

A New Transliteration Alphabet Brings New Evidence of Word Structure and Multiple "Languages" in the Voynich Manuscript

Massimiliano Zattera¹

¹ *Independent researcher*

Abstract

The question of which glyphs are actual single Voynich characters is still very open; its correct answer will greatly impact statistical analysis of the text and will guide deciphering attempts. This research uses a computational approach, including machine learning techniques, to create a new transliteration alphabet.

It is shown how the structure of Voynich words can be described by assuming each word type is composed of 12 “slots”, each being empty or containing specific glyphs. 86.6% of Voynich tokens exhibit this structure and only 1.5% of tokens are words not following this pattern and appearing at least twice in the text. It is therefore assumed that glyphs appearing in slots are the characters of the Voynich alphabet and as such have been mapped into single characters of a newly created transliteration alphabet.

This alphabet has been used by an optimization algorithm to automatically produce a formal grammar for words in the Voynich. This grammar is the one with the highest F1 score among those surveyed.

Separate grammars have been generated for different sections of the manuscript, and their differences used in a decision tree algorithm which is able to correctly classify pages into their section by looking at the occurrence of four short char sequences. This shows that the difference between sections is not a difference in Voynich vocabulary, as past research based on clustering suggested, rather the vocabularies are different because of differences in inner structure of words. This is evidence for the existence of different “languages” in the manuscript, of which Currier’s A and B are only major divisions.

The implications of these findings are then discussed.

Keywords

Voynich, transliteration_alphabet, word_structure, formal_grammar, Currier_languages

1. Introduction

Manuscript 408 in Yale University Beinecke Library, commonly referred to as the Voynich Manuscript, is a text written using an unknown script. It appears divided in sections, identified by the type of illustrations (or lack thereof) they contain. It is currently debated whether the manuscript is a hoax or if it contains an actual message, encoded using some still unrecognized cypher; needless to say, there is no consensus on the possible language of the plaintext. To answer these questions, several computational techniques have been applied, to support one theory or another.

This article starts by examining some regularities in the structure of words in the text, using them to hypothesize which glyphs should be considered single Voynich characters. These insights are used to create a transliteration alphabet and a formal grammar², based on that alphabet, describing valid strings for the Voynich text. Word features described by the grammar are then shown to be effective indicators

International Conference on the Voynich Manuscript 2022, November 30–December 1, 2022, University of Malta.

EMAIL: mzattera@gmail.com

ORCID: 0000-0001-5305-4270



© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

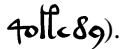
CEUR Workshop Proceedings (CEUR-WS.org)

² Throughout this article, “grammar” refers to a formal grammar describing the inner structure (sequence of glyphs) of words in the Voynich. No assumption is made about the grammar of the underlying plaintext, if any.

when predicting which section of the manuscript a page belongs to. The implications of these regularities are finally discussed.

2. Methods and results

The analysis described in this article has been conducted using v4j, a Java software library that the author created and made available as open-source software³.

The starting point for this work is the Landini-Stolfi interlinear file⁴ [14]. Unless indicated differently, this article uses the EVA alphabet to represent transliterated Voynich words, writing them in Courier font (e.g., qokedy for ). However, the results discussed in section 2.3, and following, have been obtained by processing a text transliterated using the Slot alphabet described later; they are presented here using EVA only for readers' convenience.

A “concordance” version of the interlinear file was created by merging transcriptions from different authors. In this process, characters that were not read by all authors in the same way, or flagged by any author as dubious, were marked as “unreadable” (affecting 8,108 tokens). Rare characters: g, x, v, u, j, b, and z (47 occurrences in total, 13 of them are single-letter words) were also marked as “unreadable”. These transformations reduce the number of characters, tokens and word types that are being processed but at the same time mitigate differences between transcribers and make the overall text more uniform. It must be pointed out that this work aims at modeling regularities in the text, as broadly as possible, but not single exceptions.

As a second step, tokens are created by splitting the text where a space was detected by at least one of the transcribers; there are 31,317 tokens in the text, ignoring those marked as unreadable, corresponding to 5,105 word types.

2.1. “Slots” in the structure of Voynichese words

When analyzing the word types, through a computer-assisted process, some regularities in their structure (the sequence of Voynich glyphs used to write them) began to emerge and are described in the below bullet points and in Figure 2. These rules were found to strike the best balance between simplicity and coverage of the text.

- each word type can be decomposed in 12 “slots”, for convenience numbered from 0 to 11,
- each slot can be empty or contain a single glyph,
- the choice of glyphs that can occupy a slot is very limited and, for 9 out of 12 slots, it is as low as 2-3 possible glyphs,
- each glyph can appear only in one or two slots, with exception of d that can appear in three different slots.

How strictly the words in the Voynich follow these rules is summarized below (see Figure 1):

- **Regular:** 27,114 tokens (86.6% of total), corresponding to 2,617 different word types (51.3% of total), can be decomposed in slots according to the above rules.
- **Separable:** 3,249 tokens (10.4%), corresponding to 1,892 word types (37.1%), can be divided into two parts, each composed by at least two Voynich glyphs, where each part is a regular word type. Moreover, for 2,219 separable word types (75.2% of separable word types), their constituent

³ Voynich for Java (v4j) library, URL: <https://github.com/mzattera/v4j>.

⁴ Other publicly available transliterations with a comparable coverage of the text are those by Landini-Zandbergen (limited to the Zandbergen part) and Glen Claston. These texts were not used, as they might present biases in transliteration, due to single authorship. Claston's text uses Voynich 101 alphabet which contains a hundred symbols, sometimes hardly distinguishable in the manuscript at the naked eye. It seems implausible, though not impossible, that Voynich authors could use such an extend alphabet, not to mention the difficulties in correctly transliterating the text. Most literature on the Voynich is not using this transliteration. Both texts feature more tokens (~36,000) and word types (~8,000) compared to that used in this article, which might constitute a limitation of this study. Notice however that section 2.4 uses a transcription with 38,469 tokens and 8,343 word types, As explained in footnote 17.

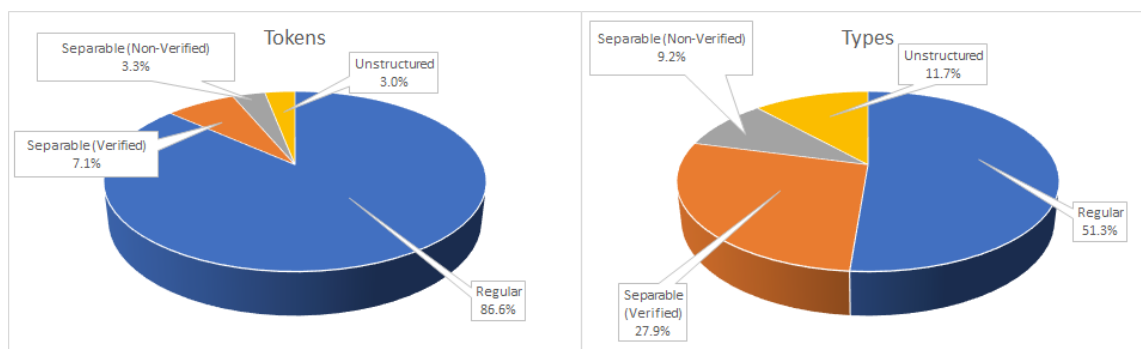


Figure 1: Classification of tokens and word types, based on their “slot” structure.

halves appear as tokens in the text at least as often as the whole word. For example, *chockhy* appears 18 times in the text; it is a separable word type that can be divided as *cho - ckhy*, each part being a regular word type appearing in the text 79 and 39 times respectively. These word types are called “**verified separable**”. This seems an indication that many separable word types are possibly composed words, or two regular words that were written together, or close enough such that the space between them was not transcribed correctly.

- **Unstructured:** Remaining 954 tokens (3.0%), corresponding to 596 different word types (11.7%), are marked as “unstructured”. 489 (82%) of these word types appear only once in the text; by contrast, this percentage is 59.8% for regular and separable word types considered together. This might suggest that unstructured words are either scribal errors, transliteration errors, or special words that are encoded differently than other words; but this is not yet clear at this stage.

2.2. The Slot alphabet

The above results suggest that slots are relevant for the structure of word types, if so, it is reasonable to assume that each glyph appearing in a slot constitutes a basic unit of information; a single character in the Voynich alphabet. Next, the relationships between glyphs, as they appear in slots, and some EVA characters are discussed.

Some glyphs (*t*, *k*, *p*, and *f*) appear taller than other characters and are traditionally referred to as “gallows”. The combination *ch* is often referred as “pedestal”. The glyphs *cth*, *ckh*, *cph*, and *cfh* appear visually as an overlap of the pedestal with one of the gallows and are therefore called “pedestalled gallows”. It has been hypothesized (e.g., [8]) that pedestalled gallows might be a “ligature” of the pedestal and gallows. However:

- The combination of gallows in slot 3 followed by a pedestal in slot 4 is quite common in the text appearing in 2,185 tokens, or 419 regular word types (16% of regular word types) and written explicitly as two glyphs; in these cases, pedestalled gallows could probably have been used instead, as a ligature.
- In 332 tokens, we have a pedestal followed by pedestalled gallows. A ligature would correspond to a double pedestal in a word, which contrasts with the structure suggested by slots; in addition, double pedestals appear only in 17 tokens, suggesting again this to be a very unlikely combination.
- Gallows and pedestalled gallows are preceded by different characters, as the formal grammar described below shows.

All of this suggests pedestalled gallows are Voynich characters in their own, and not ligatures.

In addition, *c* appears outside *ch* or pedestalled gallows only 7 times; similarly, the character *h* appears outside of *ch*, *sh* or the pedestalled gallows only 4 times. This seems a strong indication that *c* and *h* do not correspond to Voynich characters.

The characters *e* and *i* only appear in slots 6 and 9 respectively, in a sequence of 1, 2 or 3. It can be argued that these are indeed repetitions of the same character but, if this is the case, as these sequences

EVA	Slots											
	0	1	2	3	4	5	6	7	8	9	10	11
q	†											
s	2						2					
d	δ							δ			δ	
o		o							o			
y		9									9	
l			8								8	
r			9								9	
t				††								
k				††								
p				††								
f				††								
ch					εε							
sh					εε							
cth						††ε						
ckh						††ε						
cph						††ε						
cfh						††ε						
e							ε					
ee							εε					
eee							εεε					
a								α				
i									ι			
ii									ιι			
iii									ιιι			
m										ξ		
n											9	

Figure 2: “Slots” and glyphs allowed to appear in each position.

Slot	EVA	Tiltman	Bennett	FSG	Frogguy	Currier
q	q	4	D	4	4	4
s	s	2	Z	2	s	2
d	d	8	S	8	8	8
o	o	O	O	O	o	O
y	y	G	G	G	9	9
l	l	E	L	E	x	E
r	r	R	Q	R	2	R
t	t	H	H	H	qp	P
k	k	D	K	D	lp	F
p	p	H	P	P	qj	B
f	f	D	F	F	lj	V
C	ch	T	CT	T	ct	S
S	sh	S	ET	S	c't	Z
T	cth	HZ	CHT	HZ	cqpt	Q
K	ckh	DZ	CKT	DZ	clpt	X
P	cph	HZ	CPT	PZ	cqjt	W
F	cfh	DZ	CFT	FZ	cljt	Y
e	e	C	C	C	c	C
E	ee	CC	CC	CC	cc	CC
B	eee	CCC	CCC	CCC	ccc	CCC
a	a	A	A	A	a	A
i	i	I				
J	ii	II	Varies			
U	iii	III				
m	m	E		IK	ig	J
n	n	L	U	L	v	D

Figure 3: The Slot alphabet compared with other transliteration alphabets.

appear always in same slots, what it is relevant here would be the number of repetitions. Using an example with Roman numerals, the sequence “III” must not be understood as a 3-character word, rather as the number “3”. Therefore, it is assumed each such sequence constitutes a single character.

Drawing from all the above considerations, a new transliteration alphabet is proposed in Figure 3, which, for obvious reasons and a clear lack of imagination, has been called the Slot alphabet.

2.3. A grammar for Voynichese

If slots are a meaningful feature of Voynich words, and if Slot alphabet captures the Voynich alphabet, it should be possible to use them to produce a compact description of Voynich words. To explore this hypothesis, an ad-hoc algorithm⁵ has been used to create a formal grammar for the word types in the Voynich; its input was a “concordance” transliteration made using the Slot alphabet⁶; non-regular word types were ignored by the algorithm. The resulting grammar is shown in Figure 4.

2.3.1. Comparison with previous works

As early as the initial analysis by Tiltman [8], it has been clear that “each symbol behaves as if it had its own place in an order of precedence within words”. Several authors have proposed formal

⁵ The algorithm creates a context-free grammar which is then “pruned” removing production rules to optimize its F1; its code is available in the v4j library.

⁶ https://github.com/mzattera/v4j/blob/master/eclipse/io.github.mzattera.v4j/src/main/resources/Transcriptions/Interlinear_slot_ivtff_1.5.txt

```

<BEGIN>:
-> 0_d, 0_q, 0_s, 1_o, 1_y, 2_l, 2_r,
3_tpkf, 4_C, 4_S, 5_TPK, 7_d, 8_a

0_d:
d -> 4_C, 4_S
0_q:
q -> 1_o
0_s:
s -> 4_C

1_o:
o -> 2_r, 3_tpkf, 4_C, 5_TPK, 6_eEB, 7_d,
8_a
1_y:
y -> 3_tpkf, 4_C, 4_S

2_l:
l -> 3_tpkf, 4_C, 4_S
2_r:
r -> 8_a

3_tpkf:
t, p, k, f -> 4_C, 6_eEB, 8_a, 8_o, 11_y

4_C:
ch -> 6_eEB, 8_a, 8_o, 10_d, 11_y
4_S:
sh -> 6_eEB, 8_o

5_TPK:
cth, cph, ckh -> 6_eEB, 8_a, 8_o

6_eEB:
e, ee, eee -> 7_s, 8_o, 10_d, 11_y, <END>

7_d:
d -> 8_a, 8_o
7_s:
s -> <END>

8_a:
a -> 9_iJ, 10_l, 10_m, 10_n, 10_r
8_o:
o -> 10_d, 10_l, 10_r, <END>

9_iJ:
i, ii -> 10_n, 10_r

10_d:
d -> 11_y, <END>
10_l:
l -> 11_y, <END>
10_m:
m -> <END>
10_n:
n -> <END>
10_r:
r -> <END>

11_y:
y -> <END>

<END>:

```

Figure 4: Proposed formal grammar⁷ for words in the Voynich.

Slots											
0	1	2	3	4	5	6	7	8	9	10	11
+	o	κ	ff	ε	ff	ε	2	o	∖	δ	9
2	9	2	ff	ε	ff	ε	δ	a	∖	κ	
δ			ff		ff	ε		∖	2		
			ff		ff					δ	
										2	

Crust
 Mantle
 Core

Figure 5: Mapping of glyphs appearing in each slot into crust, mantle, and core layers.

grammars to capture these regularities; one can see the works from Roe [12], Palmer [6], Vogt [11], and Pelling [6].

Neal [5] published a concept very similar to the slot structure and a new transliteration scheme (NEVA); his point was that this could be the result of using a grille to produce the text, something like the more complete approach described by Rugg [10].

Stolfi is the proponent of the well-known “crust-mantle-core” decomposition [7]; in this model, each Voynich word can be divided into three layers such that the core is at the center of words, surrounded by the mantle, which in turn is surrounded by the crust. Each layer can be optionally empty and is, in general, defined by the letters it contains, as shown in Figure 5. This high-level structure “must correspond to a major feature of the VMS encoding or of its underlying plaintext”. However, by comparison with the slots model, this structure is not so evident. For example, the crust is not homogeneous as it is composed of a left and a right part which are quite different (see e.g., position of q or i). Similarly, gallsows in slots 3 and 5, which belong to the core layer, could well enclose pedestals in slots 4 that are classified as mantle (Stolfi admits: “the implied structure of the mantle is probably the weakest part of our paradigm”). At a finer level of detail, the grammar gets complex when parsing sequences of a, o, e, and y, introducing rules that Stolfi agrees are somewhat arbitrary. For 86.6% of

⁷ A rule like:

```

S:
  a, b -> X, Y

```

means that S can generate any of the two strings a or b and then be replaced by any of the rules X or Y. Performing all the possible replacements from the initial rule (<BEGIN>) to the final one (<END>) will generate all the strings for the grammar.

tokens, classified as regular by the slot model, this seems an unnecessary complication as *a*, *o*, and *y*, can be unambiguously assigned to the crust layer. A similar case can be made for *e* sequences, which appear in very definite positions. Finally, the grammar can generate over 96.5% of all the tokens in the text but it does this at the cost of generating an almost infinite number of non-Voynichese strings (~1.4e20 different word types).

Cham [1] proposes a new pattern named the “Curve-Line System” (CLS). This pattern is based on shapes of individual glyphs and the order in which they appear in words. The formal grammar described in this article aligns and independently confirms this work. Indeed, words created by the grammar follow the CLS pattern. The only exceptions are created by rules *1_o*, *8_o*, *2_r*, *2_l*, *10_l* and correspond to Cham’s “aberrant glyphs” he notices and describes in his work.

2.3.2. Grammars as classifiers

The grammars described above can be used as classifiers to discriminate strings belonging to the Voynich, which should be the ones and only generated by a grammar, from others. If this classification is sound, then the grammar provides an accurate description of words in the Voynich, which could be used to identify regularities in the text. It must be clear that regularities in the Voynich do not necessarily reflect regularities in a hypothetical plaintext. However, they could lead a step further in the decipherment of the manuscript.

In practice, there are four possible cases for each word type *T* being classified, shown in Table 1:

Table 1

Possible results for a classification task.

	Word type <i>T</i> is generated by the grammar	Word type <i>T</i> is NOT generated by the grammar
Word type <i>T</i> appears in the Voynich	True Positive (TP)	False Negative (FN)
Word type <i>T</i> does NOT appear in the Voynich	False Positive (FP)	True Negative (TN)

To evaluate how good a grammar is in this discrimination task, below metrics are commonly used:

$$Precision = True\ Positives / All\ word\ types\ generated = TP / (TP + FP)$$

$$Recall = True\ Positives / All\ word\ types\ in\ the\ Voynich = TP / (TP + FN)$$

$$F1 = 2 * precision * recall / (precision + recall)$$

F1 score tends to 0 if either precision or recall tend to 0, while it tends to 1 if both recall and precision tend to 1, indicating in this case that all and only words in the Voynich are generated by the grammar.

Table 2 compares our grammar with others found in literature, showing it exhibits the highest F1 score. It successfully models 62% of tokens that appear in the Voynich and 1,113 word types (21.6%). This result was achieved based on the Slot alphabet, which in turn is derived from the slot model, and supports the hypothesis put forward in section 2.3 above.

Table 2

Precision, Recall, and F1 score for several grammars.

Grammar	Total Generated Strings	True Positives	Precision	Recall	F1 Score
ROE [12]	120	112	0.933	0.022	0.043
PALMER ⁸ [6]	∞	4,547	0	0.884	0
PELLING ⁹ [6]	∞	259	0	0.05	0
PELLING_2 ¹⁰	1,192	259	0.217	0.05	0.081
VOGT (Recipes) [11]	32,575	424	0.013	0.19	0.024
VOGT	32,575	565	0.017	0.11	0.03

⁸ Assuming Palmer uses the standard notation for regular expressions (* = zero or more instances, + = one or more, ? = zero or one).

⁹ Assuming all arrows have the same meaning, and the red boxes are non-emitting states, the model generates an infinite number of strings.

¹⁰ Same as PELLING but assuming the model generates only 1,192 strings, as claimed by its author.

Grammar	Total Generated Strings	True Positives	Precision	Recall	F1 Score
NEAL_1a ¹¹ [15]	87,480	535	0.006	0.104	0.012
NEAL_1b ¹¹	174,818	1,782	0.01	0.347	0.02
NEAL_2 ¹¹	1,311,345	1,049	0.001	0.204	0.002
STOLFI ¹² [7]	> 1.4 x 10 ²⁰	4,527	0	0.881	0
SLOT ¹³	4,643,467	2,617	0.001	0.509	0.001
SLOT_MACHINE¹⁴	3,110	1,113	0.358	0.216	0.270

2.4. “Languages”¹⁵ in the Voynich

Formal grammars were subsequently generated for different sections of the Voynich (Herbal-A, Herbal-B, Recipes, Biological/Balneological, and Pharmaceutical)¹⁶. Character combinations appearing only in some of these grammars were identified as potentially distinctive for corresponding section. For example, in the grammar for the Recipes section (and not in the one in Figure 4), the production rule 6_eEB can be generated from the initial rule, meaning that sequences like e-, ee-, or eee- at the beginning of words are a potentially marker of the Recipes section.

The percentages of tokens in each section containing such distinctive character sequences have been used as features to train a decision tree algorithm to classify pages in the Voynich into corresponding sections¹⁷. The resulting tree (Figure 6) is surprisingly compact and succeeds in classifying pages with an accuracy of 92%, as shown by corresponding confusion matrix in Figure 7. As a test, the experiment was repeated by assigning each page to one of five random groups; the resulting decision tree had 11 rules (compared to 5) and an accuracy of only 54%, showing that structural features we found are indeed correlated with sections.

This shows that the difference between the sections is not a difference in their Voynich vocabularies (the set of Voynich word types they contain), rather a difference in inner structure of the words (the frequency with which specific glyph sequences appear in tokens); of course, the latter will cause the former, but not necessarily vice-versa. To explain this with an example, given a set of texts in English, one would expect to identify those about meteorology by looking at frequency of terms such as “thunderstorm” in the different texts (difference in vocabulary). However, it would be surprising if such classification could be made only by counting the number of words starting with “th-” (difference in word structure), but this is precisely what happens with the different sections in the Voynich.

This could be an indication that the two Currier’s languages A and B [3] are only a major distinction within a group of (at least) five such languages. Whether these languages are the result of differences in a hypothetical plaintext, or just a mere artifact of the method used to create the manuscript, cannot be decided at this stage.

3. Conclusions

There is clear empirical evidence of an inner structure of words in the Voynich text, which is captured here as the existence of “slots” in words. Slots can contain only some of the glyphs used in the Voynich script; the Slot alphabet is a new transliteration alphabet that maps each such glyph into a single character.

This “slot” model and the Slot alphabet could be new useful tools to analyze the Voynich; in this article, they were used to create a formal grammar of Voynich words which outperforms, in terms of its F1 score, any other grammar proposed in the reviewed literature. This further supports the idea that

¹¹ Three versions of Neal’s model as described in [15].

¹² Assuming any word type in the Voynich that is not listed in Solfi’s *AbnormalWord* rule is a true positive.

¹³ All strings that can be generated following rules described by the slot model in Figure 1.

¹⁴ The grammar described in this article in Figure 4.

¹⁵ The term “language” used in this section is meant, as in Currier, to identify a subset of the Voynich text that shows some particular statistical properties. No assumption is made about the language of the possibly underlying plaintext.

¹⁶ Remaining Astronomical, Cosmological, and Zodiac sections were ignored, as they are not suited for creating features like those described here, since their short text will create a high volatility in features. These sections together form ~10% of the entire text.

¹⁷ To make these percentages less volatile, a “majority” transcription using the Slot alphabet was created. Texts from different transcribers were merged by taking, for each character, the one which was used by the majority of the authors. The resulting text has 38,469 tokens and 8,343 word types.

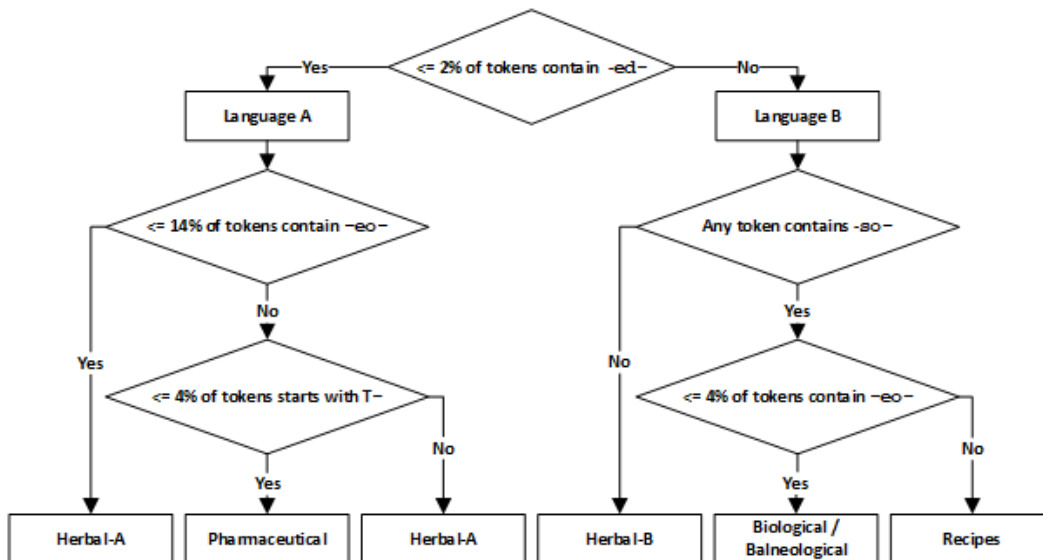


Figure 6: Decision tree to classify Voynich pages into their section.

	Biological	Herbal-A	Herbal-B	Pharmaceutical	Recipes
Biological	19				1
Herbal-A		87		2	
Herbal-B	3		25		2
Pharmaceutical		2		14	
Recipes	2		2		19

Figure 7: Confusion matrix; the row labels show the correct classification for a page, whilst the columns are the labels assigned by the decision tree.

slots are a relevant feature of words and that the Slot alphabet provides an efficient transliteration alphabet.

Finally, by comparing this grammar with grammars specific for each section of the Voynich, features in words were identified which allow assigning a page to its section with high accuracy, just by counting the number of tokens in a page containing a handful of character combinations. This is possibly the first time a formal grammar surfaces previously unrecognized features of words which are relevant in identifying regularities in the text.

The above findings have further implications:

- The Slot alphabet might be better suited than others when a transliteration is used for statistical analysis (or cypher attacks) where a one-to-one mapping between the Voynich characters and those in the transliteration alphabet is crucial.
- Since no natural language exhibits such a slot structure, any simple substitution cypher for the Voynich (e.g., [2]) should at this point be clearly ruled out. This includes cyphers where letters in words are sorted before being replaced, since in this case 1) the number of slots should equal the number of letters in the plaintext alphabet, 2) glyphs would occupy one and only one slot, and 3) repetitions should be frequent.
- The strong regularities imposed by the slot structure clearly affect character level entropy in Voynich words [4][9], which should be considered when using this metric to produce any insight about the text (e.g., to identify the plaintext language).
- The slot structure applies to the entire text; at the same time, the grammar presented here captures the structure of about 62% of tokens in the whole manuscript, an indication that the method used to create the text is somewhat uniform across the different sections. On the other hand, the fact that sections of the text can be identified based on word structure suggests that this general method has somehow been “tweaked” for each section. One could speculate that the method used to create the text, whether it is an actual cypher or a way to generate gibberish, relies on some parameters (e.g., a table, or a grill [10][13]) which are made different for each section. This also suggests that any decipherment attempt, or statistical analysis, should be conducted separately for each section.

- The difference in word structure across the different sections of the text will obviously affect the list of word types appearing in each section; in turn, this will affect any clustering algorithm that uses words as features. Previous works that used clustering to support the idea of different topics in different sections might be just surfacing a difference which is the result of the mechanism used to generate the word types, not a difference in terminology in the plaintext. This is exemplified in Figure 8.

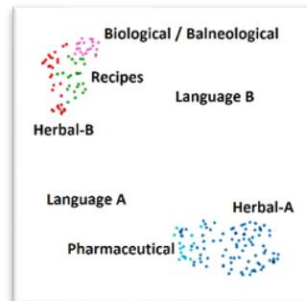


Figure 8: UMAP projection obtained by using same word features used to create the decision tree in Figure 6. Each dot represents a page in the Voynich, its color based on the section where it appears. The figure clearly shows how clustering performed using word features identified by the grammar in Figure 4 replicates results of clustering based on bag of words or TF-IDF approaches.

4. References

- [1] B. Cham, Introduction to the Curve-Line System, 2014. URL: <https://briancham1994.com/2014/12/17/curve-line-system/>.
- [2] G. Cheshire, The Language and Writing System of MS408 (Voynich) Explained, *Romance Studies* 37:1 (2019) 30–67. doi:10.1080/02639904.2019.1599566.
- [3] P. Currier, Some Important New Statistical Findings, in: M. E. D’Imperio (Ed.), *New Research on the Voynich Manuscript: Proceedings of a Seminar*, Washington, D.C., 1976.
- [4] L. Lindemann, C. Bown, Character Entropy in Modern and Historical Texts: Comparison Metrics for an Undeciphered Manuscript, 2020. URL: <https://arxiv.org/abs/2010.14697>. arXiv:2010.14697.
- [5] P. Neal, The NEVA Spaced Transcription, ?. URL: http://philipneal.net/voynichsources/transcription_neva_spaced/.
- [6] N. Pelling, Sean Palmer’s Voynichese Word Generator..., 2010. URL: <http://ciphermysteries.com/2010/11/22/sean-palmers-voynichese-word-generator>.
- [7] J. Stolfi, A Grammar for Voynichese Words, 2000. URL: <https://www.ic.unicamp.br/~stolfi/EXPORT/projects/voynich/00-06-07-word-grammar/>.
- [8] J. H. Tiltman, The Voynich Manuscript “The Most Mysterious Manuscript in the World”, *NSA Technical Journal* 12 (July 1967) 41–85.
- [9] S. Reddy, K. Knight, What We Know About the Voynich Manuscript, in: *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, Association for Computational Linguistics, 2011, pp. 78–86.
- [10] G. Rugg, An Elegant Hoax? A Possible Solution to the Voynich Manuscript, *Cryptologia* 28:1 (2004) 31–46. doi:10.1080/0161-110491892755.
- [11] E. Vogt, Grammar, 2009. URL: <https://voynichthoughts.wordpress.com/grammar/>.
- [12] R. Zandbergen, Analysis Section (3/5) - Word Structure, 2019. URL: http://www.voynich.nu/a3_para.html.
- [13] R. Zandbergen, The Cardan Grille Approach to the Voynich MS Taken to The Next Level, 2021. URL: <https://arxiv.org/abs/2104.12548>. arXiv:2104.12548, 2021.
- [14] R. Zandbergen, Text Analysis - Transliteration of the Text, 2022. URL: <http://www.voynich.nu/transcr.html>.
- [15] M. Zattera, Note 006 - Works on Word Structure, 2022. URL: <https://mzattera.github.io/v4j/006/>.