Chapter 2

How to measure syntactic diversity: Patternization, methods, algorithms

Alexander Pfaff
University of Stuttgart

This chapter develops an approach to diagnosing, comparing, and measuring word order variation in a systematic fashion, attempting to put numbers on the degrees of that variation – in isolation and in comparison. Moreover, it explores some ways of giving these numbers a graphical realization thus visualizing syntactic diversity. Since it operates on strings of syntactic categories referred to as *patterns*, the method itself will be labelled *Patternization*. Patternization is a purely mathematical approach based on some simple combinatorial and statistical notions, and presupposes an annotated corpus (minimally, part-of-speech tagging). For illustration, the discussion is primarily based on the NPEGL annotation system and the databases of Old Germanic noun phrases in NPEGL, but the methodology described here as such is intended to be applicable more generally.

1 Introduction

When comparing noun phrases in two languages such as, say, Spanish and modern German, one noticeable feature is the position of adjectives relative to their head noun: *un coche* ***rojo*** – *ein* ***rotes*** *Auto* 'a red car'. Thus when studying (word order) variation in the noun phrase, the positioning of certain elements is a useful point of departure.

In a pilot study leading up to the NPEGL project (Bech et al. 2024 [this volume]), the prenominal vs. postnominal distribution of a range of modifier elements in some Old Germanic languages was examined. Table 1 illustrates the

Alexander Pfaff. 2024. How to measure syntactic diversity: Patternization, methods, algorithms. In Kristin Bech & Alexander Pfaff (eds.), *Noun phrases in early Germanic languages*, 33–70. Berlin: Language Science Press. DOI: 10.5281/zenodo.10641185

positions of adjectives and possessives in relation to the noun (relative frequencies).[1]

Table 1: Modifier–noun order in Old English, Old High German, Old Icelandic, and Old Saxon (Bech et al. 2024: 82, Table 2 [this volume])

	Old English	Old High German	Old Icelandic	Old Saxon
ADJ–N	96.6%	81.7%	86.9%	81.3%
N–ADJ	3.4%	18.3%	13.1%	18.7%
...				
POSS–N	99.7%	82.0%	30.5%	93.7%
N–POSS	0.3%	18.0%	69.5%	6.3%
...				

Such a procedure puts numbers on the preference of a given kind of modifier to occur either in pre- or postnominal position, and these numbers can be seen a measurement of diversity. While this sort of binary approach is clearly an important first step and a widely used method, it is limited in scope. For one thing, it reveals a certain bias – justified though it may be – in that the categories to be compared are pre-determined. In a relevant sense, it is not exhaustive. Secondly, it is not very flexible in that it focuses on one binary parameter (pre- vs. postnominal) for one variable category. Thirdly, and related to the previous point, potential co-dependencies are not captured.

Relying on a number of computational methods, this chapter attempts to develop a more sophisticated and systematic approach to diagnosing, measuring and visualizing word order variation. In the remainder of this section, I will provide some information about the source material/NPEGL, and establish some technical background. Notably, I will define the central component of the approach to be developed here: the *Pattern*. Section 2 introduces the numbers of the current NPEGL entries that will be the basis for further discussion; in addition, a simple measurement for diversity is presented. In Section 3, a more subtle method to explore diversity is developed. I will show how potential permutations of category labels can be related to actual attestations of noun phrase

[1]One output of the project *Constraints on syntactic variation: Noun phrases in early Germanic languages* (NPEGL), led by Kristin Bech, is the creation of an annotated noun phrase database comprising material from Old Icelandic, Old English, Old Saxon, Old Swedish, Old High German and Gothic. For an overview and discussion, the reader is explicitly referred to Pfaff & Bouma (2024 [this volume]); relevant details are briefly discussed in Section 1.1 below.

patterns, and how this allows us to measure the degree of variation as well as the limitations of that variation. Section 4 discusses some macro specifications of "patterns" and shows how these can be used to probe for certain correlations between two categories. A somewhat different perspective is taken in Section 5, where I sketch a probabilistic model to describe the distribution of categories in the nominal space. I will also explore a possibility to visualize that probabilistic distribution. Section 6 concludes. In addition, there is an appendix briefly describing some Python methods that underlie the procedures discussed in this chapter.

1.1 The NPEGL database(s): Category labels and restrictions

Technically speaking, NPEGL is not one database, but a collection of databases (for Old Icelandic, Old English, Old Saxon, etc.) that are all based on the same annotation system. This system employs flat annotation, i.e. it essentially encodes linearity, but not dependency or constituency. On the other hand, by definition, every database entry is a constituent, viz. a noun phrase (= NP).

The central unit in this annotation system is the category: every NP component receives a category label. The notion of category underlying the NPEGL annotation conflates parts of speech and constituents; in the X-bar theoretic sense, the category inventory of NPEGL comprises both X^0s (single word units like the head noun, demonstratives, adjectives etc.) and XPs (phrasal units like genitive phrases and clauses like relative clauses). Thus, at the outset, all NP components are on equal footing due to the flat annotation; they differ primarily by their category label and their linear position. In the NPEGL system, it is possible to encode a number of dependencies; moreover, it also involves a rich annotation for morphological and semantic features, information about syntactic function, and various kinds of metainformation (see Pfaff & Bouma 2024 [this volume], Pfaff 2019a) for a detailed overview and discussion), but these aspects are irrelevant here since we will first and foremost be concerned with linear properties of categories.

Some categories allow for sub-specification of up to four levels, which is encoded via path notation (the levels are separated by a dot); for instance, the modifier category distinguishes cardinal elements and adjectives, and the adjective category, in turn, distinguishes *lexical adjectives* and *functional adjectives* etc. This is illustrated in (1), based on the NPEGL entry (OIce.629.122).

(1)	marga	aðra	röskva	menn	[er þá voru ...]
	Md.Card	Md.Aj.Fn.Dt	Md.Aj.Lx.Pro	N.C	RC
	'many	other	brave	men	who then were ...'

Here, the components of the labels of the first three elements are to be read as follows (the arrows indicate the fully specified label):

Md	= class of modifiers
Md.**Card**	= class of cardinal elements
→ Md.Card.**WQ**	= weak quantifiers
Md.**Aj**	= class of adjectives
Md.Aj.**Fn**	= class of functional adjectives
→ Md.Aj.Fn.**Dt**	= determiner-like adjectives
Md.Aj.**Lx**	= class of lexical adjectives
→ Md.Aj.Lx.**Pro**	= prototypical adjectives

In other words, depending on the level of construal, this example can be seen as involving three modifiers, or a cardinal element and two adjectives, or a weak quantifier, a functional adjective and a lexical adjective. These (sub-)category levels will be referred to as cat^0 (X), cat^1 (X.Y), cat^2 (X.Y.Z) and cat^3 (X.Y.Z.W). The class of nouns (N) allows a cat^1 distinction between common nouns (N.C) and proper nouns (N.P), whereas relative clauses (RC) are not distinguished further. Whenever I report findings from NPEGL, I will use the original annotation labels,[2] but in the running text, I will often simply use e.g. "Adj" instead of Md.Aj.Lx, "N" instead of N.C, or "Num" instead of Md.Card.Num.

The numbers to be presented here are based on the contents of the NPEGL databases, but it is essential to be explicit about what they relate to. NPGEL employs a pre-sorting strategy apriori excluding certain irrelevant (e.g. one-word) noun phrases, and, since annotation is still in progress at the time of writing, "100%" can never mean "all noun phrases in the respective text(s)", but merely "all relevant NPs currently annotated" (see Table 3). It is thus crucial to emphasize that the numbers reported here are mainly intended as an illustration for the underlying methodology rather than as final results in their own right.

For the sake of exposition and for rather practical purposes, I will put two further restrictions on the available data sets in NPEGL by creating *working databases* $\boldsymbol{ndb}_x$ (= "nominal database")[3] that

[2]With one exception: for the sake of readability, I will use "Md.Card" instead of the rather bulky label "Md.Nu/WQ" for cardinal elements used in the official NPEGL annotation.

At the end of the chapter, an overview of the category labels used here is given; for the full overview, see Pfaff & Bouma (2024 [this volume]), Pfaff (2019a).

[3]In the following, I will use the shorthand form ndb where the subscript indicates the respective language. For instance, ndb_{OEng} means "working nominal database for Old English".

(i) only include NPs that contain exactly one "N.C" (= common noun),[4] and

(ii) do not include NPs comprising a coordination structure.

Condition (i) ensures that the core component of the noun phrase, i.e. the head noun, is present; otherwise, notions like *pre-* vs. *postnominal* would be nonsensical. Condition (ii) reduces the number of unnecessary complications and unnecessarily long NPs, which do not add anything to the present discussion.

1.2 Caveat: Patternization

The ideas and methods reported here emerged from experimenting with some peculiarities of the NPEGL annotation system and the question of how the database contents can be utilized to study word order variation.[5] No excessive claim to novelty is made here insofar as the approaches taken are largely based on simple mathematical and combinatorial procedures. Yet the purpose here is not to develop a full-fledged statistical analysis (nor a syntactic analysis, for that matter); the goal is more modest, viz. to offer some practical suggestions and methodological reflections on how to think about word order variation.

At the outset, several procedures, as described here, will either appear rather trivial, or tedious and cumbersome (or downright impossible) – if performed manually. It is therefore crucial to emphasize that the methods discussed here (and their execution) rely on computational assistance, and the actual "protagonist" remains hidden: "Patternization" is a Python tool that I have been developing in the course of the above-mentioned experimenting, and it is this tool that does the actual work. In its current shape, Patternization is adapted to the NPEGL annotation system and processes the NPEGL databases.

This chapter is not, however, meant to be a tool documentation, even though some functionalities will be briefly described in the appendix. Rather its purpose is to show what Patternization actually does and what the motivation for a given procedure is, instead of focusing on technical details of execution. At a more abstract level, the intention is to motivate *Patternization* as a general approach to syntactic diversity, independent of any concrete tools and independent of a specific annotation scheme.

[4] Thus ruling out elliptic noun phrases (without overt head noun), but also proper names, which behave differently from common nouns in relevant (syntactic) respects.

[5] Originally, this chapter was intended as a mere appendix to Pfaff & Bouma (2024 [this volume]).

1.3 Patterns

Pfaff (2015, 2019b) uses the term "pattern" in order to have labels with which to describe the surface diversity found in modified definite noun phrases in Icelandic; the relevant patterns are illustrated in (2) (from Pfaff 2015: 29).

(2) a. **A-WK N-DEF** **(I)**
gul-**i** bíll -**inn**
yellow-WK car -DEF

b. **ART A-WK N** **(II)**
hinn fullkomn-**i** glæpur
ART perfect-WK crime

c. **N-DEF A-WK** **(III)**
heimspekingur -**inn** mikl-**i**
philosopher -DEF great-WK

d. **A-STR N-DEF** **(IV)**
full-**ur** strákur -**inn**
drunk-STR boy -DEF

The labels given – pattern (I), pattern (II) etc. – each stand for a (linear) surface string with specific formal properties and ordering, without, however, suggesting any theoretical status.[6] In this setup, syntactic category (Adj, N), adjectival inflection (strong/weak), and article form (free/suffixed) are formal parameters (or distinctive features) that make up a pattern.

Ultimately, these patterns are just members of a small pre-determined set. In order to deal with diversity within the noun phrase at large, however, certain extensions are inevitable since we cannot tell apriori what kind of patterns we may encounter, or how many. In the following, I will generalize this basic notion of pattern in a particular way that makes best-possible use of the annotation system in NPEGL.

Let us define a pattern simply as a string of objects within a given domain where "domain" essentially corresponds to a syntactic constituent; in the present case: domain = noun phrase/NP. A pattern will be represented as as an n-tuple constituting a linear sequence of n formal objects: (X_1, X_2, ... X_n). The most obvious value for "formal object", which we will be using here, is that of a category (label), and since NPEGL allows for four levels of categorial annotation, we have, in principle, four repositories of pattern-building elements. Differently from the

[6] Pfaff (2019b) moreover shows that the same pattern (in the sense of identical surface strings) can have a different syntactic construal at different times.

narrow conception in (2), we allow for patterns consisting of potential components from a considerably larger pool and, moreover, for patterns of variable length (minimally, though, of length > 1).

Consider the Icelandic example in Table 2 (meaning 'these two big horses') with the corresponding NPEGL category labels (see Section 1.1).

Table 2: Four pattern construals of the same NP

	þessir	*tveir*	*stóru*	*hestar*	
cat^0	**Dem**	**Md**	**Md**	**N**	→ ***patt***^{0}
cat^1	Dem	**Md.Card**	**Md.Aj**	**N.C**	→ ***patt***^{1}
cat^2	Dem	**Md.Card.Nu**	**Md.Aj.Lx**	N.C	→ ***patt***^{2}
cat^3	Dem	Md.Card.Nu	**Md.Aj.Lx.Pro**	N.C	→ ***patt***^{3}

This arrangement of labels gives us four possible pattern construals at a different level of granularity, where *patt*^{n} is to be read as "pattern instantiated by a given NP at the cat^n level of annotation (or simply cat^n pattern)":

patt^0 : (Dem, Md, Md, N)

patt^1 : (Dem, Md.Card, Md.Aj, N.C)

patt^2 : (Dem, Md.Card.Nu, Md.Aj.Lx, N.C)

patt^3 : (Dem, Md.Card.Nu, Md.Aj.Lx.Pro, N.C)

Notice that pattern construal is not limited, in principle, by category level and can also tap into the maximal pool of category labels $\text{CAT}^0 \bigcup \text{CAT}^1 \bigcup \text{CAT}^2 \bigcup \text{CAT}^3$, or a subset thereof. For instance, the above example can just as well be construed as pattern (Dem, **Md**, Md.Aj.Lx, N.C). In this pattern, the first modifier slot is underspecified as it were (restricted to *some* modifier category), so it would also capture noun phrases like

- *these **few** big horses* (Md → Md.Card.WQ),
- *these **other** big horses* (Md → Md.Aj.Fn),
- *these **beautiful** big horses* (Md → Md.Aj.Lx).

A definition of patterns as a sequence of category labels has to be understood relative to a given categorizing system. NPEGL categories include phrasal and clausal categories, thus the patterns to be discussed here are not simply sequences of words, even though the above examples may suggest so. This system also includes patterns such as the following:

- (Md.Aj.Lx, N.C, **GenP**, **PP**) genitive phrase + prepositional phrase
- (Md, N.C, Dem, **RC**) relative clause
- (Dem, Md.Aj.Lx, N.C, **CC.Fi**) complement clause (finite)

Even though GenP may and the other boldprint categories will comprise several words, formally, they are treated as one category, and in this sense, these examples behave just like the above examples, viz. as 4-tuples (= patterns involving four categories).

2 Basic numbers and Pattern Diversity

The current numbers of NPs, categories, and patterns (sorted by category level) in the NPEGL databases, more specifically, in their respective ndb databases, are illustrated in Table 3. By definition, every NP in ndb_x contains exactly one lexical noun "N.C" (see Section 1.1), thus the respective numbers of occurrences of that category is the same as the numbers of NPs given in Table 3. Table 4 lists the next three most frequent categories.

The label *Occurrences* in Table 4 indicates the absolute frequency of the respective category, while *Cat_in_Patt* indicates in how many different patterns that category occurs. As can be seen, the two numbers do not necessarily correlate; a category can be very frequent without being very versatile, and vice versa. For space reasons, we will not look at individual patterns in detail here; suffice it to say that the most frequent pattern in each ndb_x is of length 2: (N.C, Poss), (Dem, N.C), (Poss, N.C), etc.

Given the basic numbers in Table 3, we can calculate a simple type-token ratio – patterns per NPs – which will be referred to as *Pattern Diversity* (PattDiv), where, hypothetically, a value of 1.0 = 100% indicates maximal diversity (every NP instantiates a different pattern). If we take these numbers at face value, we get the ratios illustrated in Table 5.

Table 3: ndb-subdatabases in NPEGL: NPs, categories, patterns

	Old Icel.	Old English	OH German	Old Swedish	Old Saxon
NPs	7981	3260	604	687	6696
CATs					
cat^0	19	16	16	17	16
cat^1	25	22	20	21	20
cat^2	28	27	23	24	23
cat^3	34	30	28	31	28
PATTs					
$patt^0$	384	151	92	75	245
$patt^1$	509	191	103	86	289
$patt^2$	590	214	113	99	351
$patt^3$	708	260	124	107	383

Table 4: Most frequent categories at cat^2 – absolute frequencies and occurrence in patterns

	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
category	Md.Aj.Lx	Dem	Dem	GenP	Dem
abs. freq.	2013	1302	260	178	2485
Cat_in_Patt	200	75	41	21	108
category	Poss	Md.Aj.Lx	Poss	Poss	Md.Aj.Lx
abs. freq.	1706	853	134	173	1759
Cat_in_Patt	94	82	18	15	122
category	Dem	GenP	GenP	Md.Card.Nu	GenP
abs. freq.	1677	604	77	163	1642
Cat_in_Patt	162	59	20	22	124

Table 5: Pattern Diversity: Patterns per NPs (see Table 3)

	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
cat^0	4.8%	4.6%	15.2%	10.9%	3.7%
cat^1	6.4%	5.9%	17.1%	12.5%	4.3%
cat^2	7.4%	6.6%	18.7%	14.4%	5.2%
cat^3	8.9%	8.0%	20.5%	15.6%	5.7%

However, a note of caution is in order, for the numbers in Table 5 give a distorted impression. Notice, in particular, that the numbers of annotated NPs in the various language databases are of different sizes, with a significant difference between Old Icelandic/Old Saxon and Old High German/Old Swedish. In the course of annotation, a certain degree of saturation will be reached, meaning that, while the number of NPs increases steadily, it happens less and less often that a new pattern is introduced and thus the ratio – patterns per NPs – gets "diluted". In other words, for a large number of NPs, the diversity index becomes smaller.

It is, therefore, prudent to establish a *standardized common denominator* **scd** of, say, **scd** = 1000, i.e. patterns per 1000 NPs, in order to provide a more balanced picture. When calculating the values for PattDiv on that basis, we get the numbers in Table 6.[7]

Table 6: Revised PattDiv with scd = 1000

	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
cat^0	13.1%	8.6%			9.5%
cat^1	16.5%	10.6%			10.2%
cat^2	18.5%	11.7%			12.6%
cat^3	21.8%	13.8%			13.4%

One straightforward observation is that we can put a number on diversity and claims such as "the Old Icelandic noun phrase has more variation than the Old English/Saxon noun phrase" can be given numerical substance via the PattDiv index. Thus, while simple, PattDiv gives us an elegant measurement for (degrees of) syntactic diversity.

[7] In the Appendix, we will briefly address the technicalities of this procedure. Also, since the ndbs for Old High German and Old Swedish are of size < 1000, they will be ignored here.

3 Combinatorial flexibility

We will now look at some more advanced issues; consider the examples in (3), found in Old Icelandic saga texts.[8]

(3) a. sína *fullkomna* **vináttu**
poss perfect friendship

b. *fullkomna* **vináttu** sína
perfect friendship poss

c. **vináttu** sinni *fullkominni*
friendship poss perfect

d. *fullkominni* sinni **vináttu**
perfect poss friendship

e. sinni **vináttu** *fullkominni*[9]
poss friendship perfect
'his perfect/complete friendship'

These examples present a rather peculiar instance of diversity insofar as the same lexical items, and, a fortiori, the same categories are involved in all five cases, but in different constellations, i.e. patterns. Now instead of comparing frequencies, let us take the fact *attestation* at face value and focus on the three categories involved. The maximal number of permutations involving three elements, such as {N, Adj, Poss}, is $3! = 3 \times 2 \times 1 = 6$ possible constellations – five of which are shown in (3), while the missing one does not seem to be attested.[10] We can encode this observation with a feature [+/−Att], or simply assign a truth value, cf. (Table 7).

We will take the observation that five out of six possible patterns (involving three categories) are attested as a measurement of *combinatorial flexibility* and

[8]Retrieved from the *Saga Corpus*: http://malheildir.arnastofnun.is/?mode=forn#?corpus=forn.

[9]The possessive and the adjective visibly differ with respect to case, accusative vs. dative (as a consequence of being governed by different verbs). Such case differences are irrelevant in the present context.

[10]The usual disclaimers apply: "not attested" in a (historical) corpus does not necessarily entail that the construction in question is, in fact, ungrammatical.

In the following, the term attestation will be used as a binary parameter (+/−Att) indicating *whether* a particular configuration is found in a given language/text in the first place – rather than *how often*; when talking about (absolute) frequencies, we will instead use occurrence.

Table 7: Attested and non-attested patterns of {N, Poss, Adj }

	{N, Poss, Adj }		5/6
i.	**Poss Adj N**	[+Att]	**True**
ii.	**Adj N Poss**	[+Att]	**True**
iii.	**N Poss Adj**	[+Att]	**True**
iv.	**Adj Poss N**	[+Att]	**True**
v.	**Poss N Adj**	[+Att]	**True**
vi.	**N Adj Poss**	[–Att]	**False**

notate it as **CombFlex({N, Adj, Poss})** = 5/6.[11] Thus combinatorial flexibility tells us something about which categories combine in how many ways. Differently from pattern diversity, it tells us something about actual diversity in relation to potential diversity by making reference to the maximum of possible permutations.

When assessing combinatorial flexibility, the actual number of occurrences of the respective patterns is irrelevant; what counts is their attestation value. By default, [–Att] is tantamount to zero occurrences. However, for many practical purposes, a *threshold value* X might be warranted such that [+Att] requires there to be $x \geq$ X occurrences; in that case, [–Att] is the result of $x <$ X occurrences. For the sake of illustration, the following discussion is based on the minimum setting X = 1 and [+Att] $\leftrightarrow x \geq 1$.

The illuminating example (3) above was an accidental finding, but it led to an interesting way of looking at syntactic diversity. In the following, we will develop this into a full-blown method that is systematic and, above all, exhaustive in the sense that it enables us to examine the whole spectrum of attested per potential permutations in a given domain. Before addressing the actual procedure, I will give a brief definition of the mathematical notions *permutation* and *combination* and some terminology relevant for the implementation.

3.1 Basic combinatorics refresher

Combinatorics is a branch of mathematics that examines the ways in which (arrangements of) objects can be counted. For the discussion to follow, we will especially rely on the concepts *(sub-)permutation* and *combination*. Given a set S with

[11] In accordance with the project title *Constraints on syntactic variation*, Table 7 can also be given a purely extensional interpretation: Rows i-v in Table 7 represent the variation, Column vi. is the constraint (on variation).

n distinct elements, then n! (read: n factorial) is the number of possible permutations, i.e. different arrangements, of the n elements; the ordering of the elements matters. A combination is essentially a set, here a subset of S, and the number of k-combinations is the number of different subsets of S of cardinality k. We have $\binom{n}{k}$ (read: n choose k) k-combinations in S. Being a set, the internal ordering of a combination does not matter. The relevant details are summarized and illustrated below:[12]

⇒ Given a *sample space* (= set) S, with |S| = n, and $k \in \mathbb{N} \leq n$, then there are

- $n!$ $= n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1$ (full) ***permutations*** of size n
- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ ***k-combinations*** ~ sub-sets of size k
- $\binom{n}{k \times k!} = \frac{n!}{(n-k)!}$ ***k-permutations*** ~ sub-permutations of size k

⇒ Suppose **S = {A, B, C, D, E, F}** with n = |S| = **6**; let k = 3; then there are

(I) $\binom{6}{3} = \frac{6!}{3!(6-3)!}$ = **20** possible ***3-combinations***:

{A, C, B}	{A, C, D}	{A, D, E}	{A, C, E}	{A, B, D}
{A, D, F}	{A, E, F}	{A, C, F}	{A, E, B}	{A, B, F}
{F, D, C}	{F, E, B}	{B, C, D}	{C, E, B}	{B, C, F}
{B, D, F}	{C, D, E}	{B, D, E}	{C, E, F}	{D, E, F}

Combinations are sets, hence the ordering does not matter; therefore {A, C, B} = {C, A, B} = {B, A, C} = {B, C, A} etc.

(II) $\binom{6}{3 \times 3!}$ = **20 x 6 = 120** possible ***3-permutations***:

(A, C, B)	(A, B, C)	(B, A, C)	(B, C, A)	(C, A, B)	(C, B, A)
(A, C, D)	(A, D, C)	(C, A, D)	(C, D, A)	(D, A, C)	(D, C, A)
(A, D, E)	(A, E, D)	(D, A, E)	(D, E, A)	(E, A, D)	(E, D, A)
(A, C, E)	(A, E, C)	(C, A, E)	(C, E, A)	(E, A, C)	(E, C, A)
(A, B, D)	(A, D, B)	(B, A, D)	(B, D, A)	(D, A, B)	(D, B, A)
(A, D, F)	(A, F, D)	(D, A, F)	(D, F, A)	(F, A, D)	(F, D, A)

[12] Following common mathematical conventions, we will notate actual, that is unordered Sets with curly brackets: {a, b, c}, while tuples, which are ordered sequences, will be notated with parentheses: (a, b, c).

(A, E, F)	(A, F, E)	(E, A, F)	(E, F, A)	(F, A, E)	(F, E, A)
(A, C, F)	(A, F, C)	(C, A, F)	(C, F, A)	(F, A, C)	(F, C, A)
(A, E, B)	(A, B, E)	(B, A, E)	(B, E, A)	(E, A, B)	(E, B, A)
(A, B, F)	(A, F, B)	(B, A, F)	(B, F, A)	(F, A, B)	(F, B, A)
(F, D, C)	(C, F, D)	(D, C, F)	(D, F, C)	(F, C, D)	(C, D, F)
(F, E, B)	(B, F, E)	(E, B, F)	(E, F, B)	(F, B, E)	(B, E, F)
(B, C, D)	(B, D, C)	(C, B, D)	(C, D, B)	(D, B, C)	(D, C, B)
(C, E, B)	(B, E, C)	(C, B, E)	(B, C, E)	(E, B, C)	(E, C, B)
(B, C, F)	(B, F, C)	(C, B, F)	(C, F, B)	(F, B, C)	(F, C, B)
(B, D, F)	(B, F, D)	(D, B, F)	(D, F, B)	(F, B, D)	(F, D, B)
(C, D, E)	(C, E, D)	(D, C, E)	(D, E, C)	(E, C, D)	(E, D, C)
(B, D, E)	(B, E, D)	(D, B, E)	(D, E, B)	(E, B, D)	(E, D, B)
(C, E, F)	(C, F, E)	(E, C, F)	(E, F, C)	(F, C, E)	(F, E, C)
(D, E, F)	(D, F, E)	(E, D, F)	(E, F, D)	(F, D, E)	(F, E, D)

(Sub-)permutations will be represented as tuples since the ordering does matter: (A, C, B) ≠ (C, A, B) ≠ (B, C, A) etc.

In the following, I will use the term *permutation group* for the set of possible permutations of a given combination:

combination	{A, C, B}
permutation group	{ (A, B, C), (A, C, B), (B, C A), (B, A, C), (C, A, B), (C, B, A) }

3.2 Patterns and permutations

For the present purpose, the relevant sample space S_{cat} obviously makes reference to category labels (or annotation features more generally). S_{cat} may be the entire categorial inventory or constitute a more or less random selection/subset of category labels, e.g.

- S_{cat} = CAT = $cat^0 \bigcup cat^1 \bigcup cat^2 \bigcup cat^3$ (complete category set)
- S_{cat} = cat^2 (cat^2 categories)
- S_{cat} = {Poss, Md.Aj, PP, Q, Dem, GenP, N.C, RC} (random selection)

The general procedure is as follows: after establishing S_{cat} and the prospective pattern size k, we generate all $\binom{|S_{cat}|}{k}$ permutation groups, which will then serve

as search patterns to browse the database. The query results, in turn, will allow us to determine CombFlex($\{c_1, c_2 \ldots c_k\}$) for any k categories $c_1, c_2 \ldots c_k \in S_{cat}$.

For convenience, we can reduce some unnecessary noise. Since the ndb restriction guarantees that every NP contains exactly one noun, we will take advantage of that and only consider combinations that include a noun. Thus with $k = 3$, we first generate *all* 3-combinations of S_{cat}, but sort out those that do not contain a category label "N.C", as in (4). For those combinations that do, however, we will then generate the respective permutation groups, cf. (5).

(4) a. {RC, Dem, Q} (combinations *not* satisfying
b. {Mdmd, GenP, Poss} the restriction »contains "N.C"«
c. {Dem, Q, Poss} will be ignored)

(5) a. { **N.C**, Poss, Md.Aj } (satisfies the restriction)
⇒ generate permutations:
(Poss, N.C, Md.Aj), (Poss, Md.Aj, N.C), (N.C, Md.Aj, Poss),
(N.C, Poss, Md.Aj), (Md.Aj, Poss, N.C), (Md.Aj, N.C, Poss)
b. { Dem, **N.C**, RC } (satisfies the restriction)
⇒ generate permutations:
(Dem, N.C, RC), (N.C, Dem, RC), (Dem, RC, N.C),
(RC, Dem, N.C), (N.C, RC, Dem), (RC, N.C, Dem)
etc. . . . . .

In the next step, the respective ndb_x will be probed for attestations of each member of all permutation groups generated. In (6), a small selection of the results for a search in ndb_{OIcel} with $k = 3$ is given.

(6) a. **{Md.Aj, App, N.C}:** **1 / 6**
i. (App, Md.Aj, N): FALSE
ii. (App, N.C, Md.Aj) FALSE
iii. (Md.Aj, App, N.C): FALSE
iv. (Md.Aj, N.C, App): TRUE
v. (N.C, App, Md.Aj): FALSE
vi. (N.C, Md.Aj, App): FALSE

b. **{N.C, Dem, RC}:** **2 / 6**
i. (Dem, N.C, RC): TRUE
ii. (N.C, Dem, RC): TRUE
iii. (RC, N.C, Dem): FALSE

iv. (RC, Dem, N): FALSE
v. (N.C, RC, Dem): FALSE
vi. (Dem, RC, N.C): FALSE

c. **{N.C, Dem, Md.Aj.Lx}:** **3 / 6**
i. (Dem, Md.Aj, N.C): TRUE
ii. (Dem, N.C, Md.Aj): TRUE
iii. (Md.Aj, N.C, Dem): FALSE
iv. (N.C, Dem, Md.Aj): TRUE
v. (Md.Aj, Dem, N.C): FALSE
vi. (N.C, Md.Aj, Dem): FALSE

d. **{N.C, Md.Card.WQ, Md.Aj}:** **4 / 6**
i. (Md.Aj, Md.Card.WQ, N.C): TRUE
ii. (Md.Aj, N.C, Md.Card.WQ): TRUE
iii. (Md.Card.WQ, Md.Aj, N.C): TRUE
iv. (Md.Card.WQ, N.C, Md.Aj): TRUE
v. (N.C, Md.Aj, Md.Card.WQ): FALSE
vi. (N.C,Md.Card.WQ, Md.Aj): FALSE

e. **{N.C, Md.Aj, Poss}:** **5 / 6**
i. (Md.Aj, N.C, Poss): TRUE
ii. (Poss, Md.Aj, N): TRUE
iii. (N.C, Poss, Md.Aj): TRUE
iv. (Poss, N.C, Md.Aj): TRUE
v. (Md.Aj, Poss, N.C): TRUE
vi. (N.C, Md.Aj, Poss): FALSE

f. **{Q, N.C, Md.Aj}:** **6 / 6**
i. (Q, Md.Aj, N.C): TRUE
ii. (Q, N.C, Md.Aj): TRUE
iii. (Md.Aj, Q, N.C): TRUE
iv. (Md.Aj, N.C, Q): TRUE

v. (N.C, Md.Aj, Q): TRUE
vi. (N.C, Q, Md.Aj): TRUE

As might be expected, in many cases, not more than one or two of the permutations are ATTESTED, and often those are not very insightful.[13] However, we also find combinations, for which up to all six out of six possible permutations are ATTESTED, and permutation groups with COMBFLEX = 4/6 or higher are surely worth closer examination. But the most outstanding feature of this procedure is that it is completely exhaustive: for any 3-permutation in S_{cat}, we will determine whether it is ATTESTED or not, and, concomitantly, for any permutation group, we will ascertain its combinatorial flexibility – as partially illustrated in (6). In Table 9, the numbers of permutation groups for each value of COMBFLEX are given.

Table 9: Combinatorial flexibility in $S_{cat} = cat^2$ with $k = 3$

COMBFLEX	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
1/6	31	41	27	31	13
2/6	59	41	19	17	28
3/6	20	10	6	7	20
4/6	13	2	0	3	10
5/6	19	5	0	0	4
6/6	11	0	0	0	6

Thus we have, for instance, six permutation groups in Old Saxon with the maximal COMBFLEX 6/6, five permutation groups in Old English with COMBFLEX 5/6 etc. Based on those numbers, we can, in turn, calculate a mean combinatorial flexibility μ-COMBFLEX that tells us how many permutations we find on average – per permuation group and per language, cf. Table 10.

Table 10: Mean combinatorial flexibility in $S_{cat} = cat^2$ with $k = 3$

	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
μ-COMBFLEX	2.8/6	1.9/6	1.6/6	1.7/6	2.8/6

The numbers in Table 10 constitute a simplification insofar as they are based on the number of permutation groups of which at least one permutation yields

[13] For instance, the fact that only permutations with the relative clause in final position are attested, cf. (6b), is not really surprising.

TRUE, while permutation groups with COMBFLEX 0/6 are not considered here. Let us refer to a permutation group with COMBFLEX 1/6 – 6/6 as C_{att} (= "attested combination"), and conversely, to every potential permutation group generated on the basis of the respective category inventory (see Table 3) as C_{pot} (= "potential combination"). With the numbers for these, we can calculate the ratio attested permutation groups per potential permutation groups; effectively, this ratio tells us how often three categories can co-occur, given the entire spectrum of categories available and the resulting possible three-way combinations. Likewise we can calculate the mean combinatorial flexibility that includes non-ATTESTED permutation groups (i.e. with the value 0/6); call this μ-COMBFLEX^0, cf. Table 11.

Table 11: Potential and attested combinations; modified combinatorial flexibility

	Old Icelandic	Old English	OH German	Old Swedish	Old Saxon
categories	28	27	23	24	23
$\mathbf{C}_{pot}$	351	325	231	253	231
$\mathbf{C}_{att}$	153	99	52	58	81
$\frac{C_{att}}{C_{pot}}$	0.436	0.305	0.225	0.229	0.351
μ-COMBFLEX^0	**1.2/6**	**0.6/6**	**0.4/6**	**0.4/6**	**1.0/6**

Obviously, since a permutation is a discrete sequence, we cannot literally have something like *1.9* or *0.6 (out of 6) permutations*. μ-COMBFLEX and μ-COMBFLEX^0 must be understood more abstractly as the overall degree of categorial versatility indicating how likely categories $\in S_{cat}$ are to combine with other categories $\in S_{cat}$. Hence mean combinatorial flexibility allows us to measure the overall *potential* syntactic diversity in relation to a maximum – thus entailing a measurement of the constraints on that diversity.

3.3 Patterns in the patterns

Even though the procedure as described above involves permutation groups at large, implicitly we have already stipulated a condition: "N.C"; i.e. we have been looking at potential permutations in the presence of a noun. We can go one step further by fixing a second parameter. Consider the permutation group { N.C, Md.Aj, **X** } where X is a variable over categories $\in S_{cat}$. Here we are constructing a macro permutation group probing for the distribution of categories X in the context of a noun and an adjective.

For instance, with X = {Dem, Num, Poss, Q, WQ} we can examine the behaviour of elements that (on a generous conception) may be considered determiner(-like) elements in that context. Below, the results for ndb_{OIcel} are given, indicating how many and which x ∈ X are attested in the respective permutation:

(7)	a.	(**x**, Md.Aj, N.C):	5	x ∈ {WQ, Dem, Num, Poss, Q} = X
	b.	(**x**, N.C, Md.Aj):	5	x ∈ {WQ, Dem, Num, Poss, Q} = X
	c.	(Md.Aj, **x**, N.C):	5	x ∈ {WQ, Dem, Num, Poss, Q} = X
	d.	(Md.Aj, N.C, **x**):	4	x ∈ {Dem, Num, Poss, Q} ⊂ X
	e.	(N.C, **x**, Md.Aj):	4	x ∈ {Dem, Num, Poss, Q} ⊂ X
	f.	(N.C, Md.Aj, **x**):	1	x ∈ {Q} ⊂ X

For this particular sample, we can, among other things, infer that all items in X occur in the permutations in (7a)–(7c), and that demonstratives, possessives and numerals have an identical distribution in the context of nouns and adjectives: all three occur in the permutations (7a)–(7e), and all three do not occur in the permutation (7f).[14]

Provided the dataset is large enough, instead of merely considering Md.Aj, we can use any category y ∈ S_{cat} as a second parameter and let X = ∈ S_{cat} in order to probe into { N.C, **y**, **X** } and examine the entire categorial space and determine the overall extent of co-distributions.

3.4 Markedness hierarchies(?)

In Section 3.2, we looked at permutation groups and combinatorial flexibility from a purely quantitative perspective; Table 11 only gives the numbers of categories and permutation groups, but no information about *which* categories are involved in *which* permutation group, or *which* permutation groups occur in *which* language with *which* combinatorial flexibility.

Naturally, we can perform various qualitative re-runs of the whole procedure by examining *which* permutation groups are ATTESTED in all or some (or none) of the individual languages. Specifically, for every permutation group $pg \in \mathbf{C}_{pot}$ (i.e. the entirety of permutation groups generated), we can compare CombFlex(pg) for the respective languages. In Table 12, one permutation group is illustrated.

This way, we can directly compare the individual permutations and their ATTESTATION in the respective languages. That is we can examine whether there

[14] Be careful not to confuse the numbers given in (7) with values for combinatorial flexibility; CombFlex({N.C, Md.Aj, **x**}) is 6/6 for x = Q, 5/6 for x = Dem/Num/Poss, and 3/6 for x = WQ.

Table 12: CombFlex({Poss, N, Md.Aj}) in comparison

	Old Icelandic	Old English	Old High German	Old Swedish	Old Saxon
{Md.Aj, Poss, N}	5/6	1/6	2/6	3/6	2/6
(Poss, Md.Aj, N)	TRUE	TRUE	TRUE	TRUE	TRUE
(Poss, N, Md.Aj)	TRUE	FALSE	TRUE	TRUE	TRUE
(N, Poss, Md.Aj)	TRUE	FALSE	FALSE	TRUE	FALSE
(Md.Aj, N, Poss)	TRUE	FALSE	FALSE	FALSE	FALSE
(Md.Aj, Poss, N)	TRUE	FALSE	FALSE	FALSE	FALSE
(N, Md.Aj, Poss)	FALSE	FALSE	FALSE	FALSE	FALSE

is a regularity as to whether a given permutation is ATTESTED or not. Notice, in particular, that the individual permutations in Table 12 are arranged in a particular manner such that like values form "blocks" as it were: there is a TRUE-block and a FALSE-block, but no TRUE-FALSE-TRUE alternations in any language.

While this is merely an initial observation, it can be formulated as an empirical and methodological question: can all permutation groups be arranged in this way? In order to illustrate the relevance of this question, consider the scenario for the hypothetical languages V–Z in Table 13.

Table 13: CombFlex({A, B, C}) – hypothetical (idealized) scenario

	V	**W**	**X**	**Y**	**Z**
{A, B, C}	5/6	4/6	3/6	2/6	1/6
(A, B, C)	TRUE	TRUE	TRUE	TRUE	TRUE
(B, C, A)	TRUE	TRUE	TRUE	TRUE	FALSE
(B, A, C)	TRUE	TRUE	TRUE	FALSE	FALSE
(A, C, B)	TRUE	TRUE	FALSE	FALSE	FALSE
(C, A, B)	TRUE	FALSE	FALSE	FALSE	FALSE
(C, B, A)	FALSE	FALSE	FALSE	FALSE	FALSE

These "results" plausibly suggest that (A, B, C) is the unmarked or default pattern in the permutation group since it is ATTESTED in all languages under consideration. Given the arrangement, we can moreover construe the left-hand column, read top-down, as a markedness hierarchy, or even as an implicational hierarchy; e.g. if a language has (B, A, C), it also has (B, C, A) etc.

The extent to which this arrangement is possible is of course an empirical question, but whenever it is possible, CombFlex not only gives a measurement for flexibility as such, but can also be understood as an indicator of the degree of markedness possible/allowed in a given language (relative to a given permutation group).

4 Search patterns and matched patterns

So far we have used the term "patterns" indiscriminately for strings of category labels. In this subsection, we will have a look at some possible refinements. When working with databases and search interfaces, an obvious distinction is that between a query and the output to that query. Consequently, I will make a distinction between *search patterns* (S-patterns) and *matched patterns* (M-patterns), where the former abstractly define properties that we are interested in, while the latter are the concrete findings in a given database satisfying the respective criteria. Notably, we will allow specifications where the two are not necessarily a perfect match. In Table 14, some possible configurations for S-patterns (red) and corresponding M-patterns (blue) are given.[15]

Table 14: S-patterns and M-patterns

`precise_pattern`(A, B, C):	(A, B, C)
`rigid_pattern`(A, B, C):	(. . . A, B, C . . .)
`flexi_pattern`(A, B, C):	(. . . A, . . . B, . . . C . . .)
`Left_rigid_pattern`(A, B, C):	(A, B, C . . .)
`Left_flexi_pattern`(A, B, C):	(A, . . . B, . . . C . . .)
`Right_rigid_pattern`(A, B, C):	(. . . A, B, C)
`Right_flexi_pattern`(A, B, C):	(. . . A, . . . B, . . . C)

A `precise_pattern` works according to the motto *what you search is what you get*; we have a perfect match. In contrast, the corresponding `rigid_pattern` yields TRUE also for those cases that contain material preceding or following the actual search string. Finally, a `flexi_pattern` also yields TRUE if somethings intervenes between the labels specified in the S-pattern, in other words, it merely encodes the relative linear ordering, but not adjacency.

[15] More advanced refinements could include the incorporation of aspects of the regular expression syntax, which would allow S-patterns such as (A, {B OR F}, C) or (A, {NOT B}, C).

The findings presented in the previous sections are based on `precise_pattern`, but of course, CombFlex can also be computed on the basis of `rigid_pattern` or `flexi_pattern`. This may be useful e.g. when we are not interested in accompanying material such as NP-final relative clauses, or intervening adverbials (like *very*). In particular, the procedure as described so far treats e.g. (Dem, Adj, N), (Dem, Adj, Adj, N), (Q, Dem, Adj, N) and (Dem, Adj, N, RC) etc. as distinct patterns, and we may miss generalizations. Specifications such as `flexi_pattern`(Dem, Adj, N) allow us to treat those as one pattern at a relevant and more abstract level, e.g. conflate patterns where demonstratives precede adjectives in prenominal position etc.

The `Left_/Right_` alignment patterns impose the additional condition that the first/last category matches. One possible application of these will be illustrated with an example from $\text{ndb}_{\text{OIcel}}$. Demonstratives in Old Icelandic can occur prenominally and postnominally, with or without accompanying material; notably, they can occur noun phrase finally (8).

(8) sá maður — maður sá
DEM man — man DEM
'that man'

Now assume we have S_{cat} = {Poss, Q, Md.Aj.Lx, Md.Aj.Fn, Md.Card.Nu, GenP, Md.Card.WQ} and are interested in their compatibility with demonstratives postnominally. To that end, we compare two S-patterns (N, x, Dem), with $x \in S_{cat}$. In (9), the number of occurrences in that pattern are given for each category, first for an alignment pattern and next for the corresponding non-alignment version.

(9) $\forall$ *cat* $\in S_{cat}$:
$\forall$ *np* $\in \text{ndb}_{\text{OIcel}}$:

a. → `Right_flexi_pattern(`*np*`, N.C, `*cat*`, Dem)`
returns TRUE for [$_{NP}$ ... N.C ... *cat* ... Dem]

cat =

Poss	0
Q	0
Md.Aj.Lx	0
Md.Aj.Fn	0
Md.Card.Nu	0
Md.Card.WQ	0
GenP	0

b. → `flexi_pattern(`*np*`, N.C, `*cat*`, Dem)`
returns TRUE for [$_{NP}$ ... N.C ... *cat* ... Dem ...]

	Poss	21
	Q	10
	Md.Aj.Lx	5
cat =	Md.Aj.Fn	1
	Md.Card.Nu	9
	Md.Card.WQ	8
	GenP	21

We observe an interesting discrepancy. The alignment pattern in (9a) yields zero hits for each category, showing that demonstratives cannot follow those in postnominal position **and** simultaneously be pattern-final. On the other hand, (9b) shows that each pattern does occur once the alignment constraint is dropped. This means that a demonstrative actually can follow those categories postnominally provided it is itself followed by other material. In this present case, we can identify the cause as relative clauses; in Old Icelandic, the demonstrative *sá* often co-occurs with a relative clause (or sometimes a complement clause). If we modify the S-pattern accordingly, we get the results in (10).

(10) → `flexi_pattern(`*np*`, N.C, `*cat*`, Dem, RC)`
returns TRUE for [$_{NP}$ ... N.C ... *cat* ... Dem ... **RC** ...]

	Poss	20
	Q	10
	Md.Aj.Lx	5
cat =	Md.Aj.Fn	1
	Md.Card.Nu	9
	Md.Card.WQ	8
	GenP	19

These numbers are almost identical to those in (9b), suggesting that the presence of a relative clause is indeed a pre-condition for demonstratives to follow the categories in postnominal position.[16] Some examples are given for illustration in (11) (intervening material is underlined).

[16] Moreover, a closer inspection of the respective M-patterns reveals that the demonstrative must be adjacent to the relative clause in postnominal position: [... N.C ... *cat* ... Dem, RC ...]. Some authors even suggest that *sá* is a relative pronoun in this use, e.g. Wagener (2017); Sapp (2019).

(11) a. líkamir *dauðra manna* þeir er í moldu höfðu legið
bodies dead.GEN men.GEN DEM REL in ground had lain
'the bodies of dead men that had lain in the ground' (OIce.509.120)

b. konur *nokkurar* þær er hann hafði leyst af óhreinum öndum
women some DEM REL he had released of impure spirits
'some woman whom he had released of impure spirits' (OIce.861.230)

c. wind *hvassan* þann er för þeirra flutti í góða höfn
wind sharp DEM REL journey their transported in good harbour
'a sharp wind that brought them to a good harbour' (OIce.915.632)

In short, different specifications for S-patterns allow us to examine patterns at different levels of granularity; all methods described in the previous sections are applicable. Moreover, the approach of comparing two S-patterns gives us a simple method of probing for correlations or co-dependencies by examining discrepancies.

5 Schrödinger's *Cats*

In the previous sections, we examined the details of word order variation in the NP focusing on patterns and permutations. In this section, we will abstract from concrete patterns, and look at the distribution of categories from a non-discrete perspective. More specifically, we will first have a look at a probabilistic category distribution across the entire NP. In a next step, we will take the noun as an anchoring position dividing the NP into a prenominal and a postnominal space, and examine the distribution of categories (modulo N.C) in those narrow domains. Finally, we will visualize this probabilistic distribution in a Cartesian coordinate system.

5.1 Probablistic category distribution

We begin by counting category occurrences per positon. In the first round, we simply start at the NP-initial position and count the categories in position 1, position 2 ... up to position *n*, where *n* is the number of categories comprised by the longest NP in the respective database. For illustration, consider the following patterns; the subscripts indicate the position (or column in a table):

(12) a. $\text{Dem}_1\ \text{Adj}_2\ \text{N}_3$
b. $\text{Adj}_1\ \text{N}_2$

c. Q_1 Dem_2 Adj_3 N_4 RC_5
d. N_1 Dem_2 RC_3
e. Q_1 Adj_2 N_3
f. . . . . .

Since this procedure is numeric and not phrase structure sensitive, the same category can occur in different positions, and different categories can occur in the same position. In other words, this notion of position is not a syntactic one, but simply indicates left-alignment. When all NPs in a given database are thusly aligned, we add the category occurrences per column as well as the overall total of items in each column. In a parallel fashion, category occurrences can be counted backwards starting from the final position (= right-aligned), i.e. positions -1, -2, -3 ... *-n*.

In Tables 15 and 16, the overall column totals and the numbers for some categories in ndb_{OIcel} are given for the first and last five slots starting from the initial and final positions, respectively.

Table 15: Category occurrences in Old Icelandic, left-aligned

1		**2**		**3**		**4**		**5**	
N.C:	2437	N.C:	4145	N.C:	1277	RC:	391	RC:	93
Md.Aj.Lx:	1113	Md.Aj.Lx:	705	RC:	630	N.C:	117	Dem:	10
Dem:	1051	Dem:	351	Dem:	213	Dem:	50	N.C:	5
RC:	—	RC:	6	Md.Aj.Lx:	147	Md.Aj.Lx:	44	Md.Aj.Lx:	4
. . . . .		. . . . .		. . . . .		. . . . .		. . . . .	
total:	7981	total:	7981	total:	3280	total:	946	total:	163

Table 16: Category occurrences in Old Icelandic, right-aligned

-5		**-4**		**-3**		**-2**		**-1**	
Dem:	34	Dem:	174	N.C:	745	N.C:	3299	N.C:	3769
N.C:	25	N.C:	140	Dem:	544	Md.Aj.Lx:	1325	RC:	1090
Md.Aj.Lx:	10	Md.Aj.Lx:	113	Md.Aj.Lx:	368	Dem:	869	Md.Aj.Lx:	194
RC:	—	RC:	—	RC:	5	RC:	39	Dem:	54
. . . . .		. . . . .		. . . . .		. . . . .		. . . . .	
total:	163	total:	946	total:	3280	total:	7981	total:	7981

With these numbers, we can calculate some simple distributional ratios. For instance, the ratio *category column total per overall column total* indicates the probability for a randomly selected NP, that the respective position is occupied by the respective category; let us notate this as **PosProb**(*position*, *category*). For instance: PosProb(2, Md.Aj.Lx) = 8.8%, or PosProb(-1, RC) = 13.6%.

Likewise, we can calculate *category column total per overall category total* (see Table 4), which indicates the probability that the respective category will occur in that particular position; for instance: the probability that a lexical adjective will occur in the initial position is 55.3%.

In other words, these ratios allow us to map out the probabilities of category distribution within the average NP. But so far, all categories have been treated alike, and, other than left/right alignment, there is no ordering or structural criterion. A third position from either direction could, in principle, amount to a prenominal or a postnominal position – which is obviously relevant information not accessible here. Since we are investigating noun phrases, the head noun is obviously a designated category. More to the point, since, by our ndb-restriction, every NP contains exactly one noun, we can use the noun as a special anchoring point and divide the NP into a prenominal and a postnominal space, while leaving the noun as such out of the consideration (= assigning it position +/-0). This reduces the numbers of positions in a non-trivial way, and puts them in relation to the noun so that we will be talking e.g. about the *final* ***prenominal*** *position*, or the *second* ***postnominal*** *position*.

Once we have partitioned the NP relative to the N position, we apply the same procedure as described above. In Tables 17 and 18, the numbers for some categories are given.

Table 17: Category occurrences in the prenominal field, left-aligned

1		**2**		**3**		**4**	
Md.Aj.Lx:	1113	Md.Aj.Lx:	575	Md.Aj.Lx:	67	Md.Aj.Lx:	3
Dem:	1051	Dem:	62	GenP:	6	GenP:	1
GenP:	194	GenP:	35	Dem:	2	Dem:	—
. . . . .		. . . . .		. . . . .		. . . . .	
total:	5544	total:	1399	total:	122	total:	5

Table 18: Category occurrences in the postnominal field, left-aligned

1		2		3		4		5	
RC:	501	RC:	523	RC:	93	RC:	14	RC:	2
Md.Aj.Lx:	170	Md.Aj.Lx:	60	Md.Aj.Lx:	22	Md.Aj.Lx:	3	Md.Aj.Lx:	–
Dem:	488	Dem:	69	Dem:	4	Dem:	–	Dem:	–
PP:	125	PP:	28	PP:	4	PP:	1	PP:	–
.....	.....	.....	.....	.....					
total:	4212	total:	913	total:	168	total:	28	total:	2

There are four columns in Table 17 and five columns in Table 18 because that is the maximum number of categories that occur simultaneously in $\text{ndb}_{\text{OIcel}}$, prenominally and postnominally, respectively. This is an abstraction over those spaces, because the enumerations obviously also include NPs with less than four prenominal and less than five postnominal categories,[17] but disregards the noun itself. If there is only one prenominal category *cat*, the total of *cat*, and thus the column total, increases by one in position 1 (or -1),[18] but nothing happens to the other positions. For this reason, the column total is highest in position 1/-1, and decreases as we move to the left/right since there are more NPs with at least one prenominal category than with two, etc.

5.2 Distance from N: Visualizing categorial distribution

As just noted, the overall total numbers decrease for columns further to the right. But this correlation does not (necessarily) apply to the ratio PosProb; for instance, $\text{PosProb}(1, \text{Dem})_{pre}$ and $\text{PosProb}(1, \text{Md.Aj.Lx})_{pre}$ are about the same, ca. 20%. However, while that ratio steadily increases for adjectives from position 1 to 4 (20.1% – 41.1% – 54.9% – 60.0%), it decreases for demonstratives (19.0% – 4.4% – 1.6% – 0.0%).

Obviously, this trend also tells us something about the distributional properties of categories. When comparing ratios, we abstractly observe that some categories *tend to be closer to the noun*: they score high(er) in the positions to the right (e.g. adjectives), which means that they are often preceded by material, while others *tend to be further away from the noun*: they score high(er) in the

[17] Thus, for instance, 5544 is the number of NPs containing *at least* one prenominal category, 1399 NPs containing *at least* two prenominal categories etc.

[18] With only one prenominal element, the initial position is identical to the final position.

positions to the left (e.g. demonstratives), which means that they often precede material. Obviously, this is a reflex of more general word order regularities; after all when co-occurring, e.g. demonstratives normally precede adjectives (in prenominal position; see a.o. Cinque 2005). Theoretical syntax has a number of discrete, formal devices to capture those regularities, e.g. phrase structure rules, topological fields, functional sequences etc., but as stated above, in this section, we will consider category distribution in a continuous, non-discrete space.

The general idea is that, if we apply the sequences of column ratios for each category against each other in an appropriate fashion, we will get a mean value $x \in \mathbb{R}$, with $4 \geq x > 0$, for each category indicating "distance from N". For simplicity, the maximal score here is 4 because there are four columns; also, the minimal score is greater than zero since 0 abstractly denotes the noun itself. There are several possible parameters to take into consideration, but also a number of non-trivial complications. I will not discuss the mathematical technicalities of deriving an optimal algorithm to calculate x here; instead I will use a simpler method for the calculation (see Appendix). For Old Icelandic, Old English and Old Saxon, the respective scores for the most frequent categories are given in Table 19.

Table 19: "Distance-from-the-noun" scores (prenominally)

(a) OIcel		(b) OEngl		(c) OSax	
Mdmd:	4.0	Mdmd:	4.0	Mdmd:	3.9
Q:	3.6	Q:	3.7	Md.Card.Nu:	3.9
Dem:	3.1	Dem:	3.5	Dem:	3.8
Md.Card.WQ:	2.4	Poss:	3.2	Q:	3.1
Md.Card.Nu:	2.1	GenP:	2.1	Poss:	2.6
Poss:	1.9	Md.Card.Nu:	1.8	Md.Aj.Fn:	1.0
Md.Aj.Fn:	0.5	Md.Card.WQ:	1.1	GenP:	0.7
GenP:	0.5	Md.Aj.Fn:	0.3	Md.Aj.Lx:	0.3
Md.Aj.Lx:	0.2	Md.Aj.Lx:	0.1	Md.Card.WQ:	–

Now we construe the NP as a Cartesian plane such that the y-axis ($x = 0$) represents the noun (position) in abstracto, the negative x-axis the prenominal space, and the positive x-axis the postnominal space. Since we are focusing on the prenominal space, we have to conceive of the above values as negative numbers. We will furthermore map (absolute) category frequencies onto the y-axis, which allows us to treat categories as coordinates in the Cartesian plane, i.e. to

locate categories in two-dimensional space. In addition, precedence relations are represented as a graph network where precedence scores are calculated on the basis of co-occurrences of two categories A and B in the individual NPs (how often do A and B co-occur, and in which order(s)?). These precedence relations are specified as follows: A → B (red arrow) – A always precedes B when co-occurring; A → B (green arrow) – A precedes B in more than 66% of co-occurrences; A → B (blue arrow) – A always precedes B, but there are fewer than 10 co-occurrences.

In Figures 1–3 I give an illustration of the prenominal space of the Old Icelandic, Old English and Old Saxon NP based on the above scores and specifications.

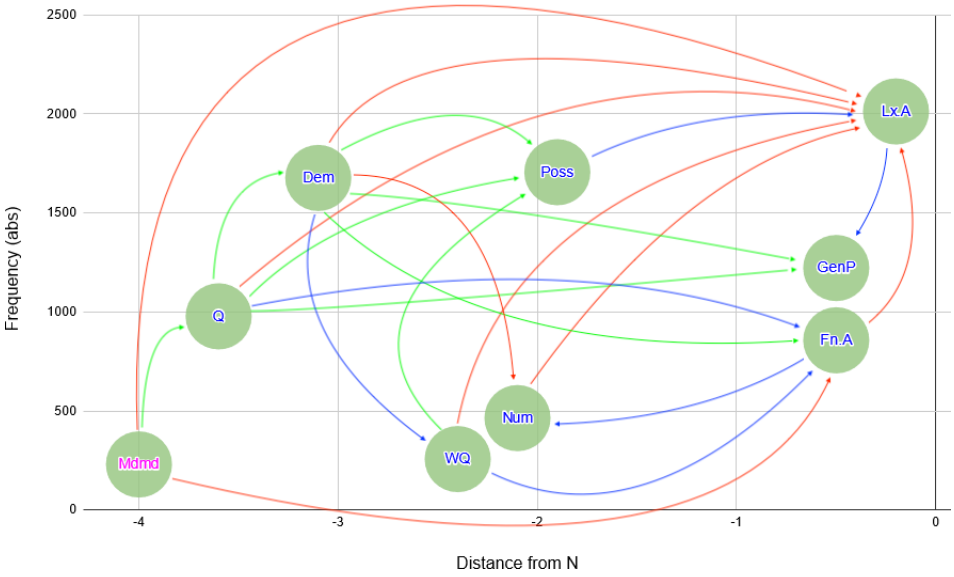

Figure 1: Categorial distribution in the Cartesian plane (Old Icelandic)

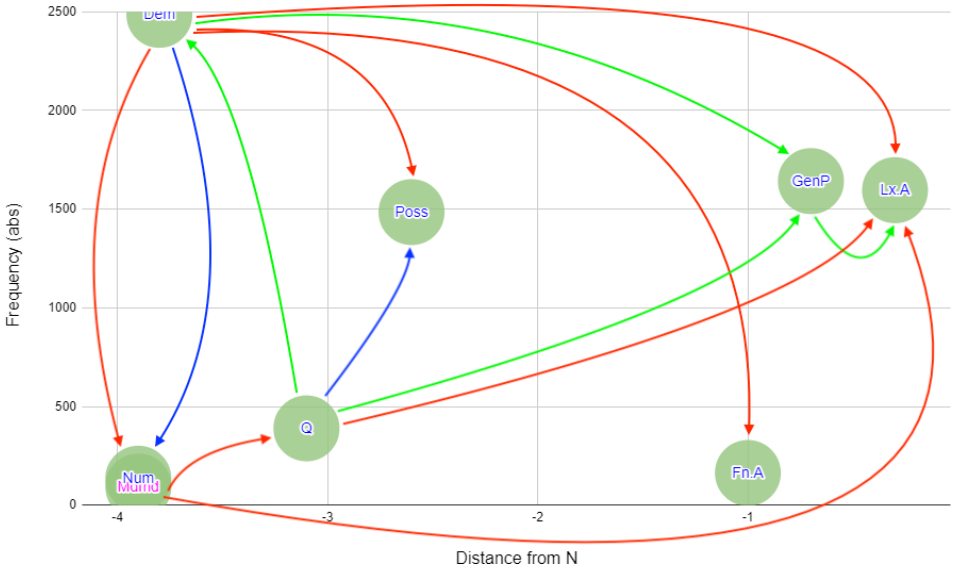

Figure 2: Categorial distribution in the Cartesian plane (Old Saxon)

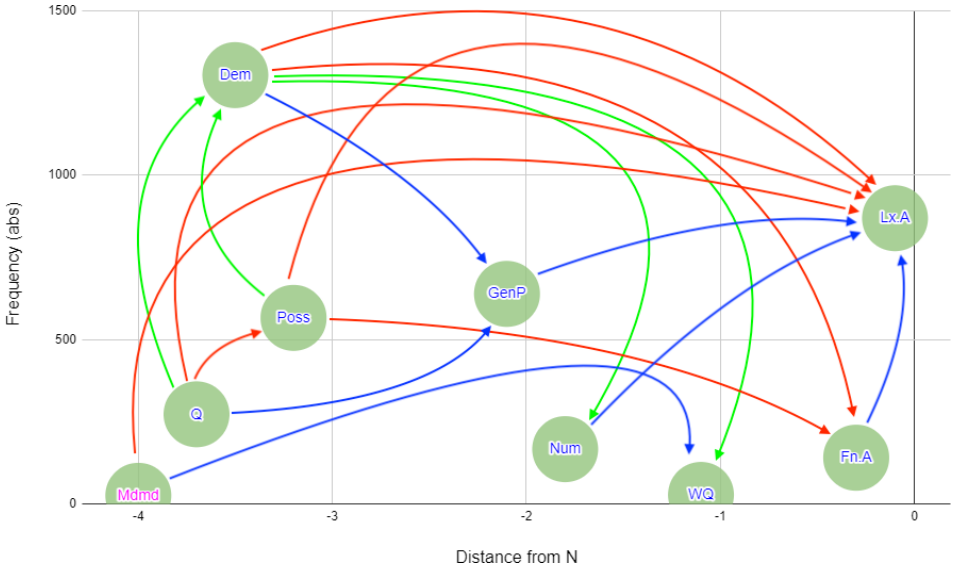

Figure 3: Categorial distribution in the Cartesian plane (Old English)

"Distance from the noun" (= position along the *x*-axis) is an abstract value without a concrete (or discrete) structural counterpart; it does not neatly map onto position or precedence, even though it is calculated on the basis of positional relations between individual categories. As shown in Figure 1, for instance, lexical adjectives have a somewhat lower score than genitive phrases, but the former precede the latter in the few instances of co-occurrences, similarly, for functional adjectives and numerals. In other words, this distance value does not translate to precedence relations.[19]

Presumably, co-occurrence frequency should be given greater prominence since it allows us to assess the generality of the precedence relation. After all, if there is only one co-occurrence of A and B, the precedence ratio is trivially 100%, but this may not always be very insightful. Since we are only considering NPs with at least two prenominal categories here, there are no isolated categories in these diagrams, i.e. categories that are not connected by an arrow. For simplicity, co-occurrence frequency is indicated by the colour code, but it could also be represented separately: for any two categories A and B that are connected by an arrow, the pair (A, B) is mapped onto the number of their co-occurrences, which could be represented as a value along the *z*-axis thus rendering a three-dimensional representation. I have refrained here from doing so mostly for practical reasons, because there are limits as to how much information can be visualized simultaneously.

In the same fashion, the postnominal space or the entire NP can be visualized. For the latter case, there are two possible scenarios: (i) the prenominal and the postnominal spaces are combined, or (ii) the scores are calculated on the basis of the numbers in Tables 15 and 16. In scenario (i), several categories will show up twice, prenominally and postnominally. Moreover, the two spaces do not communicate, and precedence relations across N (x=0) are trivial because prenominal material always precedes postnominal material. In scenario (ii), each category occurs once, and all potential precedence relations between categories are captured. However, we lose, the nominal anchoring restriction; in other words, there is no distance from the noun, but merely distance from first or final position.

Even though (several aspects of) this method can be refined in various ways, it does give us an insightful way of visualizing categorial distribution. Provided

[19] As an extreme case, consider Mdmd, which virtually has a perfect score. This is partially due to rounding and does not entail that it necessarily precedes three other categories. In the current setup, it means that it is almost never preceded by another category (the green arrow in Figure 1 indicates that it is sometimes preceded by Q), but it always precedes something else. In particular, Mdmd never occurs adjacent to the noun because there is always at least one intervening category, viz. the modified modifier, cf. *very *(big/many) horses*; this latter observation is highlighted above by a different font colour.

the dataset is large enough, the diagrams in Figures 1-3 can be seen as the "fingerprints" of the prototypical NP in the respective language (or at least, in a given database or text). Clearly, these fingerprints are different, not merely due to their distance scores, see Table 19, but also in terms of category frequency, see Table 4, and co-occurrence frequency. In other words, categorial distribution as illustrated in Figures 1–3 allows us to graphically represent distributional differences between languages, and, by extension, to visualize syntactic diversity itself.

6 Summary

I have attempted to show that there are more sophisticated ways of diagnosing and quantifying word order variation in the noun phrase than merely comparing prenominal vs. postnominal occurrences of certain elements. Based on the itself rather unspectacular notion of a pattern and some simple mathematical operations, we have given a numerical expression to various dimensions and limitations of syntactic diversity, versatility and probabilistic distribution of categories.

As has already been suggested, almost every aspect of *Patternization* can be modified and refined in various ways. For one thing, the components of patterns were characterized as "formal objects", which allows for patterns to include, apart from category/part-of-speech labels, e.g. morphological or semantic information (depending on the annotated information available in the source database). In other words, there is room for a more complex pattern architecture than the one we have used here.

The focus on noun phrase patterns in this chapter is due to the fact that this work emerged from the NPEGL project, but obviously, nothing prevents us from patternizing VPs or clauses in the same fashion. Even though the patterns may become more complex or larger, the methods for calculating PattDiv or CombFlex will be the same. We are not even obliged to merely consider constituents as the framework for patterns; in principle, any sequence can serve that purpose. We have already seen how the NP can be divided into a prenominal and a postnominal field even though neither is a constituent. Nonetheless, both can be patternized and processed in the same fashion as the NP as a whole. Even though not shown here, we can also determine PattDiv and CombFlex e.g. for the postnominal space alone.

Finally, the procedures and methods described here are, of course, not dependent on the NPEGL annotation, but are applicable more widely. The minimal prerequisite for Patternization is that a given database has at least some part-of-

speech annotation, and, when comparing two datasets, that they be annotated with the same set of labels and according to the same criteria.

I will leave further explorations to future work.

Abbreviations

+/-Att	attestation value
C_{att}	attested combination
C_{pot}	potential combination
cat^n	(sub)category at level n
CombFlex	combinatorial flexibility
μ-CombFlex	mean combinatorial flexibility
M-pattern	matched pattern
ndb	working database
$patt^n$	pattern at level n
PattDiv	pattern diversity
PosProb	probability of a category occurring in a given position
S-pattern	search pattern
S_{cat}	sample space of category labels
scd	standardized common denominator

NPEGL annotation labels

Dem	demonstrative
CC.Fi	finite complement clause
GenP	genitive phrase
Md	modifier
Md.Aj	adjective
Md.Aj.Fn	functional adjective
Md.Aj.Lx	lexical adjective
Md.Card	cardinal element
Md.Card.Num	numeral
Md.Card.WQ	weak quantifier
Mdmd	modifier of modifier
N.C	common noun
PP	prepositional phrase
Q	quantifier
RC	relative clause

Appendix

In this section, I will briefly discuss some functionalities of (the Python-based tool) *Patternization*. Patternization takes the individual annotated databases in NPEGL as input and returns *database objects*. Those objects provide some default constants, e.g. database size and a list of all annotated NPs in the database (i.e. the database itself), and a number of methods with various parameters and default settings to analyze and process the contents of the database. Some methods are described below; this is not an exhaustive list, and I will merely address issues that are pertinent to the above discussion.

Working databases

- **`restrict_Val`**(val, present=True)

This method restricts the current database in accordance with certain specifications: the argument *val* can be a category label, but also a semantic or morpho-syntactic feature, or even a lemma. The argument *present* determines whether *val* must be present or not. The ndb-restriction is encoded via `restrict_Val`("N.C", present=True) AND `restrict_Val`("&", present=False). This procedure is actually a simple query and the modified working database (= ndb) can be taken to be an output in its own right, but the method is recursive, and the modified database has the same functionalities as the original one. That means an output of `restrict_Val` can be restricted further or processed otherwise.

Categories and patterns

- **`Categorize`**(level=2)
- **`Patternize`**(level=2)
- **`Cat_in_Patt`**(cat, level=2)

These methods check the basic inventory of the current working database: `Categorize` returns all attested categories and `Patternize` all attested patterns (i.e. NP types, not tokens). The parameter *level* specifies cat^{level} (default: cat^2). `Cat_in_Patt` returns all patterns in which a given category *cat* occurs, cf. Table 4. The number of patterns and categories can be concomitantly retrieved via the Python in-built function `len`().

Pattern Diversity

- **PattDiv**(level=2, x=False, rnd=False, run=100, size=1000)
- **Randomize**(size=1000)

The method PattDiv with the default setting rnd=False calculates PattDiv as *patterns per NP*; see Section 2, Table 5. But as noticed in that section, this ratio plausibly requires a standardized common denominator, e.g. scd = 1000. The method Randomize() creates a randomized sub-database randomDB from the current working database with the default size 1000 NPs (=scd). We can now calculate $p/1000$ with p = number of patterns in a given randomDB. Due to the randomness involved, however, we are bound to get different values for p for different randomDBs. One straightforward way to establish a representative value for p is to run the procedure a sufficiently large number n of times and calculate the mean value μ_p as follows (with p_i = number of patterns in sub-database randomDB_i):

$$\mu_p = \frac{1}{n}\sum_{i=1}^{n} p_i \quad \Rightarrow \quad \text{PattDiv} = \frac{\mu_p}{\text{scd}} \tag{13}$$

The method PattDiv with the setting rnd=True does exactly that. The parameter *run* specifies the number n of repetitions, and calculates PattDiv according to (13). Obviously, the larger the value n, the more precise is the value for μ_p. There is, however, a practical (computational) problem. In a perfect world, we we should consider all possible sub-databases in order to get the most balanced value μ_p, but this is impossible. For instance, ndb_{OSax} contains 6696 NPs, so we would have $\binom{6696}{1000}$ sub-databases to take into consideration, which is a number with more than 1000 digits. Therefore, an exhaustive procedure is unrealistic. The results in Table 6 are based on the setting (rnd=True, run=500), which already returns a relatively good and stable approximation.

S-patterns and Combinatorial Flexibility

- **precise_pattern**(np, *cats)
  (likewise: **rigid_pattern**, **flexi_pattern** . . . = S-patterns, see Section 4)
- **CombFlex**(samspac, long=3, restrict="N.C", func=precise_pattern,
  count=bool, threshold=1, group_threshold=2)

The methods to diagnose S-patterns such as `precise_pattern` take an NP as a first and a sequence of category labels (i.e. a pattern) as a second argument. They return True if the NP satisfies the specification of the S-pattern (Table 14) in question.

`CombFlex` is a rather complex method, but essentially performs the procedure described in Section 3.2 to determine combinatorial flexibility. The only mandatory argument is *samspac*, which takes a list of category labels as input and thus establishes the sample space. In a first step, it will generate all combinations of length *long*, and if the argument *restrict* is specified (by default "N.C"), it will sort out those combinations that do not satisfy the restriction (here: contain "N.C"). It then generates the respective permutation groups from the combinations remaining. In a next step, it browses the current working database examining every individual NP. Every permutation generated constitutes an S-pattern specified by the parameter *func* (by default, `precise_pattern`). Essentially, the output of `CombFlex` is the number of times the method *func* yields True for each permutation, with permutations sorted into permutation groups. By default, this is encoded as Boolean values, as illustrated in (6) via the setting (count=bool); the alternative setting (count=int) gives the actual number of OCCURRENCES for each individual permutation.

The output can be modified by establishing a threshold value: the parameter *threshold* determines the minimal number of OCCURRENCES required in order for a given permutation to be considered TRUE (= +ATT; see the discussion in section 3). Similarly, the parameter *group_threshold* determines the minimal number of OCCURRENCES required within a permutation group, and can serve as a fine-tuning mechanism. Plausibly, *group_threshold* $\geq$ *threshold*. If a given pattern/permutation occurs less than *threshold* times, it is assigned the value FALSE (–ATT), and if there are less than *group_threshold* OCCURRENCES within a given permutation group, that permutation group will not be part of the output (i.e. that permutation group will not be in C_{att}).

Ranking positions and distance from the noun

- **`rankFirst/rankLast`**(level=2, part=-1)
- **`I_precede_cats/I_follow_cats`**(level=2, part=-1, pair=True)
- **`Probabilize`**(level=2, part=-1)

The ranking methods perform the procedure described in Section 5.1: they count category occurrences according to their position, where `rankFirst` starts with the first position and proceeds to the right (= left-aligned) and vice versa

for `rankLast` (= right-aligned). The parameter *part* determines which partition of the NP is to be considered: a negative value identifies the prenominal space thus producing output as displayed in Table 17, a positive value the postnominal space, cf. Table 18, and the value 0 the entire NP, see Tables 15 and 16.

The precedence methods `I_precede_cats/I_follow_cats` calculate for each category cat_A which other categories cat_{B_n} it precedes/follows, and how often. The parameter *pair* determines whether general precedence (A, . . . B) is to be counted (*pair*=False), or whether only immediate precedence (A,B) is to be considered (the default setting *pair*=True). The precedence scores graphically represented (with colours) in Figures 1-3 are based on `I_precede_cats`(part=-1, pair=True).

Finally, the method `Probabilize` calculates the distance-from-N scores (see Section 5.2) with a simple method that glosses over some complications. It considers only patterns of len > 2; for the setting *part*=0 (entire NP), this is a given, but when considering the pre- or postnominal space, it means that NPs with only one pre-/postnominal category are ignored. Each category occurrence is assigned a score depending on its relative position and pattern length in relation to a common multiple of all pattern lengths. The scores are added up per column, and once the procedure is completed, the category score is divided by the number of category occurrences in the respective column. In addition, I have appended a factor that renders the maximum score as equal to the maximum of columns (in the examples used in this chapter, it was 4), but nothing hinges on that. The scores in Table 19 are calculated with this method.

As mentioned, this is a rather simple method to calculate a mean distance value, and there are certainly more sophisticated ways. However, in several alternatives, the scores accumulate around the middle score (i.e. ca. 2.0) and hardly show any spread, which would not be a very useful basis for assessing precedence relations, and for visualization more generally. Mainly for this reason, the above method was chosen here.

Acknowledgements

This chapter grew out of tinkering with the NPEGL database material, and was originally intended to be a mere appendix to Pfaff & Bouma (2024 [this volume]). Many thanks to Gerlof Bouma for sending me the most recent database files several times. Thanks to Dag Haug and Gerlof Bouma for help with a number of Python-related questions. Finally, I would like to thank two reviewers for commenting on a previous draft of this chapter, which led to substantial improvements and clarifications.

References

Bech, Kristin, Hannah Booth, Kersti Börjars, Tine Breban, Svetlana Petrova & George Walkden. 2024. Noun phrase modifiers in early Germanic: A comparative corpus study of Old English, Old High German, Old Icelandic, and Old Saxon. In Kristin Bech & Alexander Pfaff (eds.), *Noun phrases in early Germanic languages*, 71–109. Berlin: Language Science Press. DOI: 10.5281/zenodo.10641187.

Cinque, Guglielmo. 2005. Deriving Greenberg's Universal 20 and its exceptions. *Linguistic Inquiry* 36(3). 315–332.

Pfaff, Alexander. 2015. *Adjectival and genitival modification in definite noun phrases in Icelandic: A tale of outsiders and inside jobs*. University of Tromsø. (Doctoral dissertation).

Pfaff, Alexander. 2019a. NPEGL: Annotation guidelines. Ms., University of Oslo.

Pfaff, Alexander. 2019b. Reunited after 1000 years. The development of definite articles in Icelandic. *Nordic Journal of Linguistics* 42(2). 1–43. DOI: 10.1017/S0332586519000155.

Pfaff, Alexander & Gerlof Bouma. 2024. The NPEGL noun phrase database: Design and construction. In Kristin Bech & Alexander Pfaff (eds.), *Noun phrases in early Germanic languages*, 1–32. Berlin: Language Science Press. DOI: 10.5281/zenodo.10641183.

Sapp, Christopher. 2019. Arrested development: Case attraction as a transitional stage from Old Icelandic demonstrative to relative *sá*. *Language* 95(1). e1–e40.

Wagener, Terje. 2017. *The history of Nordic relative clauses*. De Gruyter Mouton.