

arseighttranscripts

November 17, 2024

1 Algorithmic Recursive Sequence Analysis 2.0

1.1 Grammar induction from further transcripts after extensive sequence analysis of a transcript

This document presents the method of algorithmically recursive sequence analysis. The goal of this method is to extract grammatical structures from natural language sequences to extract and then induce a grammar based on it. We focus on them Analysis of transcripts prepared through extensive sequence analysis to recognize the underlying rules and patterns in the data.

1.1.1 Background

Sequence analysis in natural language data is a widely used approach in the Language processing to identify recurring patterns, structures and dependencies between to identify different parts of a sequence. In this extended version (2.0) we integrate grammatical induction as a way from the analyzed to derive a formal grammar from sequences.

1.1.2 Goal setting

The main objectives of this approach include:

1. **Extension of sequence analysis:** The analysis is applied to a broader set of transcripts to test whether the discovered patterns can be generalized beyond a single transcript.
2. **Grammar induction:** The knowledge gained from the sequences is used to extract grammatical rules and develop formalized grammars.
3. **Quality Evaluation:** The quality of the induced grammar is evaluated based on its ability to generate future sequences that match the patterns in the transcripts.

1.1.3 Methodology

The method consists of several steps that are carried out iteratively:

1. **Data preparation:** Collection and preprocessing of transcripts.
2. **Sequence analysis:** Carrying out an extensive sequence analysis with reading production and reading falsification on a transcript (loc. cit. in this repository). Conduct an extensive analysis of the transcripts, identifying statistical and syntactic patterns.
3. **Grammar induction:** Based on the identified patterns, a formal grammar (Python here in the document) is induced that can describe the generated sequences.

4. **Optimization:** The grammar is optimized (Python here in the document) and validated to increase its accuracy and generalizability.

After evaluating specialist literature on sales discussions, the following preliminary hypothetical grammar emerges: A sales conversation (VKG) consists of a greeting (BG), a sales part (VT) and a farewell (AV).

- The greeting consists of a greeting from the customer (KBG) and a greeting from the seller (VBG).
- The farewell consists of a farewell by the customer (KAV) and a farewell by the seller (VAV).
- The sales part consists of a requirements part (B) and a final part (A).
 - The needs part includes a needs clarification (BBd) and a needs argument (BA).
 - * The needs clarification consists of the customer's needs (KBBd) and the associated clarifications from the seller (VBBd).
 - * The needs argument consists of arguments from the customer (KBA) and the seller (VBA).
 - The closing part (A) consists of objections (AE) and a sales deal (AA).
 - * The objections consist of arguments from the customer (KAE) and the seller (VAE).
 - * The sales conclusion consists of arguments from the customer (KAA) and the seller (VAA).

The farewell (AV) consists of a farewell to the customer (KAV) and the seller (VAV).

The start symbol is therefore VKG, and the terminal characters for the categories are: KBG, VBG, KBBd, VBBd, KBA, VBA, KAE, VAE, KAA, VAA, KAV and VAV. From this information, create a probabilistic context-free grammar with initially assumed transition probabilities.

First, a transcript was created from an intensive sequence analysis with readings production and Reading falsification, category formation, intercoding correlation and grammar induction with schema and grammar transduction with Lisp and parsing the Terminal chain carried out with Object Pascal (loc. cit. in this repository).

The procedure can be divided into several clearly defined steps. Here's an overview to make sure all steps are understandable:

1.1.4 1. Formation of hypotheses about grammar

- Conduct a preliminary investigation to create a **hypothetical PCFG**. This serves as a guide, but is not binding for the analysis of the transcripts.

1.1.5 2. Analysis of the transcripts

- Each **transcript is analyzed individually**. The intentions of the speakers and the structure of the interactions are identified.
- Each interaction in the transcript is assigned a **terminal character**. This results in a **terminal string** for each transcript.

1.1.6 3. Induction of a grammar for each transcript

- A specific **grammar is induced** from the terminal string of each transcript.
- There is a separate **grammar per transcript** at the end (eight in total).

1.1.7 4. Unification of Grammars

- The eight individual grammars are merged into a **unified grammar** that covers the structure of all transcripts.

1.1.8 5. Parsing terminal strings

- The eight terminal strings of the transcripts are checked for **well-formedness** with respect to the unified grammar. This means that each string should be correctly recognized and parsed by the grammar.

1.1.9 6. Optimization of PCFG

- A first version of a PCFG is created from the previous analyses.
- With the created PCFG, **artificial terminal strings** are created.
- The **proportional frequency of the terminal characters** from the artificial strings is compared with the frequency of the terminal characters in the real data (the eight transcripts).
- The **significance** of this correlation is measured.
- Adjustments to the PCFG are made until the correlation is satisfactory.

1.1.10 7. Repeat Process

- Steps 6 and 7 are repeated until there is a good fit between the **empirical data** and the generated PCFG.

Here is a probabilistic context-free grammar (PCFG) based on structure. In a PCFG, probabilities for transitions between different production rules are defined. Since you just provided a structure without specific probabilities, I'll assume equal probabilities for the possible options within the same level to provide a baseline. These probabilities can be adjusted later when empirical data becomes available.

2 Probabilistische hypothetische kontextfreie Grammatik (PCFG) für ein Verkaufsgespräch

VKG -> BG VT AV [1.0]

3 Begrüßung

BG -> KBG VBG [1.0] KBG -> 'Kunden-Gruß' VBG -> 'Verkäufer-Gruß'

4 Verabschiedung

AV -> KAV VAV [1.0] KAV -> 'Kunden-Verabschiedung' VAV -> 'Verkäufer-Verabschiedung'

5 Verkaufsteil

VT -> B A [1.0]

6 Bedarfsteil

B -> BBd BA [1.0]

7 Bedarfsklärung

BBd -> KBBd VBBd [1.0] KBBd -> 'Kunden-Bedarf' VBBd -> 'Verkäufer-Klä rung'

8 Bedarfsargumentation

BA -> KBA VBA [0.5] | VBA KBA [0.5] KBA -> 'Kunden-Argument' VBA -> 'Verkäufer-Argument'

9 Abschluss

A -> AE AA [1.0]

10 Einwände

AE -> KAE VAE [0.5] | VAE KAE [0.5] KAE -> 'Kunden-Einwand' VAE -> 'Verkäufer-Einwand'

11 Verkaufsabschluss

AA -> KAA VAA [0.5] | VAA KAA [0.5] KAA -> 'Kunden-Abschluss' VAA -> 'Verkäufer-Abschluss'

The eight Transcripts (Audio file here in Repository).

11.0.1 Text 1

Datum: 28. Juni 1994, **Ort:** Metzgerei, Aachen, 11:00 Uhr

(Die Geräusche eines geschäftigen Marktplatzes im Hintergrund, Stimmen und Gemurm el)

Verkäuferin: Guten Tag, was darf es sein?

Kunde: Einmal von der groben Leberwurst, bitte.

Verkäuferin: Wie viel darf's denn sein?

Kunde: Zwei hundert Gramm.

Verkäuferin: Zwei hundert Gramm. Sonst noch etwas?

Kunde: Ja, dann noch ein Stück von dem Schwarzwälder Schinken.

Verkäuferin: Wie groß soll das Stück sein?

Kunde: So um die dreihundert Gramm.

Verkäuferin: Alles klar. Kommt sofort. *(Geräusche von Papier und Verpackung)*

Kunde: Danke schön.

Verkäuferin: Das macht dann acht Mark zwanzig.

Kunde: Bitte. *(Klimpern von Münzen, Geräusche der Kasse)*

Verkäuferin: Danke und einen schönen Tag noch!

Kunde: Danke, ebenfalls!

Ende Text 1

11.0.2 Text 2

Datum: 28. Juni 1994, **Ort:** Marktplatz, Aachen

(Ständige Hintergrundgeräusche von Stimmen und Marktatmosphäre)

Verkäufer: Kirschen kann jeder probieren hier, Kirschen kann jeder probieren hier!

Kunde 1: Ein halbes Kilo Kirschen, bitte.

Verkäufer: Ein halbes Kilo? Oder ein Kilo?

(Unverständliches Gespräch, Münzen klimpern)

Verkäufer: Danke schön!

Verkäufer: Kirschen kann jeder probieren hier! Drei Mark, bitte.

Kunde 1: Danke schön!

Verkäufer: Kirschen kann jeder probieren hier, Kirschen kann jeder probieren hier!

(Weitere Stimmen im Hintergrund, unverständliches Gespräch, Münzen klimpern)

Kunde 2: Ein halbes Kilo, bitte.

(Unverständliches Gespräch)

Ende Text 2

11.0.3 Text 3

Datum: 28. Juni 1994, **Ort:** Fischstand, Marktplatz, Aachen

(Marktatmosphäre, Gespräch im Hintergrund, teilweise unverständlich)

Kunde: Ein Pfund Seelachs, bitte.

Verkäufer: Seelachs, alles klar.

(Geräusche von Verpackung und Verkaufsvorbereitungen)

Verkäufer: Vier Mark neunzehn, bitte.

(Geräusche von Verpackung, Münzen klimpern)

Verkäufer: Schönen Dank!

Kunde: Ja, danke schön!

Ende Text 3

11.0.4 Text 4

Datum: 28. Juni 1994, **Ort:** Gemüsestand, Aachen, Marktplatz, 11:00 Uhr

(Marktatmosphäre, teilweise unverständlich)

Kunde: Hören Sie, ich nehme ein paar Champignons mit.

Verkäufer: Braune oder helle?

Kunde: Nehmen wir die hellen.

Verkäufer: Alles klar, die hellen.

(Unverständliche Unterhaltung im Hintergrund)

Verkäufer: Die sind beide frisch, keine Sorge.

Kunde: Wie ist es mit Pfifferlingen?

Verkäufer: Ah, die sind super!

(Unverständliches Gespräch)

Kunde: Kann ich die in Reissalat tun?

Verkäufer: Eher kurz anbraten in der Pfanne.

Kunde: Okay, mache ich.

Verkäufer: Die können Sie roh verwenden, aber ein bisschen anbraten ist besser.

Kunde: Verstanden.

(Weitere Unterhaltung, unverständliche Kommentare)

Verkäufer: Noch etwas anderes?

Kunde: Ja, dann nehme ich noch Erdbeeren.

(Pause, Hintergrundgeräusche von Verpackung und Stimmen)

Verkäufer: Schönen Tag noch!

Kunde: Gleichfalls!

Ende Text 4

11.0.5 Text 5

Datum: 26. Juni 1994, **Ort:** Gemüsestand, Aachen, Marktplatz, 11:00 Uhr

(Marktatmosphäre, teilweise unverständlich)

Verkäufer: So, bitte schön.

Kunde 1: Auf Wiedersehen!

Kunde 2: Ich hätte gern ein Kilo von den Granny Smith Äpfeln hier.

(Unverständliches Gespräch im Hintergrund)

Verkäufer: Sonst noch etwas?

Kunde 2: Ja, noch ein Kilo Zwiebeln.

Verkäufer: Alles klar.

(Unverständliches Gespräch, Hintergrundgeräusche)

Kunde 2: Das war's.

Verkäufer: Sechs Mark fünfundzwanzig, bitte.

(Unverständliches Gespräch, Geräusche von Münzen und Verpackung)

Verkäufer: Wiedersehen!

Kunde 2: Wiedersehen!

Ende Text 5

11.0.6 Text 6

Datum: 28. Juni 1994, **Ort:** Käseverkaufsstand, Aachen, Marktplatz

(Marktatmosphäre, Begrüßungen)

Kunde 1: Guten Morgen!

Verkäufer: Guten Morgen!

Kunde 1: Ich hätte gerne fünfhundert Gramm holländischen Gouda.

Verkäufer: Am Stück?

Kunde 1: Ja, am Stück, bitte.

Ende Text 6

11.0.7 Text 7

Datum: 28. Juni 1994, **Ort:** Bonbonstand, Aachen, Marktplatz, 11:30 Uhr

(Geräusche von Stimmen und Marktatmosphäre, teilweise unverständlich)

Kunde: Von den gemischten hätte ich gerne hundert Gramm.

(Unverständliche Fragen und Antworten)

Verkäufer: Für zu Hause oder zum Mitnehmen?

Kunde: Zum Mitnehmen, bitte.

Verkäufer: Fünfzig Pfennig, bitte.

(Klimpern von Münzen, Geräusche von Verpackung)

Kunde: Danke!

Ende Text 7

11.0.8 Text 8

Datum: 9. Juli 1994, **Ort:** Bäckerei, Aachen, 12:00 Uhr

(Schritte hörbar, Hintergrundgeräusche, teilweise unverständlich)

Kunde: Guten Tag!

(Unverständliche Begrüßung im Hintergrund)

Verkäuferin: Einmal unser bester Kaffee, frisch gemahlen, bitte.

(Geräusche der Kaffeemühle, Verpackungsgeräusche)

Verkäuferin: Sonst noch etwas?

Kunde: Ja, noch zwei Stück Obstsalat und ein Schälchen Sahne.

Verkäuferin: In Ordnung!

(Geräusche der Kaffeemühle, Papiergeräusche)

Verkäuferin: Ein kleines Schälchen Sahne, ja?

Kunde: Ja, danke.

(Türgeräusch, Lachen, Papiergeräusche)

Verkäuferin: Keiner kümmert sich darum, die Türen zu ölen.

Kunde: Ja, das ist immer so.

(Lachen, Geräusche von Münzen und Verpackung)

Verkäuferin: Das macht vierzehn Mark und neunzehn Pfennig, bitte.

Kunde: Ich zahle in Kleingeld.

(Lachen und Geräusche von Münzen)

Verkäuferin: Vielen Dank, schönen Sonntag noch!

Kunde: Danke, Ihnen auch!

Ende Text 8

[]:

Auf Grundlage der vorläufigen probabilistischen kontextfreien Grammatik (PCFG) werden die Transkripte in die entsprechenden Terminalzeichen umwandeln. Hier ist die detaillierte Zuordnung für jedes Transkript, wobei für jede relevante Aktion oder Aussage des Gesprächs ein passendes Terminalzeichen aus der vorgegebenen Grammatik verwende.

11.0.9 Transkript 1 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Metzgerei, Aachen, 11:00 Uhr

Transkriptausschnitt	Terminalzeichen
Kunde: Guten Tag	KBG (Kunden-Gruß)
Verkäuferin: Guten Tag	VBG (Verkäufer-Gruß)
Kunde: Einmal von der groben Leberwurst, bitte.	KBBd (Kunden-Bedarf)
Verkäuferin: Wie viel darf's denn sein?	VBBd (Verkäufer-Klärung)
Kunde: Zwei hundert Gramm.	KBA (Kunden-Argument)
Verkäuferin: Sonst noch etwas?	VBA (Verkäufer-Argument)
Kunde: Ja, dann noch ein Stück von dem Schwarzwälder	KBBd (Kunden-Bedarf)
Verkäuferin: Wie groß soll das Stück sein?	VBBd (Verkäufer-Klärung)
Kunde: So um die dreihundert Gramm.	KBA (Kunden-Argument)
Verkäuferin: Das macht dann acht Mark zwanzig.	VAA (Verkäufer-Abschluss)
Kunde: Bitte.	KAA (Kunden-Abschluss)
Verkäuferin: Danke und einen schönen Tag noch!	VAV (Verkäufer-Verabschiedung)
Kunde: Danke, ebenfalls!	KAV (Kunden-Verabschiedung)

11.0.10 Transkript 2 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Marktplatz, Aachen

Transkriptausschnitt	Terminalzeichen
Verkäufer: Kirschen kann jeder probieren hier!	VBG (Verkäufer-Gruß)
Kunde 1: Ein halbes Kilo Kirschen, bitte.	KBBd (Kunden-Bedarf)
Verkäufer: Ein halbes Kilo? Oder ein Kilo?	VBBd (Verkäufer-Klärung)

Transkriptausschnitt	Terminalzeichen
Verkäufer: Drei Mark, bitte.	VAA (Verkäufer-Abschluss)
Kunde 1: Danke schön!	KAA (Kunden-Abschluss)
Verkäufer: Kirschen kann jeder probieren hier!	VBG (Verkäufer-Gruß)
Kunde 2: Ein halbes Kilo, bitte.	KBBd (Kunden-Bedarf)
Verkäufer: Drei Mark, bitte.	VAA (Verkäufer-Abschluss)
Kunde 2: Danke schön!	KAA (Kunden-Abschluss)

11.0.11 Transkript 3 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Fischstand, Marktplatz, Aachen

Transkriptausschnitt	Terminalzeichen
Kunde: Ein Pfund Seelachs, bitte.	KBBd (Kunden-Bedarf)
Verkäufer: Seelachs, alles klar.	VBBd (Verkäufer-Klärung)
Verkäufer: Vier Mark neunzehn, bitte.	VAA (Verkäufer-Abschluss)
Kunde: Danke schön!	KAA (Kunden-Abschluss)

11.0.12 Transkript 4 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Gemüsestand, Aachen, Marktplatz, 11:00 Uhr

Transkriptausschnitt	Terminalzeichen
Kunde: Hören Sie, ich nehme ein paar Champignons mit.	KBBd (Kunden-Bedarf)
Verkäufer: Braune oder helle?	VBBd (Verkäufer-Klärung)
Kunde: Nehmen wir die hellen.	KBA (Kunden-Argument)
Verkäufer: Die sind beide frisch, keine Sorge.	VBA (Verkäufer-Argument)
Kunde: Wie ist es mit Pfifferlingen?	KBBd (Kunden-Bedarf)
Verkäufer: Ah, die sind super!	VBA (Verkäufer-Argument)
Kunde: Kann ich die in Reissalat tun?	KAE (Kunden-Einwand)
Verkäufer: Eher kurz anbraten in der Pfanne.	VAE (Verkäufer-Einwand)
Kunde: Okay, mache ich.	KAA (Kunden-Abschluss)
Verkäufer: Schönen Tag noch!	VAV (Verkäufer- Verabschiedung)
Kunde: Gleichfalls!	KAV (Kunden- Verabschiedung)

11.0.13 Transkript 5 - Terminalzeichen

Datum: 26. Juni 1994, **Ort:** Gemüsestand, Aachen, Marktplatz, 11:00 Uhr

Transkriptausschnitt	Terminalzeichen
Kunde 1: Auf Wiedersehen!	KAV (Kunden-Verabschiedung)
Kunde 2: Ich hätte gern ein Kilo von den Granny Smi	KBBd (Kunden-Bedarf)
Verkäufer: Sonst noch etwas?	VBBd (Verkäufer-Klärung)
Kunde 2: Ja, noch ein Kilo Zwiebeln.	KBBd (Kunden-Bedarf)
Verkäufer: Sechs Mark fünfundzwanzig, bitte.	VAA (Verkäufer-Abschluss)
Kunde 2: Auf Wiedersehen!	KAV (Kunden-Verabschiedung)

11.0.14 Transkript 6 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Käseverkaufsstand, Aachen, Marktplatz

Transkriptausschnitt	Terminalzeichen
Kunde 1: Guten Morgen!	KBG (Kunden-Gruß)
Verkäufer: Guten Morgen!	VBG (Verkäufer-Gruß)
Kunde 1: Ich hätte gerne fünfhundert Gramm holländi	KBBd (Kunden-Bedarf)
Verkäufer: Am Stück?	VBBd (Verkäufer-Klärung)
Kunde 1: Ja, am Stück, bitte.	KAA (Kunden-Abschluss)

11.0.15 Transkript 7 - Terminalzeichen

Datum: 28. Juni 1994, **Ort:** Bonbonstand, Aachen, Marktplatz, 11:30 Uhr

Transkriptausschnitt	Terminalzeichen
Kunde: Von den gemischten hätte ich gerne hundert G	KBBd (Kunden-Bedarf)
Verkäufer: Für zu Hause oder zum Mitnehmen?	VBBd (Verkäufer-Klärung)
Kunde: Zum Mitnehmen, bitte.	KBA (Kunden-Argument)
Verkäufer: Fünfzig Pfennig, bitte.	VAA (Verkäufer-Abschluss)
Kunde: Danke!	KAA (Kunden-Abschluss)

11.0.16 Transkript 8 - Terminalzeichen

Datum: 9. Juli 1994, **Ort:** Bäckerei, Aachen, 12:00 Uhr

Transkriptausschnitt	Terminalzeichen
Kunde: Guten Tag!	KBG (Kunden-Gruß)
Verkäuferin: Einmal unser bester Kaffee, frisch gem	VBBd (Verkäufer-Klärung)
Kunde: Ja, noch zwei Stück Obstsalat und ein Schäl	KBBd (Kunden-Bedarf)
Verkäuferin: In Ordnung!	VBA (Verkäufer-Argument)

Transkriptausschnitt	Terminalzeichen
Verkäuferin: Das macht vierzehn Mark und neunzehn P	VAA (Verkäufer-Abschluss)
Kunde: Ich zahle in Kleingeld.	KAA (Kunden-Abschluss)
Verkäuferin: Vielen Dank, schönen Sonntag noch!	VAV (Verkäufer- Verabschiedung)
Kunde: Danke, Ihnen auch!	KAV (Kunden- Verabschiedung)

```
[1]: import numpy as np
from scipy.stats import pearsonr

# Neue empirische Terminalzeichenketten
empirical_chains = [
    ['KBG', 'VBG', 'KBBd', 'VBBd', 'KBA', 'VBA', 'KBBd', 'VBBd', 'KBA', 'VAA', 'KAA', 'VAV', 'KAV'],
    ['VBG', 'KBBd', 'VBBd', 'VAA', 'KAA', 'VBG', 'KBBd', 'VAA', 'KAA'],
    ['KBBd', 'VBBd', 'VAA', 'KAA'],
    ['KBBd', 'VBBd', 'KBA', 'VBA', 'KBBd', 'VBA', 'KAE', 'VAE', 'KAA', 'VAV', 'KAV'],
    ['KAV', 'KBBd', 'VBBd', 'KBBd', 'VAA', 'KAV'],
    ['KBG', 'VBG', 'KBBd', 'VBBd', 'KAA'],
    ['KBBd', 'VBBd', 'KBA', 'VAA', 'KAA'],
    ['KBG', 'VBBd', 'KBBd', 'VBA', 'VAA', 'KAA', 'VAV', 'KAV']
]

# Übergangszählung initialisieren
transitions = {}
for chain in empirical_chains:
    for i in range(len(chain) - 1):
        start, end = chain[i], chain[i + 1]
        if start not in transitions:
            transitions[start] = {}
        if end not in transitions[start]:
            transitions[start][end] = 0
        transitions[start][end] += 1

# Normalisierung: Übergangswahrscheinlichkeiten berechnen
probabilities = {}
for start in transitions:
    total = sum(transitions[start].values())
    probabilities[start] = {end: count / total for end, count in transitions[start].items()}

# Terminalzeichen und Startzeichen definieren
```

```

terminal_symbols = list(set([item for sublist in empirical_chains for item in
    ↳sublist]))
start_symbol = empirical_chains[0][0]

# Funktion zur Generierung von Ketten basierend auf der Grammatik
def generate_chain(max_length=10):
    chain = [start_symbol]
    while len(chain) < max_length:
        current = chain[-1]
        if current not in probabilities:
            break
        next_symbol = np.random.choice(list(probabilities[current].keys()),
    ↳p=list(probabilities[current].values()))
        chain.append(next_symbol)
        if next_symbol not in probabilities:
            break
    return chain

# Funktion zur Berechnung relativer Häufigkeiten
def compute_frequencies(chains, terminals):
    frequency_array = np.zeros(len(terminals))
    terminal_index = {term: i for i, term in enumerate(terminals)}

    for chain in chains:
        for symbol in chain:
            if symbol in terminal_index:
                frequency_array[terminal_index[symbol]] += 1

    total = frequency_array.sum()
    if total > 0:
        frequency_array /= total # Normierung der Häufigkeiten

    return frequency_array

# Iterative Optimierung
max_iterations = 1000
tolerance = 0.01 # Toleranz für Standardmessfehler
best_correlation = 0
best_significance = 1

# Relativ Häufigkeiten der empirischen Ketten berechnen
empirical_frequencies = compute_frequencies(empirical_chains, terminal_symbols)

for iteration in range(max_iterations):
    # Generiere 8 künstliche Ketten
    generated_chains = [generate_chain() for _ in range(8)]

```

```

# Relativ Häufigkeiten der generierten Ketten berechnen
generated_frequencies = compute_frequencies(generated_chains,
↳terminal_symbols)

# Berechne die Korrelation
correlation, p_value = pearsonr(empirical_frequencies,
↳generated_frequencies)

print(f"Iteration {iteration + 1}, Korrelation: {correlation:.3f},
↳Signifikanz: {p_value:.3f}")

# Überprüfen, ob Korrelation und Signifikanz akzeptabel sind
if correlation >= 0.9 and p_value < 0.05:
    best_correlation = correlation
    best_significance = p_value
    break

# Anpassung der Wahrscheinlichkeiten basierend auf Standardmessfehler
for start in probabilities:
    for end in probabilities[start]:
        # Fehlerabschätzung basierend auf Differenz der Häufigkeiten
        empirical_prob = empirical_frequencies[terminal_symbols.index(end)]
        generated_prob = generated_frequencies[terminal_symbols.index(end)]
        error = empirical_prob - generated_prob

        # Anpassung der Wahrscheinlichkeit
        probabilities[start][end] += error * tolerance
        probabilities[start][end] = max(0, min(1,
↳probabilities[start][end])) # Begrenzen auf [0,1]

# Normalisierung
for start in probabilities:
    total = sum(probabilities[start].values())
    if total > 0:
        probabilities[start] = {end: prob / total for end, prob in
↳probabilities[start].items()}

# Ergebnis ausgeben
print("\nOptimierte probabilistische Grammatik:")
for start, transitions in probabilities.items():
    print(f"{start} → {transitions}")

print(f"\nBeste Korrelation: {best_correlation:.3f}, Signifikanz:
↳{best_significance:.3f}")

```

Iteration 1, Korrelation: 0.861, Signifikanz: 0.000

Iteration 2, Korrelation: 0.835, Signifikanz: 0.001

Iteration 3, Korrelation: 0.903, Signifikanz: 0.000

Optimierte probabilistische Grammatik:

KBG → {'VBG': 0.6665949194957433, 'VBBd': 0.33340508050425677}
VBG → {'KBBd': 1.0}
KBBd → {'VBBd': 0.6661627789328691, 'VAA': 0.16672317719273486, 'VBA': 0.16711404387439593}
VBBd → {'KBA': 0.4444746815960467, 'VAA': 0.22218925521330254, 'KBBd': 0.2214541950661155, 'KAA': 0.11188186812453531}
KBA → {'VBA': 0.5001954483997749, 'VAA': 0.4998045516002252}
VBA → {'KBBd': 0.4995306939857028, 'KAE': 0.25025106688027127, 'VAA': 0.2502182391340259}
VAA → {'KAA': 0.8573622459421577, 'KAV': 0.1426377540578423}
KAA → {'VAV': 0.750137275176534, 'VBG': 0.24986272482346608}
VAV → {'KAV': 1.0}
KAE → {'VAE': 1.0}
VAE → {'KAA': 1.0}
KAV → {'KBBd': 1.0}

Beste Korrelation: 0.903, Signifikanz: 0.000

[]: