Towards the automatic detection of syntactic differences

Kroon, M.S.

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.

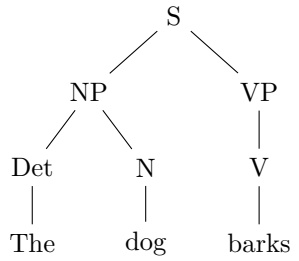


Figure 1.1: A syntactic (constituency) tree of the sentence *The dog barks* after Sanders (2007). From it, Sanders extracts leaf-ancestor paths *S-NP-Det-The*, *S-NP-N-dog* and *S-VP-V-barks*.

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]

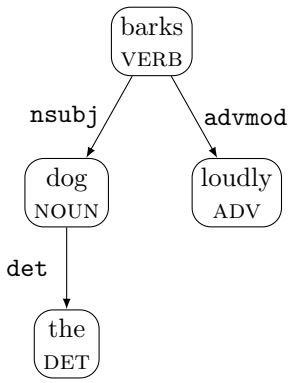


Figure 1.3: A dependency tree of *The dog barks loudly.*

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):

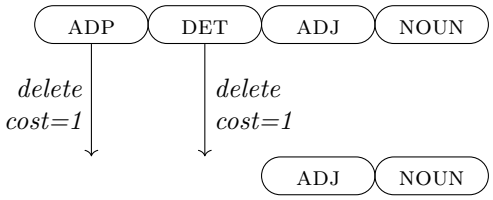
EN: "On the subject at hand, I think that the people of Europe must … "

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]

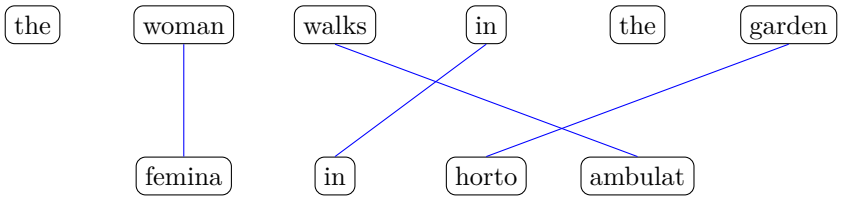


Figure 1.5: An example of word alignment in an English-Latin sentence pair. Word alignments are indicated in blue.

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.