Towards the automatic detection of syntactic differences

Kroon, M.S.

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`

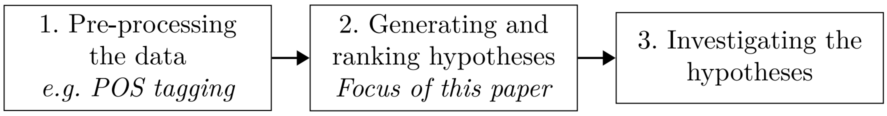
1. Pre-processing the data
e.g. POS tagging
2. Generating and ranking hypotheses
Focus of this paper
3. Investigating the hypotheses

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.

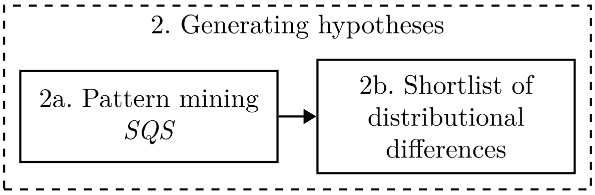

Figure 3.2: Schematic overview of hypothesis generating mechanism.

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 $\mapsto$ A	E F D
DET NOUN $\mapsto$ B	B G A
VERB PRON $\mapsto$ C	E C A
DET ADJ NOUN $\mapsto$ D	B F H C
PRON $\mapsto$ E	D G B
AUX $\mapsto$ F	B A F C
VERB $\mapsto$ G	B F C A
ADV $\mapsto$ 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 morphosyntactic 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 singular 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.