation model that generates the target language sentence from the source language sentence using a set of latent alignment variables (Östling and Tiedemann 2016). The word alignments for the sentence pair (i.e. the latent variables) are then inferred from the generat-

[3] The source code and documentation of `eflomal` can be found at `https://github.com/robertostling/eflomal`

ive model, typically using a form of the expectation maximization algorithm (Dempster, Laird and Rubin 1977). The best alignments for the sentence pair are then those that return the highest probability of the source language sentence generating the target language sentence. The inference, however, can be done in multiple ways, and many extensions or adaptations to a model using the expectation maximization algorithm have been proposed, among which is Östling and Tiedemann (2016: `eflomal`), who use a Bayesian model with Markov Chain Monte Carlo inference.

Recent years have seen the rise of neural approaches in word alignment, specifically those using word embeddings to retrieve word alignments. An example of a recent neural aligner is SimAlign (Jalili Sabet et al. 2020), which uses the cosine similarity between the word vectors to obtain the word alignments, in a way reminiscent of existing heuristic approaches. Neural approaches such as SimAlign seem to outperform statistical approaches and have the advantage that the embeddings can be trained on non-parallel data. However, word embeddings are famous for requiring vast amounts of data (usually in the order of millions of sentences) to achieve good quality embeddings. Apart from that neural approaches are much more computationally demanding than statistical approaches.

The advantages and disadvantages of different approaches, then, leave neural approaches to be most effective for language pairs for which parallel data are not sufficiently abundant, while very large non-parallel corpora exist for both languages separately. Statistical aligners give good results – and are faster – for language pairs that do have sufficiently abundant parallel data. Therefore, seeing as the quantities of data large enough to train good quality word embeddings may not be available for most languages, especially those that may be of specific interest to comparative syntactic research, `eflomal` was used in this research, also considering that the existence of sufficient parallel data in order to extract syntactic differences was a prerequisite in the setup of this research.

After alignment, the data of the source language are parsed in Universal Dependencies (UD) (Nivre et al. 2016), with UDPipe (Straka and Straková 2017). Dependency parses are used, as opposed to constituency parses, because dependency parses directly and explicitly contain syntactic relations between words, which were considered to be essential for the purposes of this research. Having access to the syntactic relations between words allows the linguist to detect differences in the order of arguments, or the position of functional elements relative to their heads. In parsing, the UD programme's annotation conventions were followed, since it is one of the most widely used dependency-grammar programmes, but in practice any dependency programme could have been used.

UDPipe is a well-established dependency parser for UD, for which many pre-trained models are available. Easy to implement with binaries in many programming languages readily available, UDPipe achieves (near) state-of-the-art parses, however sentence parses are rarely completely perfect. Depending on the model used, the labelled attachment scores (a standard measure in dependency parsing that corresponds to the percentage of words that were

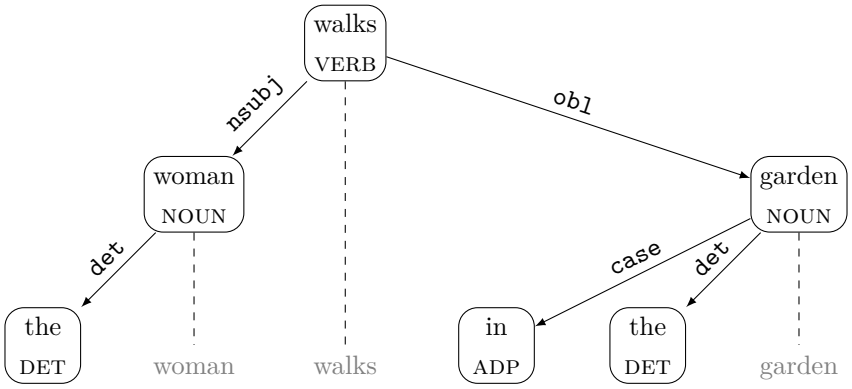


Figure 4.2: An example of an English sentence parsed in UD.

attached to the correct syntactic head with the correct syntactic relation or label) for English range between 82 to 86 per cent.

Additional to inducing a dependency tree with syntactic relations between words, UDPipe also lemmatizes, tags words for part-of-speech (POS) and provides morphological tags, which allows for the generalization over word categories and morpho-syntactic sub-categories. The accuracy of these POS tags and morphological tags range between 93 and 96 per cent for English models.

4.2.2 Extracting attributes

After tokenization, tagging and parsing, words of the source language have several attributes attached to them. These annotated words are passed on to the tools in the next step as rows in a dataframe; each row then contains a token with its attributes.[4] In this subsection all attributes in the dataframe will be described, some of which are deduced from context in the dependency parses or from the alignments.

As mentioned above, UDPipe parses sentences in UD, lemmatizes and tags words for POS and morphological tags. The result is formatted in CoNLL-U by default. The following relevant fields in the CoNLL-U output are taken up in the dataframe as columns, i.e. attributes:

`id:` contains the index of the token in the source sentence.

`form:` contains the form of the token in which it is encountered in the source sentence.

`lemma:` contains the lemma of the token.

[4] A dataframe is an efficient container object, effectively a table with labelled rows and columns. The dataframe is implemented in Python using pandas (Reback et al. 2021). The algorithms furthermore rely on `networkx` (Hagberg, Schult and Swart 2008) to efficiently process the dependency parses as graphs.

`pos:` contains the part-of-speech tag (POS tag) of the token in the context of the sentence.

`deprel:` contains the dependency relation between the token and its head. If it does not have a head, the deprel is `root`.

`feats:` contains morphological features of a token, such as singular number or third person. The complexity of these features varies from language to language.

Additionally, the attribute `translation` is added, which contains the word in the target language with which the source word was aligned. If a source word is aligned to multiple target words, all alignments are added, in which case the order of the target words is retained. For example, if the English preposition *around* is – correctly – aligned to the Dutch circumposition *om ... heen*, the `translation` field of *around* would be the list `[om, heen]`, and not `[heen, om]`.

As Kroon et al. (2020) already observed (i.e. Chapter 3), though, UD's categories (POS, morphological tags and syntactic relations) may be too coarse-grained to extract syntactic differences between languages with high precision. For example, verbs are not tagged for transitivity, but the transitivity of verbs is related to some specific morpho-syntactic differences between languages, chief among which is perhaps ergativity, in which the subject of an intransitive verb takes the same form as the object of a transitive verb, which is distinct from the subject of a transitive verb.

In order to detect differences with higher precision later, the UDPipe parses and tags are 'enriched' by adding some additional annotations that can be deduced from the trees. Among these enrichments verbs receive an additional tag for transitivity. Whenever a word in the source language is tagged as a verb, the algorithm automatically adds the sub-label `Trans` to the POS tag if it has a daughter node in the dependency tree with the dependency relation `obj` (used to denote the direct object relation between a nominal word and an active verb) or `nsubj:pass` (used to denote the subject relation between a nominal word and a passive verb).[5] Whenever a word is labelled as a verb but does not have any daughter node with one these relations, the algorithm automatically adds the sub-label `Intrans`. This is done so as to be able to distinguish between transitive and intransitive verbs in later stages.

Furthermore, for words that have the `conj` relation to their mother node in the dependency tree, denoting a conjunction relation, the dependency relation of their closest ancestor node that does not have the `conj` relation is percolated down and added as an additional relation (which in practice usually is their mother node's dependency relation, except for in nested summations). For example, in Genesis 1:1 (*"In the beginning God created the heavens and the earth."*), *the heaven* and *the earth* are conjoined. In UD *earth* receives the

[5] In cases such as *He was given a book*, the verb *given* also receives the sub-label `Trans`, because it has both an `obj` and an `nsubj:pass` daughter.

tag `conj`, being in a conjunction relation with *heaven*, while *heaven* is in an object relation to its mother node, *created*. The enrichment is then achieved by percolating the `obj` relation down, such that *earth* now has the relation `obj:conj`. A similar approach is deployed in Odijk et al. (2017), who describe that PaQu counts every conjunct in a subject conjunction as a subject of the verb, as well, a strategy they also deem reasonable and very useful. Percolating relations down opens up the possibility to distinguish between conjoined words, while still identifying their actual syntactic function. For the purposes of this research, it is mostly relevant for verbs, which give rise to a variety of syntactic differences between languages concerning conjunction – for instance, some languages may readily use participles instead of conjoining finite verbs, or may express specific instances of conjunction with a specific verbal form, such as the *te* form in Japanese.

On top of these CoNLL-U attributes and 'enrichments', a few more contextual and structural attributes that are of special interest in the detection of syntactic differences are derived from the trees: parents, children and crossings. These are explained below.

Parents and children

In order to connect possible morpho-syntactic differences to structural context, words receive two more attributes: one containing the POS tag and the dependency relation of its parent in the dependency parse; and one containing the POS tag and the dependency relation of all its children in the dependency parse. Having access to these structural contexts was deemed relevant, because knowing, for example, which personal pronouns are children of a verb, either as a subject or an object, can give a linguist all the necessary information to detect verbal paradigms or object agreement; or having direct access to a determiner's or adjective's parent's dependency relation can be telling in whether determiners or adjectives agree with their heads.

The algorithm distinguishes, however, between open and closed word categories when extracting the information of parents and children, which helps with the generalization over word classes while still retaining specificity regarding function words. Additional to the POS tag and the dependency relation, a parent or child's lemma is also extracted if its POS tag is a closed class. In this distinction, the algorithm follows the UD programme's line in their classification of open and closed word classes.[6] For illustration, consider the sentence *The woman walks in the garden* (see Figure 4.2); the word *garden*'s parent would be extracted as `VERB|root`, while its children would be extracted as `[in|ADP|case, the|DET|det]`. This allows the linguist to better distinguish

[6] The closed word class in Universal Dependencies contains the following POS tags: ADP (adpositions), AUX (auxiliaries and modals), CCONJ (coordinating conjunctions), DET (determiners, including articles and demonstratives), NUM (numerals), PART (particles), PRON (pronouns, whose subclassifications are encoded as features), and SCONJ (subordinating conjunctions).

between analytical and synthetic representations of grammatical features.

Crossings

With word order differences being a specific point of interest in comparative syntax, detecting crossing constituents or words between languages is desired, if not necessary. The alignments are therefore combined with the linguistic annotation of the source language in order to discover word order differences pertaining to specific morpho-syntactic attributes.

This is done by first checking whether there are crossings among the alignments within a sentence pair. Each alignment technically consists of a pair of indices (i, j), in which i refers to the ith word in the source language utterance (this i is identical to the `id` attribute) and j to the jth word in the target language utterance. For a pair of alignments (i, j) and (p, q), if $i < p$ (so, the ith word is on the left of the pth word in the source language), there is a crossing if $j > q$ (so, the jth word is on the right of the qth word in the target language). Similarly, there is also a crossing if $i > p$ and $j < q$. Note that if either $i = p$ or $j = q$ there is a many-to-one alignment; these cases are not considered to be crossings.

If a crossing is discovered for a pair of aligned words i and p, this is recorded for the words with `id` i and p. Each word in the source language is thus given a set of `id`s of other words in the same sentence, of which the alignments appear on the other side of the word's alignment in the target language.

For example, Figure 4.3 shows a sentence pair of English and Latin: *The woman walks in the garden* vs. *Femina in horto ambulat*. In this example English is taken as the source language, meaning that the sentence is parsed in UD, while Latin is taken as the unannotated target language. In the example, the following alignment pairs are found: $(2, 1)$, $(3, 4)$, $(4, 2)$ and $(6, 3)$. Of these alignments $(3, 4)$ and $(4, 2)$ cross, because $3 < 4$ and $4 > 2$. The alignments $(3, 4)$ and $(6, 3)$ also cross, because $3 < 6$ and $4 > 3$. Replacing the indices by the actual words, this in other words means that *walks* appears on the left of English *in*, while *ambulat* appears on the right of Latin *in*; and *walks* appears on the left of *garden*, while *ambulat* appears on the right of *horto*. It is then temporarily recorded for each English word whether it crosses and with what: the 3rd word (*walks*) crosses with the 4th word (*in*) and the 6th word (*garden*), the 4th word (*in*) crosses with the 3rd word (*walks*), and the 6th word (*garden*) crosses with the 3rd word (*walks*).

For each of these words, the shortest paths between it and the words with which it crosses are calculated, and are added as an attribute in the dataframe. In so doing each word (i.e. step) in the shortest path is retrieved as the dependency relation of the word to its mother, indexed with the depth of the step, which corresponds to the number of downward steps between the word and the start point of the path (upward steps are represented with a negative index). These paths are then sorted on the linear order of the words within the utterance. To illustrate this, let us consider Figure 4.3 again. *Walks* crosses

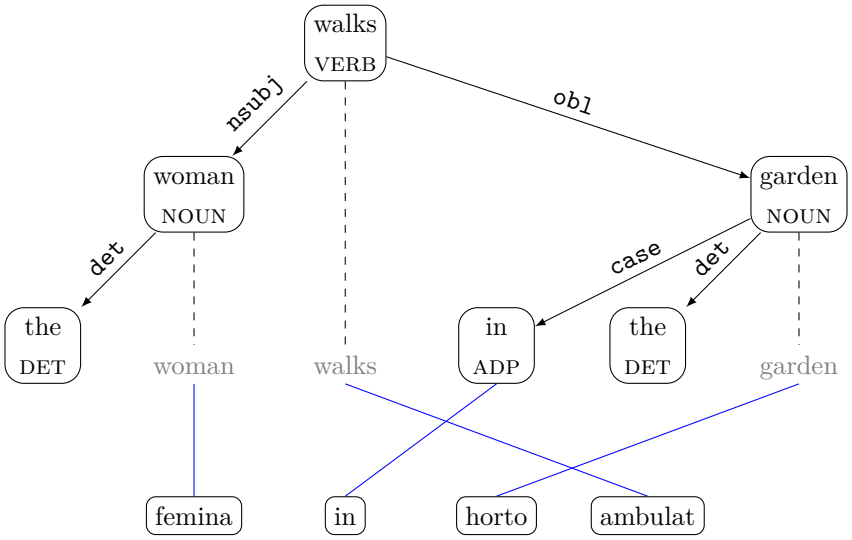


Figure 4.3: An English-Latin sentence pair. English acts as the source language, having full linguistic annotation, with a dependency tree in UD. Word alignments are indicated in blue. It can be seen that there are two crossings: $(walks, ambulat)$ crosses (in, in) and $(garden, horto)$.

with *in*. The shortest path from *walks* to *in* is *walks*—*garden*—*in*. Every word being retrieved as the dependency relation, this path becomes `root`—`obl`—`case`. Adding depth to this path renders it $\texttt{root}^0$—$\texttt{obl}^1$—$\texttt{case}^2$, meaning that from `root` it is one step down to `obl`, and two steps down to `case`. Finally sorting it on the linear order of the words in the utterance, would render it $\texttt{root}^0$—$\texttt{case}^2$—$\texttt{obl}^1$. So, as a `root`, *walks* crosses with a granddaughter `case` node, which is a daughter of an `obl`, and both of them are to its right. Crossing paths like these are useful: they show the linear order of the words and due to the depth indices the dependency structure is still retrievable.

For specific cases, however, a slightly different strategy is followed. First, in the case of crossings between nodes n_1 and n_2 that are siblings or where n_2 is a descendant[7] of a sibling of n_1, the shortest path is calculated up to the lowest common ancestor[8] in the tree and then down to n_2. For example a subject and an object swapping places, which are each other's siblings as they are both child nodes of the `root`, would render the crossing $\texttt{nsubj}^0$—$\texttt{root}^{-1}$—$\texttt{obj}^{1*}$, specifying the one step up from the subject to the root verb and then one step down from there again, indicated with the asterisk. It would also render the

[7] A descendant of node n in the dependency tree is any other node that is a child node of n or a descendant of a child node of n.

[8] An ancestor of node n in the dependency tree is any other node that is a parent node of n or an ancestor of a parent node of n. The lowest common ancestor is the ancestor node shared between two (or more) nodes that is lowest in the tree.

reverse, with the `obj` as starting point. This example could indicate that the sentence in the source language has an SVO order, while the target language, swapping subject and object, has an OVS order.[9] When this particular crossing is encountered often, one could hypothesize that the target language has the base order OVS in general.

Secondly, let us consider Figure 4.3 once more. Four crossings are found in total:

1. $\texttt{root}^{0}$—$\texttt{obl}^{1}$
2. $\texttt{root}^{0}$—$\texttt{case}^{2}$—$\texttt{obl}^{1}$
3. $\texttt{root}^{-1}$—$\texttt{obl}^{0}$
4. $\texttt{root}^{-2}$—$\texttt{case}^{0}$—$\texttt{obl}^{-1}$

Notice that 3. and 4. are the mirrors of 1. and 2., respectively. The only differences are the start and end points in the paths; in 1. and 2. one walks down from `root`, in 3. and 4. one walks up from `obl` and `case`. Now, 2. and 4. could be considered superfluous. Seeing as *walks* already crosses with *garden*, it can be expected that *walks* would cross with all of *garden*'s children nodes, too. In order to simplify the output, a crossing between word i and one of its descendants d is ignored if i also crosses with d's direct mother node in the tree. Inversely, a crossing between word i and one of its ancestors a is ignored if i also crosses with a's direct daughter node in the tree. As for crossings between word i and its siblings – or its, i.e. the sibling's, descendant – s, they are ignored if i also crosses with s's direct mother node, similar to crossings with ancestor nodes.

This reduction of the output was deemed reasonable. However a caveat: it is not necessarily a given that, if there is a crossing between two words, the words also cross with each other's children. Especially relevant in the case of extrapositions or any other form of discontinuity, consider a sentence pair such as English-Dutch *I saw a man who lives in Amsterdam : ik heb een man gezien die in Amsterdam woont*, in which the main verb (*saw : gezien*) appears to the left of the object in English, but to the right of it in Dutch. However, the relative clause (which is a daughter node of the object noun) appears to the verb's right in both languages. In this case, there are two crossings: *saw* crosses with the determiner *a* and the object noun *man*, of which the former is ignored because *a* is a child node of *man*, which also crosses with *saw*. The result is that only the crossing between *saw* and *man* is outputted, however this does not imply that there is a crossing between *saw* and all of *man*'s children: the relative clause does not cross. The output reduction therefore still retrieves the relevant crossings, but the discontinuity of the phrase and the interfering

[9] Not necessarily, however, since we do not know anything about the syntactic structure of the target language. It could be that the source language has an active sentence, while the target language has a passive sentence, in which case the order of the participants would be swapped, as well.

material are not highlighted.[10] A user must be aware of this behaviour, as it may cause for extrapositions to go unnoticed.

It similarly holds true that it is not necessarily a given that, if two words do not cross, there is also no crossing between them and any of the other's children. However, these cases do not cause any issue with the algorithm. Consider for example colloquial Russian *čto ona krasivuyu videla devušku* lit. 'that she (a) beautiful saw girl'. In this example the main verb interferes between the adjective and the object noun. Aligning this sentence with its English translation on word level shows that *saw* and *girl* do not cross, because the relative order of the Russian words to which they were aligned (*videla* and *devušku*) is the same: the object follows the verb. However, this does not imply that there is also no crossing between *saw* and any of *girl*'s descendants. In fact, *saw* crosses with *beautiful*, because *saw* appears to the left of *beautiful* while in the Russian sentence the order of the equivalent words (*videla* and *krasivuyu*) is reversed. No output reduction takes place, however, because *beautiful*'s mother node *girl* does not cross with *saw*, and the crossing between the verb and a daughter node of the object is correctly retrieved as $\texttt{ccomp}^0$—$\texttt{amod}^2$—$\texttt{obj}^1$.

As a final remark on crossings, a word's crossings are split into three categories: ancestor crossings, containing crossings with words that are its ancestors; descendant crossings, containing crossings with words that are its descendants; and sibling crossings, containing crossings with words that are its siblings or descendants of its siblings. This is necessary in order to be able to quickly distinguish between the types of crossings, and to see what kind of material a word crosses with.

4.2.3 Discovering features

After preprocessing the data and extracting attributes from the dependency parses, the dataframe is ready to be explored, and morpho-syntactic features of the target language and differences between it and the source language can be extracted from it. This is done with the help of three different tools that the author developed for this purpose: the Data Grouper for Attribute Exploration, the Generalization Tree Inducer and the Affix-Attribute Associator. The Data Grouper for Attribute Exploration, or DGAE, gives a breakdown of how often each morphological feature, crossing and translation (i.e. each meta-data attribute) occurs by *grouping key*. A grouping key can be any attribute or combination of attributes, such as POS tag, dependency relation or the combination of POS tag and dependency relation. These breakdowns can quickly provide insight in the prevalence of, e.g., determiners or adpositions in the source language that are not aligned to a word in the target language, indicating the absence of articles or the presence of cases (e.g. aligning a target language without articles to English, leaves the vast majority of articles to be

[10] In fact, even if the output reduction was not performed, the discontinuity of the phrase and the interfering material would not be highlighted.

unaligned, and quickly accessing the information that indeed very many articles are left unaligned, allows for the linguist to draw conclusions about the existence of articles in the target language). The Generalization Tree Inducer, or GTI, creates a tree based on the conditional entropy of attributes in order to better explore the co-occurrence of attributes. Though, whereas for a decision tree the most favourable split is the one that gives the highest information gain, which should lead to the correct classification as quickly as possible, this tool builds a tree by considering the most favourable split to be the one with the lowest information gain, which should lead to better generalization as opposed to identification. Finally, the Affix-Attribute Associator, or AAA, attempts to discover productive affixes in the target language and to relate them to morpho-syntactic attributes of words in the source language. All tools are explained in more detail below.

Data Grouper for Attribute Exploration

Simple yet insightful exploration of the data can already be done by means of a tool that gives attribute frequency breakdowns of the data or parts of the data. Splitting up the data, or rather grouping the observations, based on the value of an attribute can lead to the discovery of high co-occurrences between attributes. The attribute on which a split or grouping is based shall be referred to as a *grouping key*. As said above, a grouping key can be any attribute or combination of attributes, such as POS tag, dependency relation or the combination of POS tag and dependency relation, over which the data is partitioned. Patterns may arise when taking the dataframe and grouping all words by a specific attribute. For instance, grouping the data by POS tag should quickly show that pronouns are very likely not to be aligned to a word in the target language if the target language has pro-drop and the source language does not.

The author developed a tool that does exactly this: DGAE. While DGAE allows the user to group by any attribute or combination of attributes, grouping by POS tag, dependency relation or the combination of the two is probably the most useful in the case of discovering morpho-syntactic features of a language. For example, a (toy) dataframe such as the one in the top in Figure 4.4 can be grouped on the value of the dependency relation column, resulting in three smaller dataframes. In the middle, smaller dataframe a clear pattern can be observed: all `obj` nouns end in *-m* in Latin.[11] Per group, DGAE then gives a frequency breakdown of which attributes, including translations and crossings, occur with it, as shown for the `nsubj` group. It additionally gives the 20 most frequent attribute bundles in the group – i.e. which specific combinations of attributes occur most frequently – for better insight in the attribute distribution within the group, but this is not shown in the Figure.

[11] This is only the case because all words are singular in this toy example. Plural objects tend to end in *-s* or *-a* in Latin.

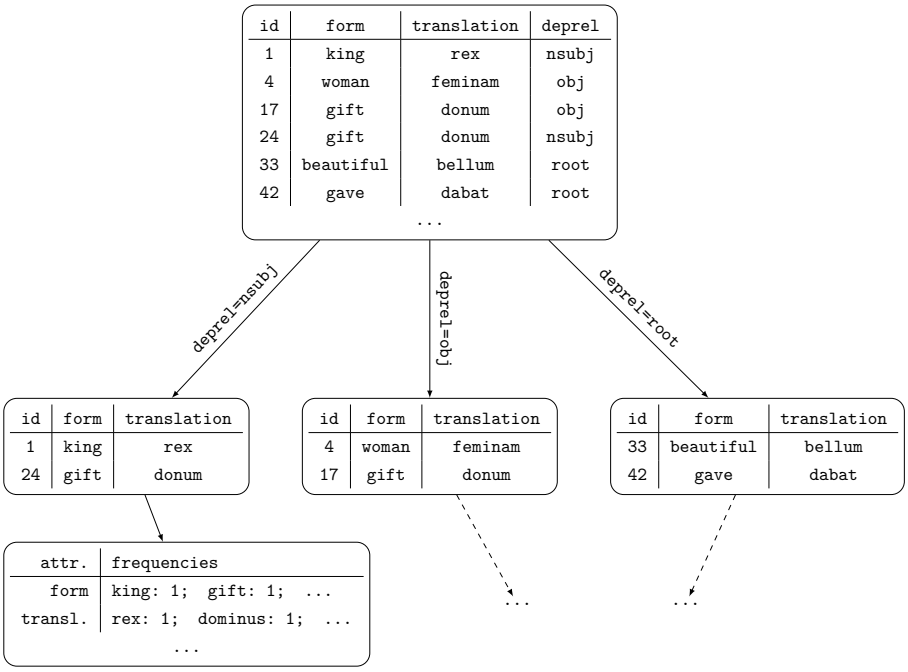


Figure 4.4: An example of grouping a dataframe by the value in the `deprel` column, short for dependency relation. A clear pattern emerges for the `obj`: all *translations* end in *-m*. DGAE then gives frequency breakdowns of attributes of the partitions, illustrated for the leftmost partition.

Some attributes can contain multiple values, such as the `feats` attribute that contain morphological features of a word as tagged by UDPipe. For these multi-valued attributes, DGAE does not count the frequency of unique feature bundles, but of the separate features instead. So, if the dataset consists of the two Latin words *anni*, the genitive singular – `(Gen, Sing)` –, and *annorum*, the genitive plural – `(Gen, Plur)` – of *annus* 'year', the frequency breakdown would record that `Gen` occurs twice, and `Sing` and `Plur` both once.

Generalization Tree Inducer

The author also developed GTI. The goal of GTI is to structure the data, in order to explore it in more detail and to be able to investigate whether certain attributes often co-occur. The data are grouped over the attributes in an attempt to generalize.

In machine learning, decision trees are a popular choice in classification tasks, where they predict the value of a target variable (such as the language in which a sentence was written in the case of language identification) on a set of observed features. They iteratively partition the data over the observed

features in order to arrive at groups in which as many items as possible have the same target value. GTI was built on this property of iteratively grouping and structuring the data, with two differences.

First, whereas decision trees canonically aim at partitioning the data based on the "most distinctive" feature, GTI aims at partitioning the data based on the least distinctive feature. This is done because the goal is not to obtain groups with a homogenous target variable, but to obtain groups with homogenous features.

In decision trees, the "distinctiveness" of a feature is usually described in terms of their influence on the *entropy* of the target variable in the partitionings. First introduced by Claude Shannon (Shannon 1948), entropy is an information theoretic term, and is often interpreted as the expected surprisal over an outcome of an event, or the amount of chaos in a system.[12] The higher the entropy, the less certain one is over the outcome of an event, meaning that there is much variation in the value of a variable. This "distinctiveness" of a feature, then, is the amount by which it reduces the entropy of the target variable – in other words, how much more it makes the outcome of the target variable homogenous.[13] GTI therefore partitions the data over the feature that reduces the entropy the least.

Secondly, in the task at hand, there is no formal target variable that needs to be predicted. In GTI, the role of target variable is therefore filled by a unique identifier for each observation (in casu, a token plus its attributes).[14] Effectively, the result is that GTI tries to group words into as large as possible groups.

The expected behaviour of this algorithm is then that it would detect "stable" features that show little variation. For instance, it can be expected that it would partition the data on POS tag very early. With the help of GTI, one can expect to find groups of words with many common features, which helps to structure the data.

However, to help the researcher explore the data more efficiently, GTI allows for the data to be pre-partitioned, for example by grouping words by POS tag. GTI is then run on each POS tag separately, which allows for the gener-

[12] The entropy H of variable X (with possible outcomes $x_1, ..., x_n$, which occur with probability $P(x_1), ..., P(x_n)$) is defined as

$$H(X) = \sum_{i=1}^{n} P(x_i) \log P(x_i)$$

[13] The amount by which it reduces the entropy is also known as *information gain*, which is defined as the difference between the entropy of a system and the entropy of a system given the outcome of another variable or the value of a feature:

$$IG(X, a) = H(X) - H(T|a)$$

[14] In fact, in this case the information gain of a feature is equal to its entropy. GTI therefore partitions the data based on the value of the attribute that has the lowest entropy.

alization over, for example, all nouns. This method produces a large file with nesting levels of indentation to represent the hierarchy in the generalization tree. Additional to the indentations, the file also contains the 20 most frequent feature bundles inside a partition (per indentation level), and it lists the (non-zero) entropies of the remaining attributes, for better exploration. Output can be suppressed to discard low-frequency data from which it is difficult to draw reliable conclusions, but this is optional and the parameters can be chosen by the user; the default settings do not print partitions that contain fewer than 5 observations or contain less than 1% of the observations of the partition one level higher (i.e. are a partition based on an attribute value that has a less than 1% probability).

It should be remarked that for multi-valued attributes, such as the `feats` attribute, the entropy calculated is the joint entropy of the technically multivariate distribution. That is to say, the entropy is calculated using the probabilities of the unique feature bundles, and not using the probabilities of the separate features. So, if the dataset consists of the two Latin words *anni*, the genitive singular – `(Gen, Sing)` –, and *annorum*, the genitive plural – `(Gen, Plur)` – of *annus* 'year', the entropy of the `feats` attribute would be 1, as calculated with the probability of `(Gen, Sing)`, 50%, and the probability of `(Gen, Plur)`, 50% – and not with the probability of `Gen` (50%), `Sing` (25%) and `Plur` (25%) separately. However, partitioning *is* done over the separate features. This allows for easier generalization over all singular nouns, for example.

Affix-Attribute Associator

Finally, the author developed AAA, that aims to generate hypotheses about which character sequences, or strings, could be affixes in the target language, and to associate them to morpho-syntactic attributes in the source language. It extracts all string pre- and suffixes (including full words) from the target language, without length restrictions, and all attribute subsets from the source language, in which it maintains a minimum frequency on both the strings and the attribute subsets in order to suppress the looming combinatorial explosion. The default minimum frequencies are 100: both strings and attribute subsets must occur at least 100 times in order to be included in the set of generated affix hypotheses. This minimum frequency is a parameter to be chosen by the user, though.

As for the attribute subsets, recall that some attributes can contain multiple values, such as the attribute `feats`. In extracting attribute subsets, AAA considers the words' full attribute bundles, in which the multi-value attributes have been flattened (i.e. the "brackets have been removed"). For example, Latin *anni* 'year' has `feats` attribute `(Gen, Sing)` as well as `lemma` *annus*. Its standard attribute bundle would be `[lemma=annus, feats=(Gen, Sing)]`, however flattening it would result in `[lemma=annus, feats=Gen, feats=Sing]`, in which all values are on the same level. From these full, flattened attribute bundles,

AAA extracts all non-empty subsets,[15] but only those that exceed the minimum frequency. This extraction is very prone to cause an exponential explosion, as the number of subsets is equal to $2^n - 1$, in which n is the number of attributes in the attribute bundle. Limiting this process is therefore very important. Imposing a minimum frequency on the attribute subsets (and therefore on the attributes themselves), as AAA does, already helps, but it is furthermore made sure that the algorithm does not extract subsets that contain the exact same observations. That is to say, if for example all words that are genitive happen to be singular as well, the algorithm will not extract both subsets `[feats=Gen]` and `[feats=Gen, feats=Sing]`, but only the latter. This drastically reduces the runtime, in practice. In the process of extracting attribute subsets, AAA furthermore ignores crossings and forms. Crossings, namely, tend to explode the number of subsets and are highly unlikely to be meaningfully associated to an affix in the target language; and forms (i.e. (inflected) forms in which words are encountered in the source language) have a strong tendency to associate to very long potential affixes, if not entire words, which does not benefit the desired generalization.[16]

AAA detects associated string-attribute subset pairs by means of *pmi*, or pointwise mutual information. Often used in natural language processing for finding collocations, pmi is an information theoretic measure of association, quantifying the amount of information learned about an outcome (e.g. it has rained) through observing the outcome of another random variable (e.g. the streets are wet).[17] To illustrate in terms of collocations, *Puerto* and *Rico* very often occur together in a corpus, which is reflected by a fairly high pmi between them. This means that the one word can fairly certainly be predicted when the other has been observed; when *Puerto* is observed, chances are very high that the next word is going to be *Rico*, and vice versa.

Pmi is calculated by

$$pmi(x;y) = \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

in which $p(x,y)$ is the joint probability of outcome x and outcome y occurring at the same time; $p(x)$ is the probability of outcome x; and $p(y)$ the probability of outcome y.

[15] In the case of *anni*, that would be: `[lemma=annus, feats=Gen, feats=Sing]`, `[lemma=annus, feats=Gen]`, `[lemma=annus, feats=Sing]`, `[feats=Gen, feats=Sing]`, `[lemma=annus]`, `[feats=Gen]`, and `[feats=Sing]`.

[16] Depending on the needs of the user, it can parametrically be specified what attributes need to be ignored or not. If it is so desired that crossings or forms are not ignored, they can be included in the attribute subsets.

[17] Pointwise mutual information is, then, the mutual information between two specific outcomes. One could compare this to the difference between self-information, which is about one outcome, and entropy, which is the expected self-information over all outcomes – pmi is about two specific outcomes, while the mutual information is the expected pmi. Mutual information is another name for information gain.

However, as can be seen from the formula, pmi is symmetric, that is to say, $pmi(x;y) = pmi(y;x)$. This is not ideal for our purposes, as some affixes may represent multiple distinct morpho-syntactic attribute subsets of a word; homomorphs. AAA therefore weights the pmi with the probability of the string conditioned by the attribute subset; that is, how likely it is that a string is encountered given that the attribute subset is known. This conditional probability is asymmetric. For each string-attribute subset pair AAA therefore calculates the following association value:[18]

$$A = P(string|attribute\ subset) \times pmi(string;\ attribute\ subset)$$

All string-attribute subset pairs are then sorted on this A, which is based on the probability of a word in the source language having attribute subset s and the word in the target language to which it was aligned having string, or potential affix, a. The higher A, the stronger the association between attribute subset s and potential affix a. It is then hypothesised that a may be an affix in the target language, associated to the attribute subset in the source language.

4.3 Evaluation

For the evaluation of the proposed method and developed tools, an experiment was run in which the researcher has linguistic knowledge of the source language, for which automatic parsers and taggers are available, while the researcher had no linguistic knowledge of the target language, in order to arrive at results as unbiased as possible. In order to gain insight into what kind of differences can be found with the tools, as well as what kind of differences cannot, the researcher compiled a list of morpho-syntactic hypotheses about features of the target language, and specifically differences between the source and the target language, based on the output of the tools. Meanwhile, a linguistic expert on the target language independently compiled a list of characteristic differences between the two languages that are prominent in the linguistic literature. These two lists were then cross-checked: which features that were found by the author were indeed correct features of target language's grammar; which hypotheses on features formed by the author were not correct; and which features were not found by the author that the expert listed as characteristic of the target language? These categories effectively correspond to true positives, false positives

[18] This is actually identical to a summand of the Kullback-Leibler Divergence between the probability distribution of the strings $Q(string)$ and the probability distribution of the strings conditioned by the attribute subset $P(string)$ in

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} p(x) \log_b \left(\frac{P(x)}{Q(x)} \right)$$

In this case, the Kullback-Leibler Divergence would be the information gain achieved if the conditional distribution P is used instead of the non-conditional distribution Q. This summand, then, represents the weighted part of the information gain for a specific string if the attribute subset is known.

and false negatives, respectively. True negatives cannot be considered, because they would correspond to features missed by the author that the expert had not listed as characteristic of the target language.

To this end, the language pair English-Hungarian was chosen, in which English served the role of the source language. Hungarian was chosen because the author had no linguistic knowledge of it. Dr. Lipták of Leiden University, who is a native speaker of Hungarian and a linguist specialized in Hungarian syntax, acted as the independent expert, and compiled the list of characteristic differences.

An English and a Hungarian Bible were used as corpora (see below). The English Bible was parsed and tagged in Universal Dependencies (Nivre et al. 2016) with UDPipe (Straka and Straková 2017), using the ParTUT model (Sanguinetti and Bosco 2015),[19] while the two Bibles were aligned on word level using `eflomal` (Östling and Tiedemann 2016), as discussed above.[20]

Limitations of this evaluation procedure will be discussed in Section 4.5.

4.3.1 Data

The corpus we use for the evaluation is, as mentioned, the Bible, specifically the English and the Hungarian Bible. In this Chapter it was opted not to use the Europarl corpus (Koehn 2005), in contrast to the previous two Chapters, because of specific shortcomings and complications inherent to the Europarl corpus that were noticed in the previous Chapters, such as the extensive presence of headlinese, a high average sentence length, some cases of code switching and untranslated utterances, and misaligned sentence pairs. The Bible is then convenient in that it is sufficiently large for many of our purposes, available in many languages, void of headlinese, monolingual, and implicitly parallellized through the way it is structured. However, the Bible is often archaic in its language and may not be representative of the way the language is spoken today. Hence, existing NLP tools and models may not be very suitable for Biblical language, and one must be very aware of the possibility of errors in parses and tagging. Nevertheless, we deemed the Bible to be a good choice, because of its ready availability and size, and because the only part of our approach that is dependent on language-specific models is the parser, which we did not expect to perform too poorly on the English Bible with the ParTUT model (Sanguinetti and Bosco 2015), because it was trained on texts that were collected from several legal and other formal texts, such as the Universal Declaration of Human Rights, the Europarl corpus and Wikipedia, which most closely resembled the formal, archaic, Biblical English of the available UD models.

[19] The model is available at `https://github.com/UniversalDependencies/UD_English-ParTUT`

[20] Given the nature of the task, it is impossible to measure the quality of the alignments a priori. We can therefore only report on the performance of `eflomal` on other languages in terms of alignment error rate (AER; Och and Ney 2003), which for closely related languages ranges between 7.6 and 10.6, while for less closely or unrelated languages ranges between 17.3 and 46.7 (Östling and Tiedemann 2016: Table 2).

In particular, we use the English and Hungarian parts of the Bible corpus by Christodoulopoulos and Steedman (2015),[21] which is a corpus consisting of over 100 different Bibles in `xml` format, annotated for book, chapter and verse, making alignment a straightforward task. The (standard) English Bible in the corpus is the King James Bible (KJB) from 1611, while the Hungarian Bible is the Vizsoly Bible (VB) from 1590, in a way the Hungarian equivalent of the KJB. Both Bibles are still widely considered the "classic" translation. Even though the age of the KJB would push the usefulness of the parser model to its limits, it also assured that it was, just like the VB, directly translated from Latin, Greek, Hebrew and Aramaic, allowing us to safely assume that the verses are, for the large part, syntactically comparable.[22] While a more recent version of the English Bible (the World English Bible; WEB) is included in the corpus, we chose not to use that, precisely for this reason: the WEB has been simplified more over the centuries than the KJB, diverging further from the syntactic structures in Latin, Greek, Hebrew and Aramaic and therefore in the VB.

	verses		words		
	original	shared	tokens	types	used for
EN	31102	28972	852606	12371	test & dev.
HU	31298		683690	61036	test
NL	29098		838324	21605	dev.
CS	31102		680938	39648	dev.

Table 4.2: The number of verses in the original Bibles, as well as the number of verses shared between the four versions. The number of words in the shared verses in terms of tokens and types is also listed per language. Hungarian is used for testing (i.e. running the experiment), Dutch and Czech only for the development of the tools, and English is used for both testing and development.

During development of the tools we used the Czech and the Dutch Bibles as well, and while the Czech version is from 1380 (so likely to be directly translated from Latin, Greek, Hebrew and Aramaic, too), the age of the Dutch Bible is not listed by Christodoulopoulos and Steedman (2015), however it seems to be from the late 1900s, probably 1987.[23] Because we wanted to make sure that the results are comparable between language pairs, we only used verses that

[21] The corpus is available on `https://github.com/christos-c/bible-corpus`

[22] Kroon et al.'s (2019) filter for syntactic comparability (see Chapter 2) was not deployed in this approach, because all filters that were developed build on existing NLP tools for both languages, except for the sentence-length filter. Seeing as the assumption that no linguistic knowledge or tools were available for the target language is an important aspect in this research, and that the sentence-length filter did not yield satisfying results, it was opted not to use a filter in this research.

[23] The age of the Dutch Bible corroborated our assumption about syntactic comparability between Bibles of similar ages. Although not quantified, the word alignments between English,

are present in all four versions of the Bible (i.e. only verses with IDs that were present in all four parts of the Bible corpus), which resulted in 28972 verses, with 852606 tokens in English and 683690 tokens in Hungarian. A full overview of the number of verses, tokens and types can be found in Table 4.2.

4.4 Results

After analysing the data and the output of the files, a list of 43 morpho-syntactic hypotheses about features of Hungarian was compiled by the author based on the output of the tools – a summary can be found in Table 4.3. Meanwhile, Dr. Lipták independently compiled a list of 32 differences between English and Hungarian that are characteristic of Hungarian and are prominent in the linguistic literature on Hungarian (henceforth the AL list). In this section we will discuss all of the (correct or wrong) hypotheses and morpho-syntactic features in detail.

In general it was observed that the majority of the hypotheses on morpho-syntactic features were correct, with 37 out of the 43 being a feature of Hungarian grammar. Two hypotheses were only half correct, painting an incomplete picture or overgeneralizing slightly. Another two raised further questions about Hungarian, about which Dr. Lipták was unsure whether they are or are not features of the Hungarian language. Only two hypotheses were actually incorrect.

Furthermore, out of the list of 32 prominent and characteristic differences on the AL list, eight were correctly discovered, while one hypothesis was contradicting a difference on the AL list. The rest –23– were missed, and are listed in Table 4.4. Each missed difference will be discussed in full in Subsections 4.4.1 to 4.4.5 below. In summary, these differences were mainly missed due to the information structure of a sentence not being annotated, or to the genre of our corpus.

In order to illustrate the process of forming hypotheses, we will begin this section by taking the hypotheses and missed differences concerning articles and demonstratives as an example, and discussing them in more detail, explaining our reasoning behind the interpretation of the data in Subsection 4.4.1.

We shall continue the section by discussing the remaining hypotheses and missed morpho-syntactic differences briefly, divided over three subsections: hypotheses and differences concerning the nominal domain (Subsection 4.4.2),

Hungarian and Czech were much better than the alignments between English and Dutch, reflected in the number of wrongly aligned words, implausible crossings and the number of unaligned words between English and Dutch. It is also reflected in the sizes of several output files, which were much larger for English-Dutch than for the other language pairs. This shows that the Dutch output can be summarized and compressed much less well; the entropy of the Dutch aligned data is much higher than for the other language pairs. We can probably conclude that the syntactic structures between the KJB and the Dutch Bible are therefore much less similar than between the KJB and VB, leading to wrong alignments, zero alignments, crossings and noisy data in general.

No.	Hypothesis	Correct?
	Nominal domain	
H1	articles	+
H2	articles come before NP	+
H3	articles do not inflect for case	+
H4	only definite articles	−
H5	licensing difference for articles	+
H6	nom. mods before and after NP	+
H7	case, expressed on noun	+
H8	locative cases	+
H9	accusative on nouns: -t	+
H10	accusative on pronouns: -t	+
H11	vowel harmony: front-back	+
H12	agglutinative	+
H13	no gender anywhere	+
H14	possessives optional: suffixes	+
H15	possessives prenominal	+
H16	3sg and 3pl same possessive	+
H17	adjectives both before and after noun	+/−
	Verbs and constituent order	
H18	free(r) word order	+
H19	transitive and intransitive verbs same position	+
H20	SV word order	+
H21	VO word order	+
H22	SVO word order	+
H23	relative order of constituents mostly same as EN	+
H24	adverbials mostly postverbal	+
H25	adv. clauses: same position EN	+
H26	adv. clauses: stricter word order than main clause	?
H27	pronouns positionally the same as nouns	+
H28	subject pro-drop	+
H29	verbs inflect for all persons: present	+
H30	verbs inflect for all persons: past	+
H31	much fewer auxiliary verb (temporal, aspectual, modal)	+
H32	synthetic passive	+
H33	no infinitival marker to	+
H34	adverbial negation	+
H35	negation comes before negated	+
H36	adverbs precede verb	+
H37	fewer pro-adverbs used	?
H38	zero copula	+/−
H39	copulae in general before predicate	+
	Other	
H40	both prepositions and postpositions	−
H41	adpositions declined for person	+
H42	coordinating conjunctions before conjunct	+
H43	subordinating conjunctions before conjunct	+

Table 4.3: A summary of the hypotheses formed about morpho-syntactic features in Hungarian, and whether they are correct or not. A plus sign means the hypothesis was correct; a minus that it was incorrect; and a question mark that the hypothesis has not yet been confirmed nor rejected.

No.	Difference
	Nominal domain
M1	demonstratives inflect for case
M2	numerals select singular noun
M3	demonstrative and article must co-occur
	Verbs and constituent order
M4	object pro-drop (singular)
M5	any number of constituents before verb
M6	wh-phrase: before finite verb in main clause
M7	wh-phrase: before finite verb in embedded clause
M8	wh-phrase: more than one before verb possible
M9	yes/no question: same word order as declarative
M10	embedded yes/no question: same word order + -e
M11	only N phrase before finite verb
M12	verbs agree with definiteness of object
M13	infinitive sometimes agree with subject
M14	singular agreement with subject with numeral
M15	verbal particle: can be before verb
M16	verbal particle: if after, free word order
M17	verbal particle: can be before auxiliary
M18	verbal particle: if before, can be reduplicated
M19	verbal particle: only before, if not only N phrase before
M20	verbal particle: idem, if not wh-phrase before
M21	verbal particle: idem, if not negation before
M22	verbal particle: idem, if not progressive aspect
	Other
M23	negative concord language

Table 4.4: A summary of the missed differences between Hungarian and English on the list compiled by Dr. Lipták.

those concerning verbs and constituent order (Subsection 4.4.3), and other hypotheses (Subsection 4.4.4).

Additionally, it was observed that our method was successful in detecting morpho-phonological features of Hungarian, including hypotheses about specific case endings, such as *-t* for accusative. Since automatic affix detection is an important goal in the field of comparative syntax, we will conclude the section by discussing all hypotheses and differences pertaining to affixes in subsection 4.4.5.

During the discussion of hypotheses and differences, we shall refer to them by their code as found in Tables 4.3 and 4.4 for easy reference.

4.4.1 Articles and demonstratives

In total five hypotheses about morpho-syntactic features of Hungarian pertain to articles and demonstratives (**H1**–**5**), four of which proved to be correct, while one was false. Meanwhile, one morpho-syntactic difference on the AL list pertains to demonstratives, which was missed (**M1**).

The hypothesis that Hungarian has articles (**H1**) turned out to be correct. To illustrate how this hypothesis was formed, Figure 4.5 shows (a fraction of) the output of the DGAE for words tagged as determiners (DET) in English – a tag that includes articles in UD – that have a `det` (determiner) relation to their head in the dependency tree. The output shows us that there are 80341 instances of such words in the English Bible. Under `form` it lists that 59116 of these words were an instance of the word *the*, and 7660 and 1582 were *a* and *an*, respectively, amounting to 9242 instances of `('DET', 'det')` having the lemma *a*. It also shows further breakdowns of features, such as that 68976 instances of `('DET', 'det')` were tagged as `'PronType=Art'` by UD – as having the pronoun type 'article'.

Importantly, under `translation`, it can be seen that 19500 instances of `('DET', 'det')` did not receive an alignment to a Hungarian word. Though a large number, it is significantly less than the number of words tagged as an article in English. This in turn means that if it were only articles that did not receive an alignment, then still $49476 (= 68976 - 19500)$ articles were aligned to a Hungarian word, amounting to at least 71.7% of all English articles having an alignment in Hungarian. This led to the correct hypothesis **H1**, that Hungarian has articles.

The DGAE output in Figure 4.5 also shows frequency breakdowns of crossings. Under ancestor crossings, it can be found that 40083 instances of `('DET', 'det')` do not cross with their ancestors, which suggests that the relative order of a determiner and its ancestors in English is the same as the relative order of the aligned-to Hungarian words.[24] Note that there can only be a crossing if a word has received an alignment: if it was not aligned, DGAE will

[24] Remember that a crossing between word i and its ancestor a are only considered – and taken up in DGAE's output – if i does not cross with a's direct daughter node. See Section 4.2.2.

('DET', 'det'): 80341	
form	('the', 59116), ('a', 7660), ('all', 3492), ('this', 1874), ('an', 1582), ...
lemma	('the', 59116), ('a', 9242), ('all', 3492), ('this', 2651), ('that', 1199), ...
pos	('DET', 80341)
deprel	('det', 80341)
feats	('PronType=Art', 68976), ('Definite=Def', 59345), ('Number=Sing', 13528), ('Definite=Ind', 9631), ('PronType=Dem', 4054), ...
translation	('a', 28600), (None, 19500), ('az', 17116), ('minden', 1926), ('e', 1309), ...
ancestor cross	('det^{0}', 40083), (None, 19500), ('det^{0}-nmod^{-1}', 2292), ('obl^{-2}-det^{0}-nmod^{-1}', 1889), ('nmod^{-2}-det^{0}-nmod^{-1}', 1480), ...
descendant cross	('det^{0}', 60833), (None, 19500)
sibling cross	('det^{0}', 50210), (None, 19500), ('case^{1*}-det^{0}-obl^{-1}', 1124), ('case^{1*}-det^{0}-nmod^{-1}', 594), ('det^{0}-obl^{-1}-nmod^{1*}', 356), ...
children	(None, 80264), ('that|fixed', 19), ('one|nummod', 13), ('be|cop', 7), ('of|case', 5), ...
parent	('NOUN|nmod', 19983), ('NOUN|obl', 18717), ('NOUN|obj', 12825), ('NOUN|nsubj', 9525), ('NOUN|root', 4178), ...

Figure 4.5: An example of the DGAE output. Displayed is a fraction of the results for English words tagged as determiners and that have a `det` relation to their head.

count that instance towards a `None` crossing. This allows us to quickly see that 65.9%($= \frac{40083}{80341-19500}$) of all aligned English determiners are on the same (i.e. left) side of their head as the Hungarian words they were aligned to. Similar to the reasoning that led to **H1**, we can see that if it were only articles that crossed with their ancestors, then still 48218($= 68976-(80341-40083-19500)$) articles showed up on the same side of their head as in English. This amounts to at least 69.9% of all Hungarian articles occurring on the left of their head, leading to hypothesis **H2** – Hungarian articles come before the NP – which turned out to be correct.

Furthermore, **H3** was formed based on the fact that there would be only four forms, which suggest a common stem *a-* under `translation`, of which only two are listed in Figure 4.5: *a*, *az*, *annak* and *ama*. The latter two, however, were much less frequent, with 361 and 224 occurrences, respectively, suggesting perhaps noise or another lemma. If articles were marked for case, it could be expected that there would be a higher entropy among the aligned-to Hungarian words, especially those that suggest a common stem. During further exploration of the GTI (output not shown) it was indeed noticed that there are only two Hungarian words clearly associated to definite articles (*a* and *az*) and only three Hungarian words clearly associated to indefinite articles (*a*, *az* and *egy*). Additionally, there was no pattern noticeable in the form of the article and the grammatical function (i.e. dependency relation) of the head of the English article, which one would expect if case is marked on articles. This led to the correct hypothesis that articles do not inflect for case. The fact that demonstratives, also tagged as `DET`, do inflect for case (**M1**)[25] can be found with the help of GTI, which shows that English demonstratives are aligned to a group of Hungarian words sharing a common stem, while the different endings show a noticeable correlation between the dependency relation of the determiner's parent node in English, suggesting that the determiners are inflected for case.

The output in Figure 4.5 shows that 9631 English determiners are tagged as having the `'Definite=Ind'` feature, meaning that they are indefinite, specifically indefinite articles. Notice that 9631 is more than the number of occurrences of the lemma *a*; this is because the label indefinite also includes the word *another*, which is not analysed as having the lemma *a*. However, with 9631 there is no clear candidate for a translation among the aligned Hungarian words; Hungarian aligned-to words either occur much more often, or much less often than indefinite articles in English. This immediately prompted further investigation with GTI, with which it was possible to observe that out of the 9631 English words tagged as an indefinite article, 4214 were aligned, constituting only 43.8%. Furthermore, as also mentioned above, there were only three Hungarian words clearly associated to indefinite articles: *a*, *az* and *egy*. However, we had already seen that *a* and *az* probably correspond to definite articles, and *egy* is rather infrequent with only 1146 occurrences, i.e. 11.9% of all English indefinite articles. It was therefore concluded that *egy* was either noise or the

[25] It turns out that *annak* is actually an inflected form of the demonstrative *az* 'that'.

cardinal number *one*, and that Hungarian only has definite articles *a* and *az* (**H4**). The fact that *a* and *az* are so often aligned to indefinite articles additionally led to the hypothesis that there is a usage difference between English and Hungarian articles (**H5**).

While **H4** proved to be incorrect with Hungarian having both definite (*a* and *az*) and indefinite articles (*egy*, which also serves the purpose of the cardinal number *one*), **H5** was correct: Hungarian does not use indefinite articles in existential and 'have' constructions and before predicate nouns, and indefinite articles can be dropped before subjects, objects or adverbials directly preceding the verb, while in all these cases an indefinite article must be present in English (Rounds 2009: 83). Although we believe that the data do point towards hypotheses **H4** and **H5** despite one of them having been proved incorrect, we also believe that the important conclusion of this showcase is that a linguist can start asking basic questions about characteristic morpho-syntactic features of a language – such as whether a language has both definite and indefinite articles – and explore the data with the help of our tools to form meaningful hypotheses on them.

For the sake of brevity, the remaining hypotheses and differences will be discussed in somewhat less detail. While **H1**–**5** served as an example to illustrate how the output of our tools are interpreted, the remaining hypotheses and missed differences will showcase the wide range of morpho-syntactic domains our tools can detect differences in.

4.4.2 Other hypotheses concerning the nominal domain

Additionally, 12 other hypotheses were formed pertaining to the nominal domain, 11 of which proved to be correct, while one was only half correct. Meanwhile, four morpho-syntactic differences on the AL list pertained to the nominal domain, of which two were discovered correctly, while the other two were missed.

It was observed in the DGAE output that an English word with a UD `nmod` relation to its heads appears without crossing it in 52.3% of the cases. It was specifically observed that in about half of the cases an `nmod` that occurs to the right of its head in English appears to the left of its head in Hungarian. Seeing as the `nmod` relation is used to denote the relationship between nominal dependents and another noun or noun phrase, corresponding functionally to an attribute (i.e. a nominal modifier; in the case of English, a prepositional complement) or a genitive complement, it was hypothesized that attributes and genitives can come both before and after their head in Hungarian (**H6**), although it is hard to identify from the output when it comes before and when it comes after its head.

It was correctly hypothesized that Hungarian has grammatical case, marked on the head noun (**H7**). This was most prominently suggested by the fact that 55.6% of all English prepositions did not have an alignment in Hungarian, as found with DGAE. This characteristic feature of Hungarian was also on the

AL list.

Further investigation in GTI showed that it was most frequently the prepositions *of*, *in*, *unto*, *to*, *with*, *from*, *upon*, *by*, *into*, *at* and *on* that did not receive an alignment, leading to hypothesis **H8**, about the presence of a genitive, inessive, dative,[26] allative, instrumental or sociative, elative or ablative, superessive, adessive, illative, and perhaps a temporal case in Hungarian – or at least something similar. Indeed, Hungarian does have all of these grammatical cases, except for the genitive, however the genitive is expressed by either a nominative noun that precedes its selecting head or a dative that follows it; in either way, possession is not expressed with a preposition.

Touching briefly on **H9** and **H10** – all hypotheses on morphology will be discussed in more detail in Subsection 4.4.5 – it was correctly hypothesized that Hungarian also has an accusative case ending in *-t*, which is marked on nouns and pronouns. Hypotheses **H9** and **H10** were formed by interpreting the results of AAA and GTI, in which the ending *-t* was highly associated to Hungarian words that were aligned to English nouns and pronouns that have an `obj` relation to their head, indicating a direct object relation. Importantly, this *-t* does not appear on Hungarian words aligned to subjects of intransitive verbs, indicating that Hungarian is not an ergative language: this conclusion allows us to form hypotheses on subjects and objects in Hungarian in Subsection 4.4.3. The discovery of the ending *-t*, which led to **H9** and **H10**, will be discussed in more detail below.

Gender was correctly hypothesized to be completely absent in Hungarian (**H13**). It was observed in DGAE and GTI that English lemmas *he* and *she* often received the same translations in Hungarian: *ő* or a suffixed form of that. There were furthermore no indications of gender being present on nouns, as there did not seem to be specific sets of affixes only occurring with one group of nouns, and not another – nor were there any attribute bundles found with AAA that are associated to two distinct endings. In fact, it may be enough to notice that there is no gender in pronouns in order to conclude that there is no gender in nouns: Greenberg's linguistic universal number 43 states that if a language has gender categories in the noun, it has gender categories in the pronoun (Greenberg 1963), although the number of languages Greenberg studies is limited. The absence of gender in Hungarian was also listed on the AL list.

Possessive pronouns were found not to be aligned to a Hungarian word in 34.9% of the cases. Based on alternations observed in GTI between suffixed and unsuffixed nouns, which seemed to be correlated to the English noun having a possessive pronoun as a child in the dependency tree, it was tentatively concluded that Hungarian suffixes the possessed noun with a personal possessive ending, making possessive pronouns redundant (**H14**). Indeed, Hungarian only uses possessive pronouns for emphasis or contrast (Rounds 2009: 140). It

[26] Bearing in mind that our corpus is the Bible – the preposition *unto* is often used as a dative construction in the KJB, e.g. Genesis 3:2: "And the woman said **unto** the serpent, We may eat of the fruit of the trees of the garden:"

was furthermore found in the data that a possessive pronoun, whenever it is expressed, precedes the noun, leading to the correct hypothesis **H15**. Additionally it was observed in the data and correctly hypothesized (**H16**) that there is no distinction between singular and plural third person possessive pronouns in Hungarian.

Hungarian words aligned to English adjectives were found not to cross with their head noun in 65.2% of occurrences and not to cross with any child nodes, i.e. modifiers, in an overwhelming 98% of occurrences. It was subsequently hypothesized that adjectives are mostly prenominal but can occur postnominally, but that the structure of the adjective phrase is the same as in English (**H17**). The latter part of the hypothesis was formed, seeing as the word order inside the adjective phrase was mostly the same, and there were hardly any crossings observed among the children of adjectives. However, while the latter part of **H17** is correct, Hungarian adjectives can only occur prenominally. Later inspection of the data showed that many adjectives were wrongly tagged as such, with many occurrences of *thy*, *unto* and *Lord* receiving the tag ADJ. The personal pronoun *I* was often interpreted by the tagger as the Roman numeral one, which was then interpreted as *first* and tagged as an adjective. The noisy nature of the ADJs highly influenced the numbers and consequently led to a partly wrong conclusion; however, we believe that thorough inspection, especially in the GTI output, could have laid bare this tagging error.

The fact that a noun phrase containing a numeral has a singular head noun in Hungarian (**M2**) was missed. Currently, there are a few complications in the data processing and output formatting that would prevent a linguist from forming a hypothesis about the grammatical number of a head noun in a noun phrase containing a numeral, even when they are specifically researching this question. Due to the way the data are represented in the dataframe, numerals can only "see" the POS of their parent nouns and what dependency relation they have to it – and not the Hungarian word their parent noun is aligned to, which is necessary to be able to see that it is singular.[27] Due to the way the output of our tools is formatted (and the way it suppresses infrequent attributes), numerals cannot be accessed and easily investigated as children of nouns, as they are so infrequent[28] that they are washed away among the much more frequent determiners or adjectives (or even `None`), making them "invisible". These complications led to **M2** currently being missed, however we believe that if a linguist could narrow down on numerals as children of nouns more easily, our tools would work well and provide linguists the information needed to form meaningful hypotheses about the grammatical number of head nouns in a noun phrase containing a numeral.

Lastly, **M3** was not found: a demonstrative and a definite article necessarily co-occur in a Hungarian noun phrase (e.g. *ez a hely* lit. 'this the place'). In our current setup it is not possible to find this: when only the demonstrative and

[27] In order to be able to notice that a word is singular, the linguist would first need to form a hypothesis about nominal paradigms.

[28] Only 6073 out of the 737319 tokens in the English Bible were tagged as a numeral: 0.8%.

the noun are aligned to an English word[29] we cannot discover in our dataframe that *a* or *az* was there in the Hungarian sentence. Adding concordances or adjacent words of aligned-to words in the target language could perhaps allow the linguist to find features such as these.

4.4.3 Verbs and constituent order

22 hypotheses were formed that concern the verbal domain or constituent order. Of these hypotheses, 19 were correct, while one was an overgeneralization. Two hypotheses have not yet been confirmed or rejected, as they require further research. Dr. Lipták compiled 25 differences between English and Hungarian that pertain to the verbal domain or the constituent order, of which 6 were correctly discovered, but 19 missed.

Unlike Japanese or Bantu languages, English does not encode the information structure of a sentence with morphemes. The result is that the dependency tree as produced by UDPipe is not annotated for the information structure of the sentence in any way, and that information-structural knowledge can therefore not be mapped onto Hungarian sentences. However, as many languages rely much more heavily on word order to encode the information structure than English does, a linguistic user of our tools can venture the hypothesis that the target language does, too. In that case, investigating crossings can provide valuable insights into the freeness of word order and consequently the information structure of the target language.

It was observed that in 39.1% of all occurrences of a verb there was a crossing between it and one of its arguments (i.e. one of its descendant nodes, which include complements, auxiliaries and adverbs), indicating a different word order than in English, while in 60.9% there was no crossing. It was thus hypothesized that Hungarian word order is much freer than in English (**H18**), because these statistics suggest that Hungarian does not systematically have the same or a different word order than English. These crossing frequencies are similar for both transitive and intransitive verbs, leading to the correct hypothesis that both types of verbs behave similarly in this respect (**H19**).[30]

Subjects were hypothesized to precede the verb in general (**H20**), observing that Hungarian words aligned to English subject nouns occur on the same side of the verb as English subject nouns in 71.8% of occurrences, which is before the verb. Similar numbers were found for object nouns, which come on the same side of the verb, i.e. after it, in 77.3% of occurrences, leading to the hypothesis that Hungarian has a standard VO order (**H21**). Together,

[29] This is what happens. The aligner learns that *a(z)* is to be aligned to *the*, and that *ez* is to be aligned to *this*, however *the* is absent. Indeed, *ez* and *a(z)* often occur together in Hungarian, but only when the demonstrative modifies a noun; if it is used predicatively, the article is absent in Hungarian, too. The aligner therefore does not learn to align *this* to both *ez* and *a(z)* at the same time, but instead leaves the article unaligned.

[30] We don't know whether transitive and intransitive verbs take up different sentence positions in any language – but at least we know it does not make a difference in Hungarian.

these numbers led to the hypothesis that Hungarian is primarily SVO (**H22**). However, while it is (or may be) quantitatively correct that SVO is the most prominent constituent order in Hungarian, all orders can occur: in about one in four sentences, Hungarian subjects and objects occur on the other side of the verb than in English, confirming the hypothesis of a freer word order (**H18**) and that of information structure being encoded through syntactic movement, as well.

Indeed, Dr. Lipták later confirmed that word order in Hungarian is in principle free and wholly determined by the information structure of a sentence. Hungarian word order is characterized by four sentence positions: Topic–Preverb–Verb–Rest. While the topic position may be empty, it is usually filled by the subject (hence the SV order) but can be filled by other constituents, too. The preverbal position neutrally contains a verbal complement, an adverbial or a coverb, prefixed to the verb, but can also contain the focus of the sentence, such as wh-words, negation or otherwise stressed phrases. Importantly, whenever the preverbal position is taken up by the focus, any other material that would have gone into the preverbal position is moved after the verb (to Rest), creating the possibility for all word orders to arise (Rounds 2009: 254; cf. also Kiss 2002).

It was furthermore hypothesized that the relative order of subjects, objects and other constituents is predominantly the same as in English (**H23**). This conclusion proved to be correct, and was made based on the fact that object nouns rarely cross (13.1%) with their sisters in the dependency tree. This means that the position of subjects, nominal adverbials (such as locative, temporal or directional (possibly prepositional) complements, excluding adverbial clauses; i.e. a noun or pronoun receiving the `obl` relation to its head), adverbs and auxiliaries relative to the object is the same in English as it is Hungarian in 86.9% of the cases. That is, if a Hungarian subject is present in the sentence (or rather, has received an alignment to an English subject) it will appear on the left of the object in the majority of the cases, while nominal adverbials are on its right, whenever they are there. However, if the order is different, then it is mostly the adverbial that comes to the left of the object, while the subject rarely comes to the right. This could be read as that whenever a subject is overt in a Hungarian sentence, it will often be the topic, and therefore on the left of the object. Meanwhile, nominal adverbials are usually to the right of the object, but can be topicalized, ending up on the left of the object, as well. And, if a subject and an object are both present in a sentence where an adverbial is fronted, the subject would still appear on the left of the object in the majority of the cases.

Adverbials that are tagged as nouns or pronouns by UDPipe with an `obl` relation to their heads (such as locative, temporal or directional complements, excluding adverbial clauses) were indeed found to be mostly postverbal, leading to correct hypothesis **H24**. 20.8% of English adverbials cause a crossing with its ancestor verb, meaning that about one in five Hungarian adverbials are on the other side of the verb compared to English. This is further supported by the

fact that English adverbials preceding the verb have their translation appear to the right of the verb in Hungarian in 32.9% of the cases, while English adverbials following the verb have their translation appear on the other side of the verb in 15.5% of the cases: Hungarian adverbials are therefore more likely to follow the verb.

Hungarian adverbial clauses, on the other hand, were correctly hypothesized to be in the same position as in English (**H25**). DGAE shows that an overwhelming 93.6% of all adverbial clauses in Hungarian appear on the same side of the main verb as in English. It is not entirely clear why, but it could be the result of translation, where word order is kept constant throughout translations, or it could show a correlation between the weight of an adverbial clause and its position.

We furthermore observed that the verb of an adverbial clause crosses with one of its arguments in 63.7% of all adverbial clauses (i.e. the argument appears on the other side of the verb compared to English), less than the amount of verb-argument crossings in a main clause. We interpreted this as perhaps being the result of a slightly less free word order in embedded clauses, leading to hypothesis **H26**. This stricter word order in adverbial clauses could again be a statistical anomaly, with Dr. Lipták not being aware of any such restriction on the order of words in adverbial clauses, but it is nonetheless a valid reason to investigate Hungarian adverbial clauses in more detail. As of yet, the correctness of **H26** is unknown.

Pronouns were correctly hypothesized to behave similarly to nouns, positionally (**H27**). Pronoun subjects, objects and adverbials appear in the same positions as nouns do, showing similar crossing statistics. This means that, in general, pronoun objects appear after the verb – and are not proclitics as in French – and that pronoun subjects and adverbials appear before and after the verb, respectively.

The AL list mentions that Hungarian allows for the dropping of subject pronouns as well as singular object pronouns. While it was indeed correctly hypothesized that Hungarian has subject pro-drop (**H28**) – by noting that English subject pronouns were not aligned to a Hungarian word in 47.6% of the cases – a hypothesis was not formed on pro-drop including singular objects (**M4**). Nonetheless, Hungarian object pro-drop could have been found, by observing in DGAE and GTI that English singular object pronouns often have no translation either.

More than half, 58.5%, of English auxiliary and modal verbs (POS tag AUX in UD) were not aligned to a Hungarian word, leading to the correct hypothesis that auxiliaries are much less frequent in Hungarian and that Hungarian is less analytical than English (**H31**). Indeed, Hungarian has a synthetic past tense, without a *have*-like auxiliary, does not have an analytical continuous, and can express the future with a present tense, similar to Finnish and Dutch, for example. However, the future can also be expressed with the auxiliary *fog*, but this was not found in DGAE or AAA, with no form of *fog* being among the top-

20 most aligned-to Hungarian words for English auxiliaries.[31] As for modals, Hungarian expresses *can* or *may* with the suffix *-hat/-het*, also adding to the unaligned auxiliaries in English.

The fact that Hungarian does not have passivization nor a passive auxiliary was on the AL list. It was found in DGAE that passive auxiliaries, which are tagged with a distinct relation to their head verb in UD, are not aligned in 64.6% of the cases, with other translations seeming to be noise,[32] suggesting that there is indeed no passive auxiliary. This discovery was however generalized to the hypothesis that Hungarian has a synthetic passive voice (**H32**). Dr. Lipták pointed out that this is not true for modern Hungarian, where the third person plural is used instead of passives. However, more archaic Hungarian, such as in the Bible, does still have passivization to some degree, making **H32** correct for this specific corpus. It must be noted, though, that it would be very difficult, if not impossible, to detect whether a language has passivization with our tools. Although AAA does retrieve (with rather low association scores) two suffixes that, to the best of our knowledge, are unique for the archaic passive forms in Hungarian (*-tik* and *-tott*), the attribute bundles with which they are associated do not contain any information about it being a passive, and only tell us that the endings are associated with an English past participle.[33] DGAE and GTI also do not grant good insight in the presence of passives in Hungarian.

It was observed in DGAE and GTI that 50.2% of all infinitival markers *to* are not aligned to a Hungarian word, with other alignments mostly containing *hogy* 'that, (in order) to' (29.5%). This led to hypothesis **H33**: Hungarian does not have an infinitival marker such as English *to*, which turned out to be correct.

English *not* was aligned in 98.1% of its occurrences, most frequently to *nem* (62.4%), *ne* (21.4%), *sem* (4.5%), *meg* (2.1%) and <,> (1.7%). It was consequently concluded that negation in Hungarian is not done with verbal morphology on the main verb. It was therefore hypothesized that Hungarian negation is done with adverbs or particles (**H34**), which turned out to be correct. The possibility of Hungarian negation being expressed through a negative auxiliary verb was ruled out, because one would expect for *not* to be aligned to more forms containing the same stem, as well as a less skewed distribution over the aligned forms corresponding to multiple grammatical persons, since we had seen in **H29** (discussed in 4.4.5) that Hungarian verbs inflect for all persons.

[31] In fact, the top-20 most aligned-to Hungarian words for auxiliaries contained many non-auxiliaries, indicating noise – with the most striking being that the comma (i.e. <,>) was the most common Hungarian alignment among English auxiliaries, but only amounted to 6% of all aligned cases. This even more strongly corroborates the conclusion of H31.

[32] Similar to non-passive auxiliaries, the most common Hungarian alignment was the comma (i.e. <,>), but it only amounted to 2.1% of all aligned cases. Other alignments (including articles, conjunctions and more punctuation) were even less frequent, suggesting that the bulk of the Hungarian alignments of English passive auxiliaries are noise.

[33] In fact, nine other endings are associated to past participles, and although these endings could be passive suffixes, they can also be active past tenses.

It was also correctly hypothesized that negation precedes the negated, specifically the verb (**H35**), by observing in GTI that *not* does not cross with its ancestor in 71.5% of its occurrences. While negation never comes left adjacent of the finite verb or auxiliary in English, it does usually come left of the main verb,[34] which is the root of the dependency tree. It can therefore be deduced from the absence of a crossing that negation comes before the verb in Hungarian, and precedes any negated word or phrase. The positioning of negation is on the AL list as well, specifically pointing out that negation is left adjacent to the finite verb or auxiliary, contrary to English.

The hypothesis that other adverbs also precede the main verb in Hungarian (**H36**), turned out to be correct, too. This was tentatively concluded based on the fact that adverbs preceding their heads in English are much more common (73.6% of all adverbs in English come in a position before their heads), while it was observed that whenever an adverb precedes its head in English, its alignment would be to the right of the head in Hungarian in 14.6% of the cases (i.e. 10.7% of all adverbs) and whenever an adverb follows its head in English, its alignment would be on the left of the head in Hungarian in 31.3% of the cases (i.e. 8.3% of all adverbs). These relative frequencies, as illustrated in Table 4.6, then show that adverbs are more likely to precede their head, primarily main verbs, in Hungarian.

		Hungarian		
		left	right	total
English	left	62.9%	10.7%	73.6%
	right	8.3%	18.1%	26.4%
	total	71.1%	28.9%	100.0%

Table 4.6: The distribution of the relative positions of adverbs in English and in Hungarian, as deduced from crossing frequencies. The conclusion is that Hungarian adverbs tend to come to the left of their heads (**H36**).

It was hypothesized that pro-adverbs, such as *then* and *so*, are not as abundant in Hungarian (**H37**). This has not yet been confirmed or rejected; neither Dr. Lipták nor a grammar of Hungarian could provide an answer to the matter. It came to be hypothesized as it was observed in DGAE that almost one in five adverbs did not receive an alignment, and further inspection in GTI showed that it was mostly due to such small adverbs.

It was noted that English copulae were not aligned to a Hungarian word in 43.8% of occurrences. This led to hypothesis **H38**: that Hungarian allows for zero copula. This is, however, a slight overgeneralization, as Hungarian only allows for the dropping of the third person forms of *van* 'to be', and only if

[34] Except for archaic constructions such as *I know not*, which, admittedly, are present in the Bible.

the predicate is a noun or an adjective. This restriction on zero copulae in Hungarian could have be found in GTI, though.

When copulae are overt in Hungarian (i.e. when an English word tagged as copula is aligned to a Hungarian word), however, they were found to cross in 31.1% of the cases. This led to the hypothesis that Hungarian copulae can come both after or before the predicate, though in general before (**H39**). It could not be found what causes these crossings, though it turns out that copulae come before the predicate if the preverbal position is taken up by e.g. negation (Rounds 2009: 254).

Several morpho-syntactic features or differences between Hungarian and English on the AL list were not found. It was not found that in Hungarian any number of constituents can come before the verb (**M5**). This is simply because our tools do not collect statistics on the number of constituents preceding or following the verb or any other head, although this could easily be implemented.

To continue, three differences concerned question phrases (**M6**–**8**), or wh-phrases, and another two concerned yes-no questions (**M9**–**10**). These differences were unfortunately not found because wh-words are not separately tagged in UD; while interrogative personal pronouns do receive a `feats` tag that distinguish them from other pronouns, other wh-words, such as *where*, *whence* and *how*, do not. It is therefore difficult to detect any morpho-syntactic features pertaining to question phrases when using UD tagging. Furthermore, sentences are not individually tagged for sentence function (declarative, interrogative, exclamative or imperative), which would be very beneficial for the detection of morpho-syntactic features pertaining to yes-no questions. Finally, questions are not all too frequent in the Bible, making the corpus somewhat unsuitable for the detection of differences with regards to questions.

The difference that *only N* phrases must be left adjacent to the finite verb or auxiliary in all types of clauses was not found (**M11**). This is due to the fact that the difference is rather fine-grained and can easily be missed if one is not looking for this difference in particular. Furthermore, the word *only* only occurs 255 times in the KJB (or at least, the section that we used), making *only N* phrases highly infrequent, and even so infrequent that they do not show up in the GTI, which limits its output. Though, even if the construction was more frequent, our tools do not automatically correlate the position of the noun in Hungarian (in terms of crossings) to the fact that it contains an aligned instance of *only*, making it difficult to spot this pattern.

Another important difference, **M12**, was also not found: Hungarian present and past tense finite verbs show agreement with the definiteness of the object; verbal paradigms depend on whether the object is definite or indefinite. Seeing as nouns are not tagged for definiteness (although articles are), it is very hard to detect a pattern in GTI, and even impossible for AAA to correctly associate specific verbal suffixes to the definiteness of the object.

It was also missed that infinitives in Hungarian sometimes agree with subjects (**M13**), which happens when an infinitive is used with an impersonal

verb such as *kell* 'must'.[35] This cannot be found because this peculiarity of Hungarian is solely dependent on a Hungarian context and cannot be clearly related to an English construction. While one may be able to find that *kell* means 'must' in GTI, it cannot be found what forms the English infinitives are aligned to in those constructions.

Furthermore, verbs show singular agreement with a noun phrase that contains a numeral (**M14**).[36] This cannot be found directly, because it is not possible in the way data are represented to see grandchildren nodes, i.e. daughter nodes of daughter nodes in the dependency tree in English, which is necessary in order to be able to see that the verb's subject (which is the verb's daughter) is modified by a numeral, represented as the subject's daughter. However, even if our tools returned statistics on grandchildren, singular agreement with a subject modified by a numeral can only be found if one already has a hypothesis about the paradigm of the verb in the target language, as otherwise it will be very challenging to notice a pattern.

Lastly on verbs, there are several differences between English and Hungarian pertaining to verbal particles on the AL list – eight, in fact. Verbal particles in Hungarian, also called preverbs in Hungarian linguistics, comprise resultative, terminative and locative elements that telicise the verb (see Ladányi 2015 for a recent overview), while Dr. Lipták took English verbal particles to be particles that associate with phrasal verbs, such as *away*, *down*, *forth* and *up*, whose functions are in many cases similar to those of Hungarian preverbs. Hungarian preverbs can come both before and after their verb. Dr. Lipták lists the following differences:

M15 A verbal particle can be left adjacent to its verb in Hungarian, while in English it cannot.

M16 If a verbal particle follows its verb in Hungarian, it can show up in any position between the constituents following the verb, while in English it can only come in fixed positions.

M17 A verbal particle can occur before an auxiliary in Hungarian, while in English it cannot. However, not all auxiliaries allow for this.

M18 A verbal particle that is left adjacent to its verb can be reduplicated in Hungarian if it is shorter than 3 syllables long.

M19 A verbal particle can only be left adjacent to its verb in Hungarian if the verb is not preceded by an *only N* phrase.

[35] With impersonal verb, we mean a modal verb without arguments.

[36] Of course, the fact that numerals select singular nouns in Hungarian (**M2**) was already missed, and because **M2** was missed, **M14** was highly unlikely to be found, too. Typologically it is not necessarily surprising that Hungarian verbs show singular agreement with a noun phrase that contains a numeral, however there also exist languages, such as Russian, that show plural agreement with a noun phrase that contains a numeral (larger than one), despite (some) numerals selecting a singular noun; and therefore a difference like **M14** would ideally be found.

M20 Idem, if the verb is not preceded by a question phrase.

M21 Idem, if the verb is not preceded by sentential negation.

M22 A verbal particle cannot be left adjacent to its verb if the clause has progressive aspect.

None of these differences were found, however. This is probably due to several reasons, including orthography: whenever a verbal particle is left adjacent to its verb in Hungarian, it is attached to the verb as a prefix, while if it follows the verb it is not. The problem is that prefixes – or any affixes, for that matter – are not analysed as a separate token by `eflomal`, the aligner that we used. This results in frequencies of preverbs (that are not attached to the verb) being heavily underrepresented, and that the co-occurrence of English verbal particles and Hungarian preverb tokens may be too low for them to be consistently aligned to each other. In turn, this leads to many English verbal particles to be unaligned incorrectly.

Another reason includes tagging of verbal particles in English. UD treats Germanic verbal particles as adpositions or adverbs, making it very hard to distinguish verbal particles from other adpositions or adverbs in DGAE or GTI. In other words, there is no simple way to identify verbal particles in the output of our tools, and therefore to draw any conclusions concerning them.

Yet, even if the alignment and tagging problems were solved, the ability to correctly detect any morpho-syntactic features concerning verbal particles in Hungarian hinges on the assumption that all English verbal particles will always have a translation in Hungarian and vice versa, which may very well not be the case as this is lexical to a significant degree. There are examples of English so-called phrasal verbs of which the verbal particle does not have a (preverbal) translation in Hungarian, e.g. *ask around* vs. *kérdezősködik* and *call up* vs. *telefonál*. Conversely, there are many examples of Hungarian preverbs that have no direct translation to an English verbal particle, e.g. *return* vs. *visszatér*, which contains the prefix *vissza-* 'back' and can be separated from the verb. The fact that the presence of a verbal particle in both languages is lexically determined to a large extent, makes it very hard to detect them in the target language, seeing as the linguistic annotation of the source language is mapped onto the target language: if there is no verbal particle in the source language, it is impossible to see if it is present in the target language.

Detecting morpho-syntactic features of Hungarian with regards to verbal particles therefore proved very difficult. Setting aside the non-distinctive tagging of verbal particles, **M15** to **M17** were not found because of the high frequency of unaligned English verbal particles, leading to the impossibility to detect the position of the preverb in Hungarian. **M18** was not found for the additional reason that our tools do not correlate or associate the presence of words (or affixes) in the target language with other words in the target language, making it hard if not impossible to see that the verbal particle can be reduplicated. Ideally, the aligner aligns the English verbal particle to both real-

izations of the preverb, but this does not happen in practice. **M19** to **M22** are not found for the same reason as why it was not found what causes the copula to end up after the predicate: no correlation can be found between the position or any feature of a word in the target language and the presence of a specific type of phrase that is not the word's head or modifier.

4.4.4 Other hypotheses

Four more hypotheses were formed about Hungarian morpho-syntax, that do not necessarily fall under the nominal or verbal domain or under constituent order. One of these four was incorrect, while one difference on the AL list was missed.

It was hypothesized that Hungarian has both prepositions and postpositions (**H40**). Out of the aligned adpositions in English 55.9% had no crossing, meaning their Hungarian alignment shows up on the same side of their head in about half of the cases. **H40**, however, is incorrect: Hungarian only has postpositions, a difference also on the AL list. Although we believe that the numbers did suggest the presence of both prepositions and postpositions, the numbers were misleading. The group of English words that received the ADP tag also includes conjunctions (e.g. *for* and *as*) and, as mentioned, verbal particles. Verbal particles, in particular, end up after the verb in English, and whenever there is an alignment to a Hungarian word, that word will also be after the verb (seeing as Hungarian verbal particles preceding the verb will be prefixed to it in writing), leading to the absence of a crossing. As for the conjunctions, they always come before the verb, in both languages, leading to the absence of a crossing, as well. Further investigation laid bare problems with alignment, as many prepositions in English were aligned to Hungarian articles and other determiners.

Coordinating conjunctions were correctly hypothesized to precede the second conjunct (**H42**), as English conjunctions did not cross with their head in 85.3% of the aligned cases.[37] Similar numbers were found for subordinating conjunctions, which do not cross with their head verb in 88.4% of the aligned cases, and do not cross with their siblings, including subjects and objects, in 86.9% of the aligned cases – this led to the correct hypothesis **H43** that subordinating conjunctions mostly end up in the same position in Hungarian as in English.

Finally, it was missed that Hungarian is a negative concord language (**M23**). This feature of Hungarian could only be detected if English *not* was aligned to multiple Hungarian words at the same time (a one-to-many alignment), but this was only very rarely observed; or by correlating the presence of a Hungarian

[37] In UD the first conjunct is the head of the clause, while all other conjuncts depend on it via the `conj` relation. If, then, for example, *John and Mary* is the object in a sentence, only *John* would have the `obj` relation to the verb, while *Mary* would be a daughter node of *John*, having the `conj` relation. The conjunction *and*, in turn, would be a daughter node of *Mary* via the `cc` relation.

negation to the presence of another word or affix in the Hungarian sentence, which, as mentioned, is not done by our tools.

4.4.5 Hypotheses on affixes

Seeing as automatic affix detection is an important goal in the field of comparative syntax, we will conclude this section by discussing all hypotheses and differences pertaining to affixes in here. Specifically, we will explain how these hypotheses were formed.

The most important tools for affix detection are the AAA and the GTI. While the AAA tries to associate affixes with attribute bundles and retrieves a list of candidate affixes in the target language, the GTI can be used to further explore the data with these candidate affixes in mind.

Revisiting **H9** and **H10**, the accusative ending in Hungarian was hypothesized to be *-t* because AAA retrieved *-t* as being associated to the attribute bundle `(deprel=obj, parent=VERB|Trans, pos=NOUN)`. This means that the affix *-t* is very common in Hungarian words that are aligned to English nouns that have a direct object relation to their head verb. In fact, this association is the highest association found; see Figure 4.6 for the top-20 affix-attribute associations. Further inspection in the GTI, which gives more detailed breakdowns of attributes than DGAE, showed that (nearly) all English object nouns were aligned to Hungarian words ending in *-t*, leading to the correct hypothesis that *-t* is the accusative ending in Hungarian (**H9**), because, as mentioned above, this *-t* does not seem to occur with subjects of intransitive verbs. Despite the fact that AAA associates *-t* most strongly with nouns, and not pronouns, further exploration in the GTI suggested that pronouns do also often bear this accusative suffix *-t* (as shown in *őt* 'him/her', *őket* 'them', *melyet* 'which' and its plural form *melyeket*, *azt* 'it, that', *minket* 'us', among others), leading to the correct hypothesis **H10**: that *-t* is also the accusative marker for pronouns.[38]

To discuss the AAA output in slightly more detail – Figure 4.6 furthermore shows many associations between attribute bundles and the prefix *mond-* or the word *monda*. It is clear from the AAA output that this Hungarian prefix or word is associated with an indicative mood, a finite verb form, being an intransitive verb, and also a third person and a past tense in English. An association with the English word *and* being a daughter node of the verb is also found. Further inspection of the data strongly suggests that *mond-* in fact means 'to say' (which turned out to be correct), a verb that is very common in the Bible, especially in the third person, past tense and with the word *and* being a daughter node.

The Hungarian affix *meg-* is reported by AAA to be somewhat highly associated with English transitive verbs in the past tense. Despite inspection of

[38] However, it turns out that there are two exceptions to this: *engem* 'me' and *teged* 'you (sg.)'. These two forms nonetheless do appear in dialectal Hungarian as *engemet* and *tegedet*, respectively, although they are not attested in the Bible translation that was used in this research.

attribute bundle	affix	weight
(feats=(Tense=Past), feats=(VerbForm=Part))	-tt	0.116655
(deprel=obj, parent=VERB|Trans, pos=NOUN)	-et	0.125745
(feats=(VerbForm=Inf))	-k	0.126978
(feats=(Tense=Past), pos=VERB|Intrans)	mond-	0.127453
(deprel=obl, feats=(Number=Sing), pos=NOUN)	-l	0.127497
(feats=(Tense=Past), feats=(VerbForm=Part))	-k	0.128595
(children=NOUN|nmod, children=the|det, pos=NOUN)	f-	0.130742
(children=and|cc, feats=(Mood=Ind), feats=(VerbForm=Fin))	monda	0.132111
(feats=(PronType=Prs))	nék-	0.132633
(children=of|case, pos=NOUN)	-nak	0.135104
(feats=(Mood=Ind), feats=(Person=3), feats=(Tense=Past), feats=(VerbForm=Fin))	monda	0.138214
(feats=(Number=Plur), parent=NOUN)	-k	0.142976
(feats=(Mood=Ind), feats=(VerbForm=Fin), pos=VERB|Intrans)	monda	0.144797
(children=of|case, pos=NOUN)	-k	0.145477
(feats=(Tense=Past), pos=VERB|Trans)	meg-	0.145837
(feats=(Number=Sing), parent=VERB|Trans, pos=NOUN)	-t	0.155972
(children=and|cc, feats=(Mood=Ind), feats=(VerbForm=Fin))	mond-	0.168884
(feats=(Mood=Ind), feats=(Person=3), feats=(Tense=Past), feats=(VerbForm=Fin))	mond-	0.172108
(feats=(Mood=Ind), feats=(VerbForm=Fin), pos=VERB|Intrans)	mond-	0.225624
(deprel=obj, parent=VERB|Trans, pos=NOUN)	-t	0.41646

Figure 4.6: The top-20 affix-attribute bundle associations as retrieved by AAA. The higher the weight, the higher the association is between the affix and the attribute bundles.

the data it could not be narrowed down what this prefix means, however it was later revealed in Rounds (2009) that *meg-* is an aspectual prefix called a preverb, which are also discussed above.

In Figure 4.6 it can also be observed that the ending *-k* is associated with several attribute bundles. Indeed, many Hungarian words that are aligned to English nouns that have a child preposition *of* end in *-k* (because a genitive construction in Hungarian is expressed using the suffix *-nak* – which can also be found in Figure 4.6 – and *-nek*, or using a zero suffix; Rounds 2009), and to English plural nouns (the nominative plural ends in *-k*, possibly with a preceding linking vowel; Rounds 2009). Though the AAA output shows that the ending is also associated with English past participles and infinitives, Hungarian past participles and infinitives do not end in *-k* (Rounds 2009). This association as returned by AAA can be explained through the fact that Hungarian main verbs are aligned to English main verbs, and English main verbs are often non-finite, with auxiliaries showing finite verbal morphology. Indeed, Rounds (2009) confirms that many Hungarian finite verbal forms (which would be aligned to English non-finite, main verbs; a result of Hungarian having much fewer auxiliaries than English, see **H31**) end in *-k*: plural forms all end in *-k*, as well as the first person singular in certain forms. The fact that AAA wrongly (or at least incompletely) retrieves the suffix *-k* as what can be interpreted as a genitive suffix and as what can be interpreted as a non-finite verbal suffix, can therefore be explained by the confusion of multiple alternating suffixes (such as *-nak* and *-nek*), as well as different suffixes that share an attribute in the English annotation and happen to both end in *-k*.

Among the top-20 affix-attribute bundle associations is also the ending *-l*, which is associated to singular nouns that have an `obl` relation to their head, used to denote non-core (oblique) arguments or adjuncts. While *-l* is not a case suffix in Hungarian nominal morphology in itself, several case endings end in *-l*: the elative, delative, adessive, ablative, instrumental and sociative cases are all denoted with a suffix that ends in *-l* (Rounds 2009). Such noun phrases would typically receive an `obl` relation in UD. Similar to what happens with *-k*, *-l* is retrieved because of the confusion of multiple, longer suffixes that share an attribute in the English annotation and happen to all end in *-l*.

AAA found several suffix pairs associated with the same attribute bundles that had an alternating vowel. For example, *-nak* and *-nek* both appear three times in the AAA output: once associated with `(children=of|case, pos=NOUN)` (also seen in Figure 4.6); once with `(deprel=nmod, feats=(Number=Sing), parent=NOUN, pos=NOUN)`; and once with `(deprel=nmod, parent=NOUN, pos=NOUN)`. In a similar fashion, *-ból* and *-ből* are associated to the same attribute bundles, as are *-ban/-ben* and *-tok/-tek*. The fact that this alternation did not seem to be caused by any other morpho-syntactic or lexical feature, such as gender (cf. *embernek* 'man' vs. *királynak* 'king') led to the correct hypothesis **H11**: that Hungarian shows a systematic form of vowel harmony, most

likely to be front-back vowel harmony.[39]

The stacking of suffixes, such as seen in *őt* 'him/her' vs. *őket* 'them', *melyet* vs. *melyeket* 'which', as well as in *ember* 'man' vs. *embernek* 'of/to (the) man' vs. *emberek* 'men' vs. *embereknek* 'of/to (the) men', led to the correct hypothesis that Hungarian is agglutinative (**H12**), as case suffixes, such as *-nak/-nek* and *-t*, are stacked onto the plural suffix *-k*, sometimes with a linking vowel. AAA found a few stacked affixes, among which *-(o)kat* (associated with plural nouns, as well as object nouns; the suffix indeed corresponds to plural object nouns) and *-knak* (equivalent to plural *-k* + dative *-nak*).

Concerning verbal morphology, it was correctly hypothesized that verbs inflect for all persons in both present and past tense (**H29** and **H30**), a morpho-syntactic difference between English and Hungarian also on the AL list. This feature of Hungarian was found by observing very rich morphology in DGAE and GTI; English verb lemmas were aligned to a plethora of different Hungarian words, which occurred in different forms with distinct endings and prefixes. By identifying the subjects of the English verbs in the attribute bundles (subjects are children of the verb in UD), these endings could be clearly matched to a grammatical person. As such, the verbs *mond-* 'to see' and *tud-* 'to know' can be observed in several forms in the present and past tense, where all persons receive distinct endings; see Table 4.8. Of the observed endings in the Table, AAA correctly discovered *-ának* as being associated with third person plural past tense indicative. It also found *-om* though with an incomplete attribute bundle associated to it. AAA additionally found front-vowel counterparts of two listed suffixes: *-em* and *-ünk*, though both with incomplete attribute bundles as well.

It must be noted that the columns in Table 4.8 turned out to contain multiple paradigms: while *mondom* is indicative, *mondjak* is subjunctive, for instance. However, the observed forms still prove that verbs decline for all persons, in both present and past (although the second person plural in the past tense was not observed for both verbs; it still seemed a safe – and indeed correct – assumption that it would receive a suffix distinct from all other persons). Also note that the first person plural either seems to receive *-unk* or *-juk* in the present tense: this is due to Hungarian verbs agreeing with the definiteness of the object, a missed difference also discussed above (**M12**).

With AAA it was also hypothesized that infinitives in Hungarian end in *-ni* (which is correct) and that *-á/-é* is a frequent past tense suffix, possibly third person singular (correct). Furthermore, *-tt* was hypothesized to be a past participle – however, the real ending turned out to be *-ott/-ött*, and turned

[39] There are more possible explanations for the alternation of suffixes, such as dissimilation, gender or other word classes, or simply multiple noun declinations, that should in principle be tested. However, when reviewing the data, I noticed there seemed to be a correlation between the presence of certain vowels in the stem and the vowel in the suffix, but I never quantified this correlation. In forming the hypothesis, I may have been somewhat guided by my limited knowledge of Finnish, a language related to Hungarian, of which I know it has vowel harmony.

subject	***mond-* 'to see'**		***tud-* 'to know'**	
	PRS	PST	PRS	PST
I	mond-om, mond-jak	mond-ám	tud-om	
thou		mond-ál	tud-od	
he/she/it	mond-ja	mond-a		
we	mond-unk	mond-ottuk	tud-juk	
ye			tud-játok	
they	mond-ják	mond-ának	tud-nak	tud-ják

Table 4.8: Observed forms of the Hungarian verbs *mond* 'to see' and *tud* 'to know', in present and past. Note that the columns contain multiple paradigms.

out to be also used in the finite past tense. The ending *-ék* was hypothesized to be third person past indicative, but that is not entirely correct: the third person singular does not show any such ending, while the plural does but with an additional *j*, *t* or *n* before it. The ending *-ék* is therefore likely to be a result of the algorithm trying to generalize over *-jék*, *-ték* and *-nék*.

head	***nék-* 'unto'**	***ellen-* 'against'**
me	nék-em	ellen-em
thee		ellen-ed(?)
him/her	nék-i	ellen-e
us		ellen-ünk
you (pl.)	nék-tek	ellen-etek
them	nék-ik	ellen-ük(?)
NOUN		ellen

Table 4.9: Some postpositions in Hungarian decline for person, such as *nék-* 'unto' and *ellen-* 'against'. Listed are some attested forms.

Furthermore on adpositions, it was correctly hypothesized that some postpositions in Hungarian decline for person (**H41**). This was found by observing that the prepositions *unto* and *against* were aligned to multiple Hungarian words, depending on the head of the preposition.[40] It was thus observed that all Hungarian aligned words started with *nék-* 'unto'[41] and *ellen-* 'against', and have different endings for each different pronominal head (which are reminiscent of verbal endings) as shown in Table 4.9. The preposition *against*

[40] In UD nouns and pronouns are the heads of prepositions, because it follows the convention that all functional words are dependent on content words. This is done in order to parse sentences more uniformly cross-linguistically.

[41] *Nék-* is an archaic or dialectal variant of modern *nek-* 'to, for'.

furthermore shows that it does not receive an ending if its head is a noun.

4.5 Discussion

4.5.1 On the results and subjectivity

The results discussed in the previous section show that our tools are effective and useful in the detection of morpho-syntactic features of a language. It was observed that the large majority of the hypotheses that were formed by analyzing the output of the DGAE, GTI and AAA are correct. Not only do the hypotheses formed have a high precision, the output of the tools even gave rise to two questions about Hungarian syntax, the answers to which are to the best of our knowledge as of yet unknown: both H26 (there is a stricter word order in Hungarian subordinated clauses than in main clauses) and H37 (Hungarian uses fewer pro-adverbs than English) remain to be confirmed or rejected. On the other hand, several differences on the AL list were not found, indicating that our tools do not detect every difference. However, many of these missed differences can be attributed to either the interpretation of the output by the linguist (e.g. M4), lacking annotations (e.g. M6–8) or the processing and formatting of the data by the tools (e.g. M2 and M3), for all of which there is room for improvement.

Of course, there is no objective measure of performance of our tools. In this research, we tried to overcome this lack of a formal test set by compiling a list of hypotheses based on the output of the tools, while an expert of Hungarian independently compiled a list of characteristic morpho-syntactic differences between Hungarian and English (the AL list). Both lists are far from complete, and many more differences could have been discovered (and hypotheses formed) with the help of our tools, and many more differences exist that were not on the AL list. While we think we have sufficiently shown that our tools can successfully aid a linguist in the detection of syntactic differences between a source and a target language, the evaluation carried out in this chapter does not give a complete overview of the full range of possibilities and, especially, the shortcomings of the proposed method and presented tools. Ideally, a more objective measure or a dataset should be developed to more adequately grasp the performance of tools for the automatic detection of syntactic differences between languages, but it is not clear at present how this could be achieved.

As mentioned, many of the missed differences can be attributed to the interpretation of the linguist. In our tools, we have left substantial room for the linguist to interpret results. While the advantage is that the linguist can use any prior knowledge about the language or its family, or more general linguistic expertise that they may possess in order to form more informed hypotheses, this can lead to bias. We have seen this happen in the forming of H11, in which it was hypothesized that Hungarian has vowel harmony. Although it turned out to be correct, the conclusion may have been guided by the author's knowledge

of Finnish, a language related to Hungarian, which also has productive front-back vowel harmony, and was drawn too quickly, as there are other plausible explanations of the vowel alternation that was observed in a few suffixes. Other hypotheses may have been somewhat steep as well, but the interpretation of the output is sometimes difficult, in which case linguistic knowledge can aid the user to arrive at the forming of a hypothesis – whether correct or wrong, a hypothesis should always lead to closer inspection of the data.

It can be argued that the interpretation of the output should be made less subjective, by having a computer interpret (a part) of the results and automatizing the generation of hypotheses. One can think of a list of questions about the target language that a linguist will always ask and the tools should minimally be able to answer, but while it will reduce the subjectivity of the results, one will only get answers to questions directly posed to the algorithm beforehand. That is to say, the algorithm will only discover differences for which it was expressly programmed to look, and the output will only be interpreted by the algorithm in ways it was expressly programmed to do so. We believe that a good balance can be struck between the freedom for subjective interpretation on the one hand and the more computer-driven generation of hypotheses on the other, though whatever the tendency in the balance struck, the expertise and subjective interpretation of the linguist will always be there: either the linguistic bias will be present in the interpretation of the output, or the linguistic bias will be put in the design of the algorithm.

4.5.2 Other remarks on the methodology

Several other factors that influence which hypotheses are or can be formed can be identified, apart from the interpretation of the output. First, it was observed that the choice of source language and target language influences the results, despite the tools having been designed to be language-independent. Due to the unilateral mapping of linguistic annotation from the source language onto the target language based on word alignments, the user may fail to detect any morpho-syntactic features that concern unaligned words in the target language. For example, English allows for the dropping of the conjunction *that* in relative and subordinating clauses. Hungarian, however, does not allow for the dropping of its equivalents *hogy* 'that (conj.)' and *(a)mely* 'that, which', but if English *that* is absent no linguistic annotations are mapped onto *hogy* or *(a)mely* through alignment, and the Hungarian words are in fact completely absent in our tools' output, leading to this difference being undetected. Similarly, differences can remain undetected when a word type and its equivalent in the target language occur in a completely complementary distribution. As a fictive example, it could have been the case that the English infinitival marker *to* only occurred after aspectual verbs, while a Hungarian equivalent infinitival marker only occurred after modal verbs. In that case, the linguist would be led to form a hypothesis such as H33 (that Hungarian does not have infinitival markers at all), because English *to* would never be aligned.

Giving a frequency overview of all unaligned target-language words will most likely not provide further information, because there would be no linguistic information or annotation mapped onto them; the linguist would not know what each word means and in what context it was encountered. It would therefore be very hard to conclude anything about unaligned target-language words, and to form hypotheses about morpho-syntactic differences based on them.

This 'blind spot' could perhaps be remedied in several ways. Choosing two languages that are closely related could maximize the number of words in the source language being aligned, securing a high quantity of linguistic annotation being mapped onto the target language. Similarly, one could argue to choose a source language that is highly analytical, which could ensure that as many words in the target language as possible are aligned to a morpheme in the source language. Yet another remedy would be to run the entire experiment twice, with two different source languages. The right choice of two (or in fact, more) complementary source languages (e.g. one language that has reflexive verbs and one that does not) can diminish the size of the blind spot. We believe the latter remedy is the most straightforward and feasible option when there is no linguistic knowledge of the target language at all.

When linguistically annotated corpora or automatic taggers and parsers for the target language do exist, the linguist can also consider to run the experiment twice, but with the source and the target language swapped. Words in the target language that do not receive an alignment in the first run will be linguistically annotated in the second run, allowing for the linguist to form hypotheses. However, annotated corpora or taggers and parsers for the target language were assumed not to exist for the purpose of this research. Additionally, adding annotations for the target language may have negative effects, especially when the annotations are not perfect: Kroon et al. (2020) report that the quality of the annotations led to noisy, hard to interpret results and to the detection of differences in annotation guidelines.

Secondly, the user chooses a few parameters that are passed to the tools, the choice of which may influence results, as well. For instance, in our experiment the GTI output is suppressed by not outputting partitionings of the data if they are smaller than 1% of their parent partition or if the partition contains fewer than five words. While it is meant to control the overflow of output and to suppress noise, it also can also result in some infrequent phenomena not being retrieved by GTI. One of the issues why M2 (Hungarian noun phrases containing a numeral have a singular head noun) was missed, is this suppressing of the output, as numerals are relatively rare. Only 6073 out of the 737319 tokens in the English Bible were tagged as a numeral, amounting to only 0.8%. This suppression threshold, however, leads to a trade-off, as increasing it may lead to more infrequent phenomena being missed, while lowering it may retrieve more noise, which could increase the number of incorrect hypotheses formed.

A last factor that can influence the results is the matter of the genre of the corpus. As with any linguistic research, our tools and method are subject to

the genre of the input corpus, and can only detect differences that are extant in the data. In the case of English and Hungarian Bibles, it will not be found that Hungarian has a distinct second person singular and second person plural pronoun, a difference with modern English. This is because in the KJB the now somewhat archaic pronoun *thou* is still frequently used for singular, while *you* is exclusive to plural (where *ye* is also abundantly used). Similarly, M6, M7, M8, M9 and M10 were all missed (which all have to do with questions) partly because direct questions are not very frequent in the Bible. A final example of the influence of corpus genre on our results is H32, which expresses our hypothesis that Hungarian has a synthetic passive voice. Dr. Lipták pointed out that modern Hungarian does not have a passive voice at all, but in the Bible, which is written in more archaic Hungarian, there still exists a synthetic passive, making our hypothesis only true for this specific corpus.

4.5.3 Points of improvement and future research

Some specific points of interest for future research and the improvement of results can also be identified. Perhaps the most prominent possible improvement is the implementation of automatic outlier detection. By for instance automatically retrieving combinations of attributes that are unexpectedly frequent, the linguist will be aided by being pointed towards possible differences for which they may not have been looking (e.g. Dutch verbs in a subordinating clause are "unexpectedly" frequently occurring with a crossing with the object when compared to English, directly leading a linguist to Dutch's SOV order in subordinating clauses). In turn, this would increase the number of differences found as well as leave less room for subjective interpretation, which would play into the balance between automation and interpretation discussed above.

On that note, it would be very helpful if co-occurrences of attributes were reported in the output. As of now, our tools only output frequencies of single attributes. While this is already very useful, unusually frequent co-occurrences can lead a linguist to forming more informed hypotheses. In order to suppress the output somewhat, because the number of combinations of attributes quickly explodes, one could perform some statistical test and only return the most statistically significant or those that exceed some threshold.

Furthermore, it can be insightful to track adjacencies in the target language. That is to say, the linguist can discover more differences pertaining to (phonological) context or possibly to target-language words that were left unaligned, when the words directly adjacent to the aligned-to word in the target language are also present among the source-language word's attributes. For instance, the difference in usage between Hungarian *a* and *az* 'the' can only be discovered when the word directly following it is somehow accessible in the output of our tools; only then can it be observed that *a* precedes only words beginning with a consonant and *az* only words beginning with a vowel. Moreover, it would allow the linguist to discover that demonstratives and articles must co-occur in Hungarian (M3).

Deriving more information from the dependency tree in the source language can also be beneficial. In our current approach it was already derived that a verb is transitive or intransitive, but it could similarly be derived that a verb is ditransitive, or that it takes a complement in a specific case form, which could lay bare more differences between two languages. Additionally, it could be useful to automatically derive from the dependency tree that a verb is third person when its subject is non-pronominal.

It was observed in the English-Hungarian experiment that our AAA tool may not be ideal for agglutinative languages. While it already retrieved some useful potential affixes in Hungarian, many affixes turned out to be incomplete or noise. We think this may be the case because it was designed only to consider prefixes and suffixes that include the beginning or the end of the word. The result of this is that if suffixes are stacked in the target language – for instance the Hungarian plural marker *-ak* and the inessive marker *-ban* – AAA will calculate an association value between *-ban* and the attribute `(children=(in|case))`, and between *-akban* and the attribute bundle `(children=(in|case), feats=(Number=Plur))`, but not between *-ak* and the attribute `(feats=(Number=Plur))`, thus underrepresenting the frequency and the association value of the plural marker. Ideally AAA also considers affixes that do not necessarily contain the word boundary, as well as even discontinuous affixes (such as the Hungarian superlative circumfix *leg*⟩…⟨*bb*), however the number of affixes to consider would grow exponentially, making the current algorithmic design unfeasible. Given that AAA is already subject to an exponential blow-up as a result of considering all potential attribute sub-bundles, AAA in particular should be improved by increasing its computational efficiency, especially when discontinuous affixes and infixes are to be considered as well.

Another very interesting potential improvement would be to tag adverbs for their type, such as modal, temporal, aspectual or even more detailed. As of now, adverbs are indiscriminately tagged in UD, but distinguishing between different subtypes would make it possible to automatically test the hierarchy of clausal functional projections as proposed by Cinque (1999) with our tools, and to detect any differences in use or relative order of adverbs between the source and the target language.

Similarly, tagging verbs or sentences for aspect would allow our tools to successfully detect the Hungarian coverbs, such as *meg-*, along with associating it with their aspectual attribute.

On the subject of improving tagging and parsing, the used parser model is of course not fully appropriate for use on the Bible. Additionally, any improvements in aligning will benefit the proposed method, as alignments obtained with `eflomal` (Östling and Tiedemann 2016) were far from perfect and newer neural approaches such as SimAlign (Jalili Sabet et al. 2020) only marginally improve on older models in exchange for higher computational requirements and a very steep increase in run-time. However, despite imperfect parses, tags and alignments, we have found many correct hypotheses on Hungarian morpho-syntax,

underlining the power of our method and tools. One could only speculate on the quality and quantity of the hypotheses and detected differences when the corpus were perfectly annotated.

4.6 Conclusion

In this chapter I have explored the possibility of detecting morpho-syntactic differences between an annotated source language and an un-annotated target language by using bitext alignment in order to map the annotation of the source language onto the target language and to derive several morpho-syntactic features of the target language. It was shown that our tools can be used effectively to form many correct hypotheses on differences between English and Hungarian in several syntactic domains and to extract potential affixes in Hungarian. Despite some room for improvement, I believe this research can pave the way for future research towards a pipeline for automated comparative-syntactic research.

CHAPTER 5

Discussion and conclusion

Over the course of this dissertation I have researched the question of whether it is possible to automatically detect syntactic differences and, if so, how. Before concluding and answering that question, I will briefly summarize the findings of each Chapter, and discuss the findings of all the Chapters in their respective relative contexts.

5.1 Brief summary of previous Chapters

In Chapter 2 the issue of syntactically incomparable sentence pairs was addressed. In parallel corpora it is not a given that sentences that are aligned to one another are syntactically comparable, as they may exhibit vastly different constructions or a free translation. A method and measure was needed to filter out sentence pairs that are syntactically too different, because using free translations, wrongly aligned sentence pairs or translations that are structurally too different for the detection of syntactic differences between the two languages can influence the results negatively.

To this end, four different filtering approaches (one based on the sentence-length ratio, one based on the Levenshtein distance on POS tags, one on the graph edit distance (GED) on dependency parses and one that combines the previous three filters in a regression model) were explored. The results of the ex-

periments on datasets of English, Dutch and German parallel sentences suggest chiefly that filtering for syntactic comparability is a hard task, in part because syntactic comparability is hard to define, which interacts with the trade-off between cleaner data and losing desired variation. The fact that the task is hard was also corroborated by the only moderate inter-annotator agreement, which ranged between 0.61 and 0.26. Nevertheless, the presented filters are useful tools for automatizing the selection of syntactically comparable sentences from a parallel corpus. The filtering approach that combines the other three filters works best, however it requires the existence of a pre-labelled dataset on which it can be trained, is computationally expensive and has a high risk of overfitting on the dataset. In general it was observed that, as expected, using syntactic information (of any kind) gives better results: the Levenshtein distance and the GED outperform the sentence-length ratio. The robustness in its parameters throughout the language pairs furthermore suggested that the GED approach can be used as a default filter, especially when a pre-labelled dataset is not available. This would make sense, as the GED filter uses the most syntactic information and is less sensitive to phrases or constituents transposing. The Levenshtein distance can also give reasonable results, but is expected only to perform well on closely related language pairs, in which the word order is more or less similar.

In Chapter 3 I presented a systematic approach to detect and rank hypotheses about possible syntactic differences for further investigation by leveraging parallel data and using the Minimum Description Length (MDL) principle, which provides an elegant paradigm to find structure in data (among others Grünwald 2007; Barron, Rissanen and Yu 1998). The approach deploys the MDL-based pattern mining algorithm SQS ('Summarising event seQuenceS'; Tatti and Vreeken 2012) to extract sequences of POS tags that can be considered 'typical' syntactic building blocks of a language. From the lists of these POS patterns of two languages, a shortlist of potential syntactic differences is created based on the number of parallel sentences with a mismatch in pattern occurrence. The patterns are then ranked on a χ^2 value calculated from these mismatch frequencies, generating hypotheses on where syntactic differences may be found within the language pair.

The approach was evaluated on parallel corpora of English, Dutch and Czech, and proved useful in both retrieving POS building blocks of a language, which can already be of use to detect broad typological characteristics, as well as pointing to meaningful syntactic differences between languages. Apart from that, with the approach it is possible to detect tagging inconsistencies between two languages easily. It was however observed that the approach is very sensitive to tagging quality, with tagging inconsistencies between languages (i.e. different conventions or annotation guidelines) and tagging inaccuracies within languages (i.e. tagging errors) heavily influencing results. Despite this clear sensitivity to tagging quality, our results and approach are promising, with many hypotheses being generated by the algorithm that proved to be correct.

In Chapter 4 a different approach was explored to detect morpho-syntactic

differences that is, unlike the MDL approach of Chapter 3, not dependent on the availability of natural language processing (NLP) tools for both languages under investigation. The key question of Chapter 4 was whether it is possible to use fully annotated text in language A (called the source language) to detect grammatical properties of a different, less well-described language B (called the target language), and differences between the two languages, in parallel text. To this end, word alignment is used to map source language words to target language words with the aim of detecting syntactic features of the target language and differences between source and target language by semi-automatically analysing this mapping. Three tools were developed to detect syntactic properties and differences from parallel data aligned on a word level: the Data Grouper for Attribute Exploration (DGAE), the Generalization Tree Inducer (GTI), and the Affix-Attribute Associator (AAA). These three tools were evaluated on the language pair English-Hungarian. With the help of the tools 43 hypotheses on morpho-syntactic features of Hungarian or differences between it and English were generated. The hypotheses were independently checked by a native speaker and expert of Hungarian and its syntax, and cross-checked with a list of characteristic differences between Hungarian and English independently compiled by said expert. It was concluded that the tools can be used very effectively to form many correct hypotheses on differences between the languages in several syntactic domains. With the help of the tools, I even generated two hypotheses of which the correctness is yet to be investigated, highlighting the power of the tools in the search for syntactic differences between languages.

5.2 Relating the filter to MDL and alignment

In Chapter 3 we have experimented with the influence of the filter from Chapter 2 on the results from the automatic detection of syntactic differences. It was observed that the Europarl corpus (Koehn 2005), of which a fragment was used in Chapter 3, suffered from free translations and wrongly aligned sentence pairs, which led us to believe that the filter could be deployed successfully.

The design of the filters made it such that the combination, i.e. regression, filter requires a training set of sentence pairs binarily labelled for syntactic comparability, and that the other three filters use a threshold value, which can be set manually or with the use of a grid search on a training set. However, because there was no pre-labelled data set on which the filter could be trained for the purposes of the research of Chapter 3, there was no possibility to deploy the combination filter, that had been found to work best, or to do a grid search for the other three filters. Instead, the GED-based filter was used with a threshold value of 4, which was already suggested as a possible default value for the GED filter in Chapter 2.

The results of the experiments with the filter in Chapter 3 show that using the filter does indeed influence the results. First and foremost, applying the filter results in a significant loss of data. After filtering out incomparable sen-

tence pairs using the GED-based filter, only about one fifth or one sixth of the sentences remained in the data.

This strong reduction of data is probably due to a three-way interaction. The first factor is simply the noisiness of the data: there are a significant number of sentences that are wrongly aligned in Europarl, and an even larger number whose translations are too free for the purposes of comparative-syntactic research. These sentence pairs we wanted to filter out. The second factor is that the filter, not unlike the MDL approach itself, is sensitive to tagging errors and inconsistencies: if a label is incorrect, the edit distance between the two sentences will be higher, which may push the sentence pair over the threshold and have it be discarded wrongly. The last factor is that it may be the case that the threshold value of 4 is not appropriate for the dataset used. Since a training set was not available for the setting of the threshold, however, we had to resort to parameters that were shown to work well in Chapter 2 for a filter that was hypothesized to be robust throughout different language pairs.

It was furthermore observed that the filter had only a marginal effect on the quality of the output of the MDL approach. Filtering resulted in somewhat more useful hypotheses on syntactic differences between English and Dutch, as it reduced that number of patterns ranking highly due to tagging issues. As for the Czech runs, the opposite was true. While for the comparison between Dutch and Czech the difference seemed insignificant, for the comparison between English and Czech the number of useful patterns went down and it strikingly made the approach unable to detect that Czech does not have articles. Nevertheless, filtering the data makes the patterns easier to interpret, because they are generally shorter and contain less noise.

The filter was not deployed in Chapter 4. This is because the combination filter, GED-based filter and the Levenshtein-distance filter require the availability of annotation tools for both languages under consideration, while the alignment approach was developed with the assumption that annotation tools would only be available for one of the two languages. In principle the sentence-length filter could have been deployed, but it was seen in Chapter 2 that the sentence-length filter did not yield satisfactory results and we therefore opted not to deploy the filter at all.

On the influence of the filter on the results of the alignment approach when tools are available for both languages one can speculate that the filter can be of added value. It can be expected that applying the filter on the data before running the tools of the alignment approach will mostly have an effect on the quality of the alignments. The result will be that zero-alignments, i.e. words that do not get aligned to a word in the other language, and noisy alignment crossings will be less frequent, because the sentence pairs are more translationally equivalent and syntactically comparable. In general it can be expected that it will lead to more interpretable output of the tools and better hypotheses, however I did not experiment with the application of the filter to the alignment approach.

All in all, applying the filter is a trade-off between more comparable and

"cleaner" data and more interpretable output of the tools on the one hand, and the undesired removal of variation from the data on the other, which ties in with the discussion on what syntactic comparability is in Chapter 2. All of this raises the question: is using the filter for syntactically incomparable parallel sentences necessary when automatically detecting syntactic differences? I would like to hypothesize here that it depends on the sensitivity to noise of the method to detect syntactic differences that is used. The MDL approach fundamentally uses high frequencies in both the mining for patterns and the detection of differences, so it can be expected that the effect of the filter remains minimal as long as the size of the data is sufficiently large for the signal-to-noise ratio to be largely in favour of the signal – for the more frequent patterns, that is. As for the bottom half of the pattern lists, it can be expected that the effect of the filter is much larger, because a small change in frequency of a less frequent pattern (as a result of the filter) has a larger impact on its ability to efficiently compress the data and its statistical significance. The alignment approach, on the other hand, is probably much more sensitive to the effect of the filter, as was already discussed above.

In Chapter 3 it was already concluded that filtering out syntactically incomparable sentences is beneficial to the results. However, it depends on the situation, and the user should consider several things.

First, applying the filter drastically reduces the size of the data. When a user only has a fairly small dataset at their disposal, applying the filter may therefore be ill-advised. Though, when a user has a large dataset at their disposal, applying the filter may not be necessary when the tool used for the detection of syntactic differences is not very sensitive to noise, as was seen with the MDL approach, and may even be advised against due to the filter's computational expense, especially that of the GED-based filter. Applying the filter is therefore most interesting for middle-sized datasets, however it is very difficult to demarcate the boundaries of what constitutes a small, middle or large dataset. The issue of drastic data reduction would be greatly counteracted if a filter is developed that selects syntactically comparable sentence fragments. A possible way to achieve this is by for instance using punctuation to delineate smaller clauses and use those instead of full sentences, however the details to the implementation of this is left to future research endeavours.

Secondly, it depends on the noisiness of the data. As long as the signal-to-noise ratio is in favour of the signal, that is to say the data are clean, then applying the filter will not be necessary. However, when the data are noticeably noisy, i.e. containing many wrongly aligned sentence pairs, many free translation or syntactically incomparable constructions, then the user may opt to deploy the filter. It may therefore be advised first to run the MDL or alignment approach and to see if results are good.

Lastly, deploying the filter depends on the availability of a training set of sentence pairs, binarily labelled for syntactic comparability. The best filter was the combination filter, which was built on a logistic regression model and can only be used when a training set exists. Otherwise, the user would have

to resort to one of the other three developed filters and use a manually set threshold value, which may not be appropriate for the dataset in question.

As for the choice of which filter to use, the use of the combination filter is to be advised, but this can only be done, as said, when a training set exists. If such a set does not exist, the GED-based filter would be advised, but requires that there exist parsers for both languages that use the same annotation guidelines (such as Universal Dependencies). However, parses are rarely perfect which can lead to sentence pairs incorrectly being discarded, and, as mentioned, the GED-based filter is notably slow. The Levenshtein-distance filter is advised only when working with closely-related languages, because the Levenshtein distance is very sensitive to whole phrases transposing, and requires the existence of POS taggers for both languages that use the same annotation guidelines as well. If the user, for instance, is comparing English to Japanese, it would be ill-advised to use the Levenshtein-distance filter, but comparing Dutch to German should give reasonable results. The sentence-length filter is not advised, because it generally is too coarse-grained and does not use syntactic information.

5.3 Comparing MDL and alignment

In this Section I will compare the MDL approach of Chapter 3 to the alignment approach of Chapter 4. Very globally it can already be established that the MDL approach finds other types of differences than the alignment approach, simply because they process different types of data: the MDL approach uses linear POS tags and sequences, while the alignment approach operationalizes bitext word alignment and makes use of hierarchical dependency parses containing syntactic relations, POS tags and morphological features. Nevertheless, some valuable observations can be made when contrasting the results of the two approaches. Of course, the alignment approach put forth in Chapter 4 was developed from the assumption that no automatic annotation tools are available for one of the two languages under investigation, while the MDL approach of Chapter 3 requires the existence of (at least) POS taggers for both languages (that use the same tag set). Therefore the two approaches may be used in complementary situations, however for the purposes of this Section, I will assume a situation in which annotation tools are available for both languages so that both approaches could be deployed.

The foremost question is perhaps that of which type of syntactic differences can be found with the one approach but not with the other. The global answer to this question is that it depends on which information is passed to the system. As said above, the MDL approach uses POS tags and sequences, while the alignment approach uses dependency parses and alignment. The result is that any differences regarding syntactic function (i.e. dependency relation) or morphology can in principle not be found with the MDL approach without extensive manual research within the generated hypotheses, unless it is specifically coded into the POS tags. In Chapter 3 it was already discussed that the user could

opt for expanding the tag set such that it also reflects morphological, or indeed syntactic, information, by for instance appending the grammatical number to a POS tag. The issue with this, however, is that the MDL approach treats tags in a univariate way, i.e. a tag NOUN:Num=Sing (for singular nouns) is fundamentally distinct from a tag NOUN:Num=Plur (for plural nouns), as much as it is distinct from a tag for third person singular auxiliary verbs. This algorithmic behaviour is contrasted with the alignment approach, in which all annotation is processed in a multivariate way, such that the algorithm recognizes that a singular noun and a plural noun are both nouns and therefore more similar to each other than to an auxiliary verb. Expanding the tag set in MDL, therefore, is a trade-off between richer annotation and therefore more detailed syntactic differences that can be discovered on the one hand, and a loss of information and desired similarity between words due to further discretization of the data on the other.

Due to its more coarse-grained input and univariate nature, it can be concluded that the MDL approach is more prone to overgeneralization than the alignment approach. For instance, with the MDL approach it can be detected that pro-drop is extant in Finnish, but because the algorithm cannot distinguish between first, second and third person pronouns without creating more tags, causing the issue described above, it cannot directly show the linguist that Finnish pro-drop only affects non-third person pronouns.[1] A difference found with MDL should therefore very expressly lead to further investigation.

Meanwhile, the opposite holds true for the alignment approach. Its multivariate way of processing data and access to more detailed annotation lead it to being more prone to undergeneralization. This was for instance seen with the missed difference M4 from Chapter 4, which signified Hungarian's pro-drop also applying to singular object pronouns: I undergeneralized over the output of the tools and only concluded from the data that Hungarian has subject pro-drop. Although we have seen that the output of the alignment approach can lead to overgeneralization, too, the linguist may fail to detect a difference or feature as a result of being confronted with too much information.

Some smaller observations can also be made when comparing the two approaches. Related to the dropping of material, a notable difference between the two approaches is their applicability in tracking potential words or word types that are not overt in the other language, often involving functional material such as articles or personal pronouns which may be dropped or even be entirely absent in a language. Because the alignment approach operationalizes word alignment, it is fairly straightforward to track with it which POS tags (or even which combination of attributes of a word) often remain unaligned and untranslated in the unannotated target language: the developed tools retrieve the frequencies of unaligned cases of particular (combinations of) attributes, which in Chapter 4 quickly laid bare that Hungarian exhibits pro-drop, be-

[1] Only in very specific cases can third person pronouns be dropped in Finnish, such as answers to yes-no questions or when the dropped pronouns is c-commanded by a pronoun that is spelled out (Holmberg 2016).

cause pronouns remained unaligned very often. The MDL approach, however, can (and did) also detect that, e.g., pro-drop is extant in a language, although it is less straightforward to do so. In Chapter 3 it was shown that patterns with a pronoun in it were often absent in Czech while they were present in English and Dutch, strongly suggesting there may be pro-drop in Czech, but this must be deduced from the ranking of the patterns that contain a pronoun tag.

Another example that was already addressed in Chapter 3 and which may cause a linguist to miss that a word or word type is absent in one of the languages under investigation, is that of Ancient Greek and Turkish: whereas Ancient Greek only has definite articles, Turkish only has indefinite articles, which means that in every case that Ancient Greek has an article, Turkish will not have an article, and vice versa. Because definite and indefinite articles are tagged uniformly as DET in Universal Dependencies, and because the MDL approach does not use alignment to count the mismatches of patterns, the linguist may miss that articles exist in a complementary distribution in Ancient Greek and Turkish. The alignment approach is better at detecting this difference, due to it using word alignment and it having access to the subcategory attributes that distinguish definite from indefinite articles. It must however be noticed that the alignment method was designed to work on a language pair in which one of the two languages does not have available annotation tools, and due to the unilateral mapping of linguistic annotation from the source language onto the target language based on word alignments, the user may fail to detect any morpho-syntactic features that concern unaligned words in the target language. For example, let us assume that there are no annotation tools available for Ancient Greek, then the fact that indefinite articles are absent in Ancient Greek can be detected due to the Turkish indefinite articles remaining unaligned, however, because definite articles are absent in Turkish, no linguistic annotations are mapped onto the Ancient Greek definite articles through alignment, leading to the Ancient Greek definite articles being completely absent in the output of the developed tools.

Furthermore, the MDL approach is better at detecting differences in the linear ordering and adjacencies of elements. While the alignment approach does take into consideration the relative order by counting crossing alignments, it only shows the linguist, e.g., that an auxiliary verb comes before the main verb in Dutch (in main clauses), but it shows only very indirectly that there may be interfering material, such as adverbials or an object. The linguist may therefore miss the difference with English, where the possibility of intervening material between the auxiliary and the main verb is highly restricted. Related to this weakness of the alignment approach is that it was missed in Chapter 4 that Hungarian demonstrative pronouns must be directly followed by a definite pronoun. It was already suggested in that Chapter that the tools should consider adjacencies of words, so that these types of collocations in the target language can be discovered, however this will likely not solve the issue with interfering material.

The alignment approach has the advantage over the MDL approach that

it also considers word forms, which makes it possible to deploy the developed AAA tool, designed to detect potential affixes in the target language and to associate them to attributes of the annotated source language. It also makes it possible to detect morphological properties from the output of the other tools, for instance in Chapter 4 the fact that Hungarian has grammatical case.

All these discussed differences follow from the difference between the two approaches in information input and the way in which it is processed. An interesting line for future research would be to adapt the MDL approach to process syntactic trees in a multivariate way. Instead of linear POS sequences, it would then extract patterns that are parts of syntactic trees, in which nodes (i.e. words) contain multiple channels of annotation, with the preferable possibility of gapping over words akin to SQS, although it would require the existence of parsers for both languages, making it less broadly deployable. It is currently also unclear how this could be implemented and whether it would be computationally feasible.

It may furthermore be valuable to briefly discuss the difference in complexity of the outputs. While allowing for gaps in the patterns intuitively makes it easier to map differences in e.g. the use of articles, it was observed that gaps can make interpretation very complicated. Because the SQS algorithm used allows that the number of elements skipped over be strictly one less than the length of the pattern under consideration, it becomes increasingly difficult to understand a pattern as it grows in length. A pattern consisting of nine tags, such as PUNCT DET NOUN AUX ADP NUM NOUN VERB PUNCT, found in the English-Dutch run in Chapter 3, may have skipped over eight other tags, such as an adjective, an adverb or a verb, making it hard for the linguist to translate this sequence into something meaningful from which to derive a hypothesis on syntactic differences.

Apart from the difficulties that may arise from gapping, the MDL output is much more straightforward than the output of the alignment approach. Whereas the MDL approach ranks its output on relevance, the alignment approach does not, leaving the linguist to fully interpret the data by themselves, which may demand more practice.

So, from the point of view of the user and the usability of the tools, the choice between the MDL approach and the alignment approach is a trade-off between richer, more detailed annotation and therefore more detailed differences found on the one hand, and a much more complex interpretation of the output of the tools on the other. MDL more easily guides the linguist where to investigate further, whereas the alignment approach requires more input from the linguist to generalize and to find directions for further investigation.

5.4 General observations and findings

Over the entirety of this dissertation, some more general observations were made. In this Section I will discuss several findings that come to light when

comparing all three Chapters together.

5.4.1 On tagging and automatic annotation

An important observation that was made concerns the quality of tagging and parsing. The tools from all Chapters were shown to be very sensitive to tagging accuracy and consistency. As mentioned, a tagging error may push a sentence pair over the threshold and cause the filters from Chapter 2 to discard it wrongly. This is because a tagging error constitutes a higher edit distance, and because the edit distance is a discrete integer value, there is not much room for small errors.[2]

Chapter 3, too, saw a strong influence of tagging errors on the results, because of a ripple effect down the line. A tagging error causes a distortion in the frequency of a pattern, causing it to compress the data less well and reducing the chance for it to be mined by SQS. A distortion in the frequency of a pattern due to a tagging error also distorts the frequencies of the mismatches, which are crucial in the ranking of the differences, and may cause the difference to be ranked much lower than it should have been, and to be missed by the linguist.

Tagging errors also cause issues for the tools of Chapter 4. Because the alignment approach uses so much annotation – not only POS tags – the chances of one of the attributes to be incorrect goes up. This causes the output to be very noisy, which may cause syntactic differences to be missed, partly because the noisiness raises the necessity for suppressing the output.

Tagging inconsistencies, as opposed to tagging errors, also raised issues in a similar way for the filters and the MDL approach. Whereas a tagging error is the assigning of a wrong label, a tagging inconsistency is the assigning of a label that is justified within the grammar of a language, but not between two languages. If the two languages under investigation have even slightly different annotation guidelines, a NOUN tag in the one language may not fully correspond to a NOUN tag in the other, which will lead to more mismatching occurrences and consequently to patterns with a high χ^2 value that in fact do not indicate a syntactic difference. As pointed out in Chapter 3, we found that in English many more words were tagged as PROPN than in Dutch and Czech, despite having clear nominal or adjectival morpho-syntactic properties and the direct translations in the latter two languages were often tagged as nouns or adjectives, capitalized or not. Although it may be true and solidly justified to have the words be tagged as proper nouns in a language's linguistic tradition, this inconsistency led to the MDL approach finding many syntactic differences between English and the other two languages that arguably do not signify true differences in the syntactic potential of the languages in question. While it was observed that Universal Dependencies guidelines may not always

[2] Of course, this does not hold true for the sentence-length filter, because it does not use syntactic information. A tagging inaccuracy therefore has no effect on its results.

be as consistent throughout languages as desired, the contribution that Universal Dependencies have made to the universalization of annotation guidelines throughout languages and therefore the possibility to more efficiently compare languages to one another cannot be denied and has proven vital in this dissertation and beyond.

5.4.2 Corpus choice

On the matter of corpus genre, it was observed that both the Europarl corpus and the Bible, used throughout this dissertation, were rather particular in their language use. The Europarl corpus shows a very high average sentence length and frequent formulaic utterances common for language used in Parliament, and the Bible shows many archaisms, distinguishing both corpora from day-to-day language. The result is that certain constructions are overrepresented in the data while others are underrepresented. Despite their shown effectiveness in the detection of syntactic differences, the tools developed in Chapters 3 and 4 were therefore not able to detect every difference between the languages under investigation. Of course, corpus choice and the genre of the corpus are crucial in any natural-language processing task, as was also pointed out by Wälchli (2007), which was extensively discussed in Chapter 1. As a result, one of the conclusions of this dissertation is that corpus choice influences the results of the automatic detection of syntactic differences, and that a potential user of the tools must be aware of the possibility of syntactic differences being missed.

As for corpus size, it is difficult to say how large a dataset should be in order to be able to successfully detect syntactic differences automatically from it. Chapter 4 generally describes good results, although some characteristic differences between English and Hungarian were not found, but the Bible, with a version containing 28,972 verses used in this dissertation,[3] is considered to be a relatively small corpus and one could expect to be able to detect the missed differences using a larger corpus. However, the data used in Chapter 3 were much smaller (only 10,000 sentence pairs)[4] – especially after filtering out syntactically incomparable sentence pairs which saw a reduction of the data to one fifth to one sixth of the original number of sentence pairs – and good results and meaningful hypotheses were nonetheless obtained. The influence of corpus size was all in all not strongly noticed: the reduction of corpus size due to the filter only marginally influenced the results, and no differences between the MDL approach and the alignment approach could be traced back to a difference in corpus size. This is in line with Sanders (2007), who showed that the size of the data can be reduced in comparison to Nerbonne and Wiersma (2006), and can in fact be relatively small in order to be able still obtain significant results. Sanders (2007) argued that there may be a lower limit to the data size of around 250,000 words (for his method, at least). However, during the MDL

[3] Containing around 850,000 tokens for English and 680,000 for Hungarian.
[4] Containing around 220,000 tokens for English, 225,000 for Dutch and 190,000 for Czech.

experiments with the filter many fewer words were used (between 9,000 and 17,000, depending on the language pair), but results were still significant.

In fact, using very large corpora may not be advisable. This is not only because it may make the interpretation of the results of the MDL approach and especially the alignment approach even more complicated, but mostly because the algorithms of the filter, the MDL approach and the alignment approach are computationally complex. As for the filter, especially the GED-based filter is computationally complex, given that it was proven that calculating the exact GED is NP-hard (Zeng et al. 2009) and that the problem is even APX, meaning that it is hard to approximate as well (Lin 1994). The MDL approach is computationally expensive due to its relying on the SQS algorithm, the complexity of which can grow cubically with the size of the data, although in practice it is much faster (Tatti and Vreeken 2012). Finally, the alignment approach also suffers from data size limitations, especially the GTI, which produces a massive output as a result of iterative nesting, and the AAA, which has a looming danger of combinatorial complexity (a growth curve even worse than exponential). There may therefore be an upper limit to the size of the data that can be used with the tools developed for the purposes of this dissertation, however it is hard to determine this limit.[5]

The use of parallel corpora was shown to be of added value to the automatic detection of syntactic differences. Although Wiersma, Nerbonne and Lauttamus (2011) already successfully extracted syntactic differences from non-parallel corpora, the use of parallel corpora allowed us to identify in which contexts the differences occur, and even to generate hypotheses on syntactic differences between an annotated language and an unannotated language with the help of alignment (which is only possible in parallel corpora). The MDL approach, the way it is designed in this dissertation, also relies on parallel corpora, because it counts the mismatches of patterns between sentence pairs, which allows for more precise frequencies and circumvents the need for a complex statistical test to mitigate for non-parallellity – although an adaptation to the algorithm could probably be devised so that it works on non-parallel data, too.

Because of the way the tools were designed, I did not compare results from experiments with parallel data with results from experiments with non-parallel data, although differences with the results from others were discussed in the previous Chapters (chiefly among which Nerbonne and Wiersma 2006; Wiersma, Nerbonne and Lauttamus 2011). Wälchli (2007) already extensively argued for the use of parallel corpora, as discussed in Chapter 1. To add to this discussion, it is most desirable to use very homogenous data when trying to detect syntactic differences between languages, so that any variation found between

[5] I also firmly believe that the complexity of the algorithms and the size of the data should be considered more often in academia, because the carbon footprint of complex calculations is much higher than people realize. My colleague dr. Alex Brandsen already noted that the carbon footprint from the GPU usage during his PhD research was equivalent to that of a flight from Amsterdam to Prague, and that less computationally expensive methods are therefore preferable (Brandsen 2022: proposition no. 7).

the languages can be traced back to the syntactic variation. The use of a parallel corpus removes unwanted sources of variation, such as variation in speaker, genre, and text length, making it ideal for the purposes of comparative-syntactic research.

One point of concern regarding the use of corpora (at all, both parallel and non-parallel) is that it has a confirmation bias, because in general they only contain correct utterances, while in comparative-syntactic research it can be very insightful to have a few ungrammatical sentences at one's disposal,[6] especially when access to large datasets is limited: the range and limits of syntactic variation are not merely defined by what can be said, but also by what cannot be said. The tools developed for and presented in this dissertation should therefore always be considered as complementary to traditional comparative-syntactic research.

5.4.3 Some remarks on future research

In the task of automatically detecting morpho-syntactic differences between languages, it is important that the output of the algorithm, as well as the algorithm itself, are transparent and interpretable for the human linguist, so that phenomena can be researched more closely, cross-linguistic theories on syntactic variation can more easily be formulated and the research remains replicable and reproducible. While the interpretability of the algorithm and its transparency are known problems for deep learning approaches, the future may hold more direct applications of deep learning in the task of automatically detecting syntactic differences,[7] especially in light of the more recent developments concerning the opening of the 'black boxes' that deep learning models are famous for. The architecture of a more deep-learning driven approach to detecting syntactic differences, though, remains unclear. Ideally a transparent and interpretable unsupervised deep-learning method will be deployed, in which the output is not restricted to predefined labels and syntactic differences can be detected that were hitherto unknown.

A more clear future for machine learning approaches can be seen when labels for morpho-syntactic properties of languages or language varieties are already available, in which case the properties can be used to cluster languages based on syntactic 'behaviour' so as to cluster languages on their phylogenetic relationship (cf. Spruit 2008, who clustered Dutch dialects based on discrete syntactic properties), or to detect associations and correlations between the properties so as to reduce them to fewer overarching syntactic properties or phenomena (cf. e.g. also Spruit 2008, as well as Van Craenenbroeck, Koppen

[6] These ungrammatical sentences have usually been very carefully selected or in fact, in most cases, been constructed.

[7] This dissertation already saw the use of deep learning methods in less direct ways, namely in the preparation of the data. UDPipe, for instance, uses models that are trained using deep learning algorithms, but the transparency of the tools for data preparation were deemed to be of less importance than of the algorithms that detect the differences.

and Bosch 2019).

As for the future of the influence of the human linguist in the process of automatically detecting syntactic differences, I think it can be stated that the human linguist cannot be removed from the equation. As already said in Chapter 4, I believe that a good balance can be struck between the freedom for subjective interpretation on the one hand and the more computer-driven generation of hypotheses on the other, though whatever the tendency in the balance struck, the expertise and subjective interpretation of the linguist will always be there: either the linguistic bias will be present in the interpretation of the output, or the linguistic bias will be put in the design of the algorithm.

The question of what this balance should look like is interesting, however. In Chapter 3 and 4 it was already seen that the approaches require drastically different inputs from the linguist: while the difficulty with MDL mostly resided in the interpretation of the longer patterns and specifying (as opposed to generalizing over) differences by going back to the data, the difficulty with the alignment approach mostly resided in making sense of zero-alignment frequencies, crossings and other annotations and generalizing over several differences that cover one larger phenomenon. The latter of the approaches required more practice, and in the future the linguist could definitely benefit from a better user interface. It would even be possible to have the linguist interact with the algorithm during the process.

5.5 Conclusion

Relating this all back to the research question of whether it is possible to automatically detect syntactic differences and, if so, how, it was shown that correct hypotheses on syntactic differences between languages can be generated from parallel corpora through the use of the minimum description length principle, counting mismatches between part-of-speech pattern occurrences, word alignment and mapping annotation from an annotated language onto another unannotated language. The automatic detection of syntactic differences between languages is therefore possible, yes. The tools developed for the purposes of this research work well and can aid a linguist significantly in their search for differences or similarities. However, it was also seen that the tools do not work perfectly, for instance hampered by the quality of the data and annotations, and the process may, for now, not be as detailed, automatized or objective as one would wish, leaving much room for future endeavours.

Bibliography

Aarts, Flor G. A. M. and Herman Chr. Wekker (1987). *A contrastive grammar of English and Dutch: Contrastieve grammatica Engels / Nederlands*. Dordrecht: Springer. DOI: https://doi.org/10.1007/978-94-017-4984-8.

Abu-Aisheh, Zeina et al. (2015). 'An exact graph edit distance algorithm for solving pattern recognition problems'. In: *4th International Conference on Pattern Recognition Applications and Methods 2015*.

Abzianidze, Lasha et al. (2017). 'The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations'. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 242–247. URL: https://aclanthology.org/E17-2039.

Agić, Željko and Ivan Vulić (2019). 'JW300: A wide-coverage parallel corpus for low-resource languages'. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3204–3210. DOI: 10.18653/v1/P19-1310. URL: https://aclanthology.org/P19-1310.

Babická, Blanka et al. (2008). 'The passive voice in English and Czech and some implications for teaching'. In: *Discourse and Interaction* 1.2, pp. 19–30.

Barbiers, Sjef (2009). 'Locus and limits of syntactic microvariation'. In: *Lingua* 119.11, pp. 1607–1623.

Barbiers, Sjef et al. (2005/2008). *Syntactic atlas of the Dutch dialects*. 2 vols. Amsterdam University Press.

Bard, Gregory V. (2007). 'Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric'. In: *Proceedings of the fifth Australasian symposium on ACSW frontiers – Volume 68*. Citeseer, pp. 117–124.

Barron, Andrew, Jorma Rissanen and Bin Yu (1998). 'The minimum description length principle in coding and modeling'. In: *IEEE Transactions on Information Theory* 44.6, pp. 2743–2760.

Benjamini, Yoav and Yosef Hochberg (1995). 'Controlling the false discovery rate: A practical and powerful approach to multiple testing'. In: *Journal of the Royal statistical society: Series B (Methodological)* 57.1, pp. 289–300.

Bertens, Roel, Jilles Vreeken and Arno Siebes (2016). 'Keeping it short and simple: Summarising complex event sequences with multivariate patterns'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 735–744.

Berwick, Robert C. and Noam Chomsky (2016). *Why only us: Language and evolution*. Cambridge, MA: MIT press.

Bonferroni, Carlo (1936). 'Teoria statistica delle classi e calcolo delle probabilità'. In: *Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8, pp. 3–62.

Brandsen, Alex (2022). 'Digging in documents: using text mining to access the hidden knowledge in Dutch archaeological excavation reports'. PhD thesis. Leiden University. URL: `https://hdl.handle.net/1887/3274287`.

Brill, Eric (1992). 'A simple rule-based part of speech tagger'. In: *HLT*.

Broekhuis, Hans (2020). *R-pronominalization and R-words*. Retrieved October 14, 2020 from `https://www.taalportaal.org/taalportaal/topic/link/syntax__Dutch__adp__adp5__P5_strand.xml`.

Brown, Peter F. et al. (1993). 'The mathematics of statistical machine translation'. In: *Computational Linguistics* 19.2, pp. 263–313. URL: `http://acl.ldc.upenn.edu/J/J93/J93-2003.pdf`.

Christodoulopoulos, Christos and Mark Steedman (2015). 'A massively parallel corpus: The Bible in 100 languages'. In: *Language resources and evaluation* 49.2, pp. 375–395.

Cinque, Guglielmo (1999). *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press.

Cinque, Guglielmo and Richard S. Kayne, eds. (2005). *The Oxford handbook of comparative syntax*. Oxford University Press.

Cohen, Jacob (1960). 'A coefficient of agreement for nominal scales'. In: *Educational and Psychological Measurement* 20.1, pp. 37–46.

Cysouw, Michael (2010). 'Semantic maps as metrics on meaning'. In: *Linguistic Discovery* 8.1, pp. 70–95.

Cysouw, Michael and Bernhard Wälchli (2007). 'Parallel texts: Using translational equivalents in linguistic typology'. In: *Language typology and universals* 60.2, pp. 95–99.

Dahl, Östen (2007). 'From questionnaires to parallel corpora in typology'. In: *Language Typology and Universals* 60.2, pp. 172–181.

Davies, Mark (2008). *The sorpus of contemporary American English*. URL: `www.english-corpora.org/coca/`.

de Lange, Joke (2004). 'Article omission in child speech and headlines'. In: *Utrecht Institute of Linguistics OTS Yearbook*, pp. 109–119.

Dempster, Arthur P., Nan M. Laird and Donald B. Rubin (1977). 'Maximum likelihood from incomplete data via the EM algorithm'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Dice, Lee R. (1945). 'Measures of the amount of ecologic association between species'. In: *Ecology* 26.3, pp. 297–302.

Donaldson, Bruce (2008). *Dutch: A comprehensive grammar*. 2nd ed. Comprehensive Grammars. Routledge.

Dušková, Libuše (1991). 'The complex sentence in British and Czech grammar'. In: *Brno studies in English* 19.1, pp. 65–75. URL: http://hdl.handle.net/11222.digilib/104417.

Dyer, Chris, Victor Chahuneau and Noah A. Smith (2013). 'A simple, fast, and effective reparameterization of IBM Model 2'. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 644–648.

Eberhard, David M., Gary F. Simons and Charles D. Fennig, eds. (2021). *Ethnologue: Languages of the World*. 24th ed. Dallas, Texas: SIL International.

Fleiss, J. L. and Jacob Cohen (1973). 'The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability'. In: *Educational and Psychological Measurement* 33, pp. 613–619.

Greenberg, Joseph H. (1963). 'Some universals of grammar with particular reference to the order of meaningful elements'. In: *Universals of Language*. Ed. by Joseph H. Greenberg. Cambridge, MA.: MIT Press, pp. 110–113.

Grünwald, Peter D. (2007). *The minimum description length principle*. MIT press.

Hagberg, Aric A., Daniel A. Schult and Pieter J. Swart (2008). 'Exploring network structure, dynamics, and function using NetworkX'. In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Ed. by Gäel Varoquaux, Travis Vaught and Jarrod Millman. Pasadena, CA USA, pp. 11–15.

Haspelmath, Martin (1997). *Indefinite pronouns*. Oxford University Press.

— (2003). 'The geometry of grammatical meaning: Semantic maps and cross-linguistic comparison'. In: *The new psychology of language*. Psychology Press, pp. 217–248.

Hinrichs, Frauke and Jilles Vreeken (2017). 'Characterising the difference and the norm between sequence databases'. In: *Proceedings of the 4th Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP)*.

Holmberg, Anders (2016). 'Null subjects in Finnish and the typology of pro-drop'. In: *Uralic Syntax book project. Cambridge: Cambridge University*.

Jalili Sabet, Masoud et al. (2020). 'SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings'. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, pp. 1627–1643. DOI: 10.18653/v1/2020.findings-emnlp.147. URL: https://aclanthology.org/2020.findings-emnlp.147.

Kiss, Katalin É. (2002). *The syntax of Hungarian*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511755088.

Koehn, Philipp (2005). 'Europarl: A parallel corpus for statistical machine translation'. In: *MT summit*. Vol. 5, pp. 79–86.

Kroon, Martin et al. (2019). 'A filter for syntactically incomparable parallel sentences'. In: *Linguistics in the Netherlands* 36. Ed. by Janine Berns and Elena Tribushinina, pp. 147–161. DOI: https://doi.org/10.1075/avt.00029.kro.

— (2020). 'Detecting syntactic differences automatically using the Minimum Description Length principle'. In: *Computational Linguistics in the Netherlands Journal* 10, pp. 109–127.

Ladányi, Mária (2015). 'Particle verbs in Hungarian'. In: *Word-formation: An international handbook of the languages of Europe*. Ed. by Peter O. Müller et al. Vol. 1. De Gruyter Mouton, pp. 660–672.

Levenshtein, Vladimir I. (1966). 'Binary codes capable of correcting deletions, insertions, and reversals'. In: *Soviet physics doklady* 10.8, pp. 707–710.

Lin, Chih-Long (1994). 'Hardness of approximating graph transformation problem'. In: *Algorithms and Computation*. Ed. by Ding-Zhu Du and Xiang-Sun Zhang. Vol. 843. Lecture Notes in Computer Science. Berlin: Springer, pp. 74–82.

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

Malá, Markéta (2014). *English copular verbs: A contrastive corpus-supported view*. Filozofická fakulta Univerzity Karlovy.

Mårdh, Ingrid (1980). *Headlinese: On the grammar of English front page headlines*. Vol. 58. Liberläromedel/Gleerup.

McNemar, Quinn (1947). 'Note on the sampling error of the difference between correlated proportions or percentages'. In: *Psychometrika* 12.2, pp. 153–157.

Naughton, James (2005). *Czech: An essential grammar*. Essential grammars. Routledge.

Nerbonne, John and Wybo Wiersma (2006). 'A measure of aggregate syntactic distance'. In: *Proceedings of the Workshop on Linguistic Distances*. Association for Computational Linguistics, pp. 82–90.

Nivre, Joakim et al. (2016). 'Universal Dependencies v1: A Multilingual Treebank Collection'. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), pp. 1659–1666. URL: https://aclanthology.org/L16-1262.

Och, Franz Josef and Hermann Ney (2003). 'A Systematic Comparison of Various Statistical Alignment Models'. In: *Computational Linguistics* 29.1, pp. 19–51.

Odijk, Jan et al. (2017). 'The parse and query (PaQu) application'. In: *CLARIN in the Low Countries*. Ed. by Jan Odijk and A. Hessen. London: Ubiquity Press, pp. 281–297. DOI: https://doi.org/10.5334/bbi.23.

Oostdijk, Nelleke et al. (2013). 'The construction of a 500-million-word reference corpus of contemporary written Dutch'. In: *Essential Speech and Language Technology for Dutch: Results by the STEVIN-programme*. Springer Verlag. Chap. 13.

Osborne, Miles (1999a). 'DCG induction using MDL and parsed corpora'. In: *International Conference on Learning Language in Logic*. Springer, pp. 184–198.

— (1999b). 'MDL-based DCG induction for NP identification'. In: *EACL 1999: CoNLL-99 Computational Natural Language Learning*.

Östling, Robert and Jörg Tiedemann (2016). 'Efficient word alignment with Markov Chain Monte Carlo'. In: *Prague Bulletin of Mathematical Linguistics* 106, pp. 125–146. URL: http://ufal.mff.cuni.cz/pbml/106/art-ostling-tiedemann.pdf.

Radford, A. (2004). *English syntax: An introduction*. Cambridge University Press. ISBN: 9780521542753. URL: https://books.google.nl/books?id=LdAi292Q4-0C.

Reback, Jeff et al. (Mar. 2021). *pandas-dev/pandas: Pandas 1.2.3*. Version v1.2.3. DOI: 10.5281/zenodo.4572994. URL: https://doi.org/10.5281/zenodo.4572994.

Rounds, Carol (2009). *Hungarian: An essential grammar*. Routledge.

Sampson, Geoffrey (2000). 'A proposal for improving the measurement of parse accuracy'. In: *International Journal of Corpus Linguistics* 5.1, pp. 53–68.

Sanders, Nathan C. (2007). 'Measuring syntactic difference in British English'. In: *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*. Association for Computational Linguistics, pp. 1–6.

— (2010). 'A statistical method for syntactic dialectometry'. PhD thesis. Indiana University.

Sanguinetti, Manuela and Cristina Bosco (2015). 'PartTUT: The Turin University Parallel Treebank'. In: *Harmonization and development of resources and tools for Italian natural language processing within the PARLI project*. Springer, pp. 51–69.

Shannon, Claude E. (1948). 'A mathematical theory of communication'. In: *The Bell system technical journal* 27.3, pp. 379–423.

Spruit, Marco René (2008). 'Quantitative perspectives on syntactic variation in Dutch dialects'. PhD thesis. University of Amsterdam.

Straka, Milan and Jana Straková (2017). 'Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe'. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver, Canada: Association for Computational Linguistics, pp. 88–99. URL: http://www.aclweb.org/anthology/K/K17/K17-3009.pdf.

Tatti, Nikolaj and Jilles Vreeken (2012). 'The long and the short of it: Summarising event sequences with serial episodes'. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 462–470.

Taylor, Ann, Mitchell Marcus and Beatrice Santorini (2003). 'The Penn treebank: An overview'. In: *Treebanks*, pp. 5–22.

Tiedemann, Jörg (2011). 'Bitext alignment'. In: *Synthesis Lectures on Human Language Technologies* 4.2, pp. 1–165.

— (2012). 'Parallel data, tools and interfaces in OPUS'. In: *LREC*. Vol. 2012. Citeseer, pp. 2214–2218.

Toutanova, Kristina et al. (2003). 'Feature-rich part-of-speech tagging with a cyclic dependency network'. In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 252–259.

Van Craenenbroeck, Jeroen, Marjo van Koppen and Antal van den Bosch (2019). 'A quantitative-theoretical analysis of syntactic microvariation: Word order in Dutch verb clusters'. In: *Language* 95.2, pp. 333–370.

van den Bosch, Antal et al. (2007). 'An efficient memory-based morphosyntactic tagger and parser for Dutch'. In: *LOT Occasional Series* 7, pp. 191–206.

van der Klis, Martijn, Bert Le Bruyn and Henriette De Swart (2017). 'Mapping the PERFECT via translation mining'. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Vol. 2, pp. 497–502.

Wälchli, Bernhard (2007). 'Advantages and disadvantages of using parallel texts in typological investigations'. In: *Language Typology and Universals* 60.2, pp. 118–134.

— (2010). 'Similarity semantics and building probabilistic semantic maps from parallel texts'. In: *Linguistic Discovery* 8.1, pp. 331–371.

Weir, Andrew (2009). 'Article drop in English headlinese'. MA thesis. University College London.

Wiersma, Wybo, John Nerbonne and Timo Lauttamus (2011). 'Automatically extracting typical syntactic differences from corpora'. In: *Literary and Linguistic Computing* 26.1, pp. 107–124.

Wong, Tak-sum et al. (2017). 'Quantitative comparative syntax on the Cantonese-Mandarin parallel dependency treebank'. In: *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*. Pisa,Italy: Linköping University Electronic Press, pp. 266–275. URL: https://www.aclweb.org/anthology/W17-6530.

Youden, William J. (1950). 'Index for rating diagnostic tests'. In: *Cancer* 3.1, pp. 32–35.

Zeng, Zhiping et al. (2009). 'Comparing stars: On approximating graph edit distance'. In: *Proceedings of the VLDB Endowment* 2.1, pp. 25–36.

Zwart, Jan-Wouter (2011). *The syntax of Dutch*. Cambridge Syntax Guides. Cambridge University Press. DOI: 10.1017/CBO9780511977763.

Overview of URLs to used, referenced and developed tools and datasets

Detecting Syntactic Differences Automatically — *This dissertation*

General link:
`https://github.com/mskroon/DeSDA`

Bible corpus — (Christodoulopoulos and Steedman 2015)

General link:
`https://github.com/christos-c/bible-corpus`

DITTO — (Bertens, Vreeken and Siebes 2016)

General link:
`http://eda.mmci.uni-saarland.de/prj/ditto`

`eflomal` — (Östling and Tiedemann 2016)

General link:
`https://github.com/robertostling/eflomal`

Europarl v7 corpus — (Koehn 2005)

General link:
`https://www.statmt.org/europarl`

`fast_align` (Dyer, Chahuneau and Smith 2013)

General link:

`https://github.com/clab/fast_align`

Frog tagger (van den Bosch et al. 2007)

General link:

`http://languagemachines.github.io/frog`

GIZA++ (Och and Ney 2003)

General link:

`https://www.statmt.org/moses/giza/GIZA++.html`

`networkx` (Hagberg, Schult and Swart 2008)

General link:

`https://networkx.org`

Download:

`https://pypi.org/project/networkx`

Opus corpus (including Europarl v7) (Tiedemann 2012)

General link:

`https://opus.nlpl.eu`

`pandas` (Reback et al. 2021)

General link:

`https://pandas.pydata.org`

Download:

`https://pypi.org/project/pandas`

SimAlign (Jalili Sabet et al. 2020)

General link:

`https://github.com/cisnlp/simalign`

SQS (Tatti and Vreeken 2012)

General link:

`http://adrem.uantwerpen.be/sqs`

Stanford tagger (Toutanova et al. 2003)

General link:

`https://nlp.stanford.edu/software/tagger.shtml`

UDPipe (Straka and Straková 2017)

General link:

`https://ufal.mff.cuni.cz/udpipe`

Tool:

`https://github.com/ufal/udpipe`

Models (from 15 Nov 2018; used in Chapters 2 and 3):

`https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2898`

Model (English ParTUT; used in Chapter 4):

`https://github.com/UniversalDependencies/UD_English-ParTUT`

Universal Dependencies (Nivre et al. 2016)

General link:

`https://universaldependencies.org`

Samenvatting

De zinsbouw – of syntaxis – van natuurlijke taal is een systeem van combinatorische regels waarmee uit woorden en morfemen[1] complexe hiërarchische structuren worden gebouwd, zoals zinnen en woordgroepen. Het inzicht dat de woorden in een zin niet alleen lineair maar ook hiërarchisch zijn geordend, namelijk als woordgroepen die woordgroepen bevatten, staat centraal in de moderne taalwetenschap (cf., e.g., Berwick and Chomsky 2016).

Oppervlakkige vergelijking van verschillende talen lijkt erop te duiden dat hun syntaxis in grote mate verschilt: bijvoorbeeld, variatie in woordvolgorde, variatie in de aan- of afwezigheid van een morfeem, woord of woordgroep, het verdubbelen van grammaticale kenmerken, of variatie in de morfosyntactische expressie van grammaticale relaties zoals congruentie tussen het onderwerp en de persoonsvorm. Niettemin kan men in de syntactische literatuur talloze argumenten vinden voor de hypothese dat alle menselijke talen dezelfde abstracte set syntactische principes delen. Het hoofddoel van theoretisch comparatief-syntactisch onderzoek is om de syntactische variatie tussen natuurlijke talen in kaart te brengen door hun structuren te vergelijken en de syntactische overeenkomsten en verschillen te beschrijven, om vervolgens een alomvattende, taaloverstijgende theorie te kunnen formuleren die deze variatie beschrijft en verklaart (Cinque and Kayne 2005). Het vakgebied probeert antwoorden te vinden op vragen als: wat is een (on)mogelijke natuurlijke taal, welke syntactische eigenschappen zijn universeel en welke zijn taalspecifiek, en is de syntactische variatie een eigenschap van de module van de mentale grammatica die hiërarchische structuren bouwt of is het mogelijk om syntactische variatie te herleiden tot eigenschappen van andere modules van de grammatica, zoals het lexicon of de module die zorgt voor de fonologische *spellout* en linearisatie?

Het is gebruikelijk voor syntactici om hun eigen moedertaal met andere talen te vergelijken door het bestuderen van gedetailleerde grammatica's en

[1] Een morfeem is de kleinste lexicale eenheid die betekenis draagt in taal. Bijvoorbeeld, het woord *draaglijk* bestaat uit de morfemen *draag* en *-lijk*, die zelf niet verder kunnen worden opgedeeld in kleinere, betekenisdragende stukjes.

andere taalkundige literatuur en de (on-)grammaticaliteit van zinnen te toetsen door ze voor te leggen aan vakgenoten en andere proefpersonen. Maar door het enorme aantal natuurlijke talen en dialecten, de hoge mate van variatie die ze vertonen (zelfs tussen nauw verwante talen of dialecten) en het technisch gezien oneindige aantal mogelijke zinnen per taal of dialect die de taalkundige kan onderzoeken is systematische vergelijking een vrijwel onmogelijke opgave.

Het gevolg hiervan is dat syntactici veel verschillen en associaties tussen die verschillen over het hoofd kunnen zien en dat formele beschrijvingen van taal onvolledig blijven. Het vakgebied zou daarom sterk gebaat zijn bij een (gedeeltelijke) automatisering van het proces, omdat dit het onderzoek zou versnellen en het grootschaliger, systematischer en reproduceerbaarder zou maken. Een computer kan veel meer data in veel meer talen veel systematischer verwerken en analyseren, wat het waarschijnlijker maakt dat nieuwe variatie in syntaxis kan worden ontdekt en correlaties tussen variabelen die terug te voeren zijn tot abstractere, onderliggende kenmerken kunnen worden gevonden. De vraag blijft echter: **kunnen syntactische verschillen tussen talen automatisch ontdekt worden en, zo ja, hoe dan?** Dit is de vraag die in dit proefschrift centraal staat.

Hoofdstuk 1 is een uitgebreide inleiding, waarin onder andere de literatuur over dit vraagstuk besproken wordt en de aard van de data wordt beschreven. De data waarmee in dit proefschrift wordt gewerkt bestaan uit parallelle tekstcorpora: grote tekstverzamelingen waarbij elke zin gelinkt is aan een vertaling in een tweede taal. Specifiek wordt gebruik gemaakt van de Bijbel in het Engels, Hongaars en Nederlands, en van een Engels, Nederlands en Tsjechisch fragment van het Europarl corpus (Koehn 2005), wat bestaat uit de vergaderingen van het Europees Parlement, die vertaald worden in alle talen van de Europese Unie. Deze tekstcorpora worden voor de analyse door middel van bestaande algoritmes automatisch syntactisch geannoteerd, waar ze dat nog niet waren.

In Hoofdstuk 2 komt het probleem van syntactisch onvergelijkbare zinsparen aan bod. In parallelle corpora zijn zinsparen op syntactisch niveau lang niet altijd vergelijkbaar: twee zinnen die hetzelfde betekenen kunnen zeer verschillende constructies bevatten of vrije vertalingen betreffen. Wanneer dit soort zinnen worden gebruikt om syntactische verschillen te ontdekken, worden er vele verschillende soorten verschillen gevonden die niet informatief zijn voor de taalkundige. “Vrije” vertalingen moeten daarom verwijderd worden uit de dataset, maar door de omvang van gebruikte datasets is het veelal onmogelijk om dit met de hand te doen. Er is daarom een methode en kwantitatieve maat nodig om zinsparen automatisch uit de dataset te kunnen filteren die niet syntactisch vergelijkbaar zijn.

In dit hoofdstuk worden vier manieren om dit te bewerkstelligen verkend en geëvalueerd met datasets met Engelse, Nederlandse en Duitse parallelle zinsparen. Voor de zinsparen is van tevoren met de hand gedetermineerd of ze wel of niet syntactisch vergelijkbaar zijn. Het eerste filter is gebaseerd op de Levenshteinafstand op POS-tags (woordsoortlabels), een bekend algoritme dat het minimaal aantal bewerkingen berekent dat nodig is om de ene sequentie in

de andere sequentie te veranderen (Levenshtein 1966). Het tweede filter maakt gebruik van de zinslengteratio tussen de twee zinnen onder de aanname dat als een zin significant langer is dan zijn vertaling, het zinspaar waarschijnlijk te sterk van elkaar verschilt of zelfs foutief aan elkaar is verbonden. Het derde filter is gebaseerd op de bewerkingsafstand tussen de syntactische dependentiebomen van de twee zinnen. Deze bewerkingsafstand is equivalent aan de Levenshteinafstand, maar toegepast op hiërarchische structuren in plaats van lineaire sequenties. Het laatste filter combineert de vorige drie filters in een logistisch regressiemodel.

De resultaten van Hoofdstuk 2 laten vooral zien dat filteren op syntactische vergelijkbaarheid een moeilijke opgave is, deels omdat syntactische vergelijkbaarheid lastig te definiëren is. Niettemin zijn de filters bruikbare tools voor de automatische selectie van syntactisch vergelijkbare zinsparen uit een parallel corpus. De beste resultaten kunnen worden behaald met het filter dat gebruik maakt van een logistisch regressiemodel, terwijl de filters die gebruik maken van de Levenshteinafstand en de bewerkingsafstand tussen de syntactische bomen gebruikt kunnen worden met redelijk resultaat.

In Hoofstuk 3 presenteer ik een systematische methode om mogelijke syntactische verschillen te detecteren en hypotheses erover te rangschikken voor verder onderzoek door gebruik te maken van parallelle data en het *Minimum Description Length*-principe (MDL). MDL biedt een elegant paradigma voor het ontdekken van structuur in data. Het formaliseert het idee dat elke regelmatigheid in de data kan worden gebruikt om de data te comprimeren (among others Grünwald 2007; Barron, Rissanen and Yu 1998). Deze regelmatigheden kunnen dan worden beschouwd als karakteristieke bouwstenen onderliggend aan de data. Ik maak hierbij gebruik van het SQS-algoritme ('Summarising event seQuenceS'; Tatti and Vreeken 2012) – een algoritme ontwikkeld om patronen in sequentiële data te ontdekken met MDL – om 'typische' sequenties van POS-tags te *minen* voor elke taal die wordt onderzocht. SQS produceert inderdaad lijsten met daarin verwachte patronen van POS-tags die men als karakteristiek voor een taal zou beschouwen. Uit deze lijsten wordt een lijst van mogelijke syntactische verschillen geproduceerd op basis van het aantal parallelle zinnen waar een patroon voorkomt in de ene taal maar niet in de andere. Met behulp van een statistische test worden dan hypotheses gegenereerd over waar er syntactische verschillen kunnen worden gevonden tussen het taalpaar in kwestie. In het hoofdstuk wordt de methode toegepast op parallelle corpora van het Engels, Nederlands en Tsjechisch en ik onderzoek het effect van het filter van Hoofdstuk 2 op de resultaten. De resultaten laten zien dat de methode veelbelovend is in zowel het *minen* van karakteristieke bouwstenen van een taal, alsook het ontdekken van bruikbare syntactische verschillen tussen talen.

Waar de methode van Hoofdstuk 3 aanneemt dat er POS-taggers (programma's die automatisch woorden voorzien van een woordsoortlabel) beschikbaar zijn voor beide talen die worden onderzocht en dat beide talen zijn geannoteerd met dezelfde set labels en volgende dezelfde conventies, is dit niet

altijd het geval. Sterker nog, hoewel het Universal Dependencies-programma (UD; Nivre et al. 2016) streeft naar consistente tagging en annotatie van syntactische dependentiebomen tussen talen,[2] kunnen de richtlijnen van taal tot taal significant verschillen (waarvoor altijd goed onderbouwde redenen zijn).

In Hoofdstuk 4 wordt daarom een andere methode onderzocht om syntactische verschillen te ontdekken, die niet afhankelijk is van de beschikbaarheid van annotatietools voor beide talen. De hoofdvraag van het hoofdstuk is of het mogelijk is om in parallelle tekst volledig geannoteerde tekst in de ene taal (die we de brontaal noemen) te gebruiken om grammaticale eigenschappen van een andere, minder goed beschreven taal (die we de doeltaal noemen) te ontdekken, en verschillen tussen de twee talen.

Hiertoe wordt gebruik gemaakt van *word alignment*, het automatisch oplijnen van woorden die elkaars vertaling zijn binnen twee zinnen. Aan de hand van *word alignment* wordt de annotatie van woorden van de brontaal op woorden van de doeltaal geprojecteerd, met het doel om syntactische eigenschappen van de doeltaal en verschillen tussen de bron- en doeltaal in kaart te brengen door deze projecties semi-automatisch te analyseren. Er zijn drie algoritmes ontwikkeld om de met *word alignment* opgelijnde data te analyseren: de Data Grouper for Attribute Exploration (DGAE), waarmee handige overzichten worden gegeven van de frequentie van annotaties en eigenschappen binnen groepen woorden; de Generalization Tree Inducer (GTI), waarmee de data wordt gestructureerd op basis van de entropie van de annotaties in een poging om te generaliseren over woordklassen; en de Affix-Attribute Associator (AAA), waarmee hypotheses worden gegenereerd over welke tekenreeksen, of *strings*, mogelijk affixen zijn in de doeltaal door ze te associëren met morfosyntactische eigenschappen van woorden in de brontaal. Deze drie tools zijn geëvalueerd op het taalpaar Engels-Hongaars. Zonder enige kennis te hebben van het Hongaars heb ik de tools gebruikt om 43 hypotheses te vormen aangaande morfosyntactische eigenschappen van het Hongaars of verschillen met het Engels. Deze hypotheses zijn onafhankelijk gecontroleerd door een moedertaalspreker en een expert van het Hongaars en zijn syntaxis en zijn getoetst aan een lijst van karakteristieke verschillen tussen het Hongaars en het Engels die van tevoren onafhankelijk door dezelfde expert was samengesteld. De conclusie luidt dat de tools zeer effectief gebruikt kunnen worden om veel correcte hypotheses te vormen over verschillen tussen de talen, verspreid over meerdere syntactische domeinen. Met behulp van de tools heb ik zelfs twee hypotheses gevormd waarvan het vooralsnog onbekend is of ze correct zijn of niet, wat de kracht van de tools in de zoektocht naar syntactische verschillen tussen talen louter onderstreept.

De dissertatie wordt afgesloten met een uitgebreide discussie in Hoofdstuk 5, waarin alle observaties van de voorgaande hoofdstukken bijeengebracht worden en aan elkaar worden verbonden, hetgeen leidt tot nieuwe, overkoepelende observaties en conclusies. Daarin is de belangrijkste conclusie dat het

[2] `universaldependencies.org`

mogelijk is om automatisch syntactische verschillen te ontdekken. De tools die zijn ontwikkeld in het kader van dit onderzoek werken goed en kunnen een taalkundige aanzienlijk helpen in de zoektocht naar verschillen of overeenkomsten. Niettemin werken de tools niet perfect en zijn ze bijvoorbeeld afhankelijk van de kwaliteit van de data en de annotaties: het proces is daarom, vooralsnog, wellicht niet zo gedetailleerd, geautomatiseerd of objectief als men zou willen, maar de tools bieden een goed uitgangspunt voor vervolgonderzoek.

Curriculum vitae

Martin Kroon werd op 6 november 1993 geboren te Groningen waar hij vanaf 2005 onderwijs genoot aan het Praedinius Gymnasium. In 2011 deed hij eindexamen in de richting Natuur en Techniek met wiskunde D en zeven talen waaronder Grieks, Latijn en Russisch. Vervolgens begon hij aan de Rijksuniversiteit Groningen de studie Taalwetenschap. Daarnaast studeerde hij in het eerste jaar Griekse en Latijnse Taal en Cultuur en volgde hij over de gehele periode van zijn bachelor meerdere bijvakken in onder andere Sanskriet, Zweeds en Fins. Na zijn bachelorscriptie over de typologie van polysyndetische structuren vervolgde hij in 2014 zijn studie met het Erasmus Mundus European Masters Program in Language and Communication Technologies aan de Rijksuniversiteit Groningen en de Universiteit van Lotharingen in Nancy, Frankrijk, die hij afsloot met een masterscriptie over de automatische detectie van cognaten en het extraheren van synchrone transitieregels tussen talen. In april 2017 begon hij zijn PhD-onderzoek naar de automatische detectie van syntactische verschillen tussen talen aan de Universiteit Leiden. Het resultaat van dit onderzoek ligt thans voor u. Ten tijde van schrijven is hij verbonden aan de Universiteit Leiden als docent en postdoc betrokken bij CLARIAH.