Superflo für die Liga freischwärmender Sprachakrobaten (Aufbauorganisation) • MITTEILUNG 01/2026

LEOFLATION - LEOFIZIERENDE WORTBILDUNG IN DEUTSCH UND ENGLISCH

Wahrscheinlichkeit von und Zeichenwachstum bei formal kleinst gefasster Leoflation.

Eine Festschrift zu Leo im Deutschen und Leo im Englischen. Mit Leo und einem Bigramm-Markowmodell zur Ermittlung wie oft.

Theleodoxes Seminar (flo) mit freundlicher Genehmigung des Zirkels gut abgehangener Glossolalisten sowie der Klosterbrüder der Sophisten und Suppenpotentaten. | 14. November 2026

Abstract. Wie viel `leo` verträgt ein durchschnittlicher Satz, wenn pro Wort höchstens ein Zeichen eingefügt, ersetzt oder gelöscht werden darf? Aus den Zeichenpaarhäufigkeiten von Tesseract-Trainingsdaten konstruieren wir ein Markow Modell von Bigrammen mit Leo-Affix-Charakter. Unter einer transparenten Einfügepriorität zeigen Wir das mit keinesfalls weniger als **0,964 Leos pro deutschem** und **1,226 pro englischem** Durchschnittssatz zu rechnen hat wer damit beginnt. Ebenso berechnen wir den expandierenden(leoflativen) Charakter von Sätzen die Leo unterliegen. Unser Modell demonstriert monoton steigende Satzlängen um **0,91 %** bzw. **1,06 %**. Die Ergebnisse sind Modellwerte, keine direkte Messung an tatsächlich leoflationierten Texten, das war uns zu gefährlich. Aber Sprache mittelt sich eh auf Dauer daher passt das auch so.

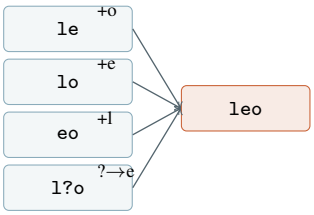


Abbildung 1: Die vier lokalen Leomotive. Die Fälle `le`, `lo`, `eo` liefern Einfügungen; `l?o` erlaubt eine Ersetzung.

Tabelle 1: Häufigkeit der drei Einfügungsbigramme in Prozent aller berücksichtigten Zeichenpaare.

Motiv	Deutsch	Englisch
`le`	0,9687	0,9091
`lo`	0,2382	0,4988
`eo`	0,0606	0,1104
Summe	1,2675	1,5183

1 Fragestellung und Formalismus

Eine besonders einfältige Form der Wortspielerei ergibt sich aus der einfachen Substition, Ergänzung oder Streichung eines Buchstaben zwecks austausch eines Prä- In- oder Suffixes und motiviert die Forschungsfrage: Wie häufig lässt sich in natürlichsprachigen Wörtern die Zeichenfolge `leo` durch *genau eine* Editoperation erzeugen? Anders als freie Wortspiele oder gar $Kunst$ halten wir uns für die L erfinden. Wir betrachten Wörter als Folgen aus dem Alphabet $\Sigma = \{a, \dots, z\}$; Groß- und Kleinschreibung werden vereinheitlicht. Leerzeichen und Satzzeichen sind *nicht* Teil des statistischen Wortmodells.

Sei $w \in \Sigma^n$, $L =$ `leo` und d_{Lev} die Levenshtein-Distanz. Ein Leotreffer liegt vor, wenn ein Teilwort u von w die Bedingung $d_{\mathrm{Lev}}(u, L) \leq 1$ erfüllt. Die Editionen werden pro Wort in der Reihenfolge *bereits vorhanden* $\succ$ *Einfügung* $\succ$ *Ersetzung* $\succ$ *Löschung* priorisiert. Höchstens eine Mutation wird ausgeführt, auch wenn mehrere Stellen möglich wären.

1.1 Vier fundamentale Motive

Die vollständige Charakterisierung ist erfreulich kurz:

$$\boxed{\mathcal{M} = \{\texttt{le}, \texttt{lo}, \texttt{eo}, \texttt{l?o}\}.} \tag{1}$$

Hier ist `?` ein einzelner, beliebiger Buchstabe; das vorhandene `leo` ist eingeschlossen. Jedes `le`, `lo` oder `eo` lässt sich durch Einfügung der fehlenden Letter vervollständigen. Das Muster `l?o` lässt sich durch Ersetzung des Mittelzeichens vervollständigen. Weitere einstellige Ersetzungen bei `?eo` und `le?` sind durch die genannten Bigramme schon erfasst.

1.2 Der Leowachstumssatz

Kann durch Löschung eines Zeichens aus einem Viererwort `leo` werden, so enthält die ursprüngliche Viererfolge mindestens eines der Paare `le`, `lo` oder `eo`. Eine Einfügung ist deshalb ebenfalls möglich und erhält Vorrang. Folglich ist für die gewählte Politik

$$\boxed{\Delta|w| \in \{0, 1\}.} \tag{2}$$

Der Algorithmus ist somit ein *monotoner Zeicheninflationsoperator*. Quod erat leonstrandum.

2 Daten und statistische Methode

2.1 Beobachtete Bigramme

Grundlage sind die Dateien `deu` und `eng` aus dem öffentlichen Tesseract-Projekt [1, 2]. Alle Zweierfolgen aus `a-z` werden nach Kleinschreibung aufsummiert; Paare mit anderen Zeichen bleiben ausgeschlossen. Daraus resultieren etwa $413{,}6 \cdot 10^6$ deutsche und $617{,}0 \cdot 10^6$ englische Paare.

Ein unabhängiges Buchstabenmodell würde z. B. $P(\texttt{eo}) = P(e)P(o)$ unterstellen. Im deutschen Zeichensatz liefert es ungefähr $0{,}634\,\%$, beobachtet werden aber nur $0{,}061\,\%$. Nicht die Häufigkeit einzelner Buchstaben ist entscheidend, sondern ihre Nachbarschaft. Gerade diese Unabhängigkeitsannahme würde die Leodichte stark verzerren.

2.2 Markowkette und Musterautomat

Mit $N(ab)$ als Häufigkeit des Paares ab setzen wir

$$P(b \mid a) = \frac{N(ab)}{\sum_{x \in \Sigma} N(ax)}. \tag{3}$$

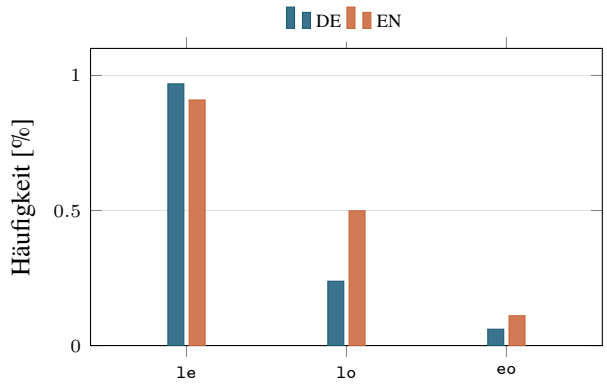


Abbildung 2: Empirische Einfügungsbigramme. `lo` ist im Englischen deutlich häufiger.

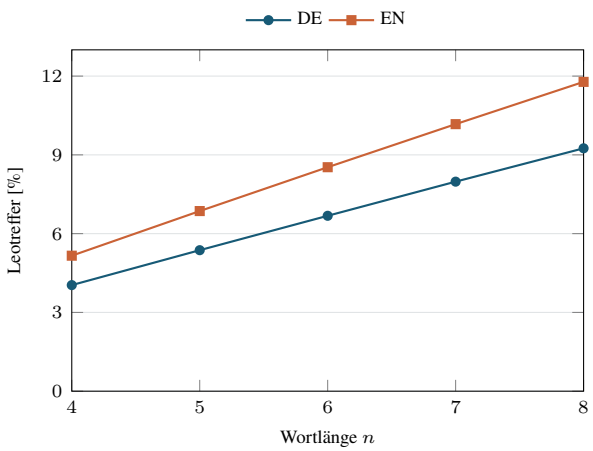


Abbildung 3: Modellierte Wort-Leofizierbarkeit nach Wortlänge. Schon ein einziger erlaubter Edit erzeugt messbare Trefferquoten.

Tabelle 2: Disjunkte Ausgaben der Modellentscheidung für $n = 6$ (DE) und $n = 5$ (EN). Angaben in Prozent.

Klasse	DE	EN
Bereits `leo`	0,0152	0,0298
Einfügen p_+	6,2934	6,0973
Ersetzen $p_\sim$	0,3748	0,7358
Keine Mutation	93,3167	93,1371
Leo insgesamt	**6,6833**	**6,8629**

Tabelle 3: Modellhafte Satz-Leofizierung; Rundungen nach der Berechnung.

Kennzahl	DE	EN
Wörter / Satz	14,42	17,86
Leos / Satz	0,964	1,226
$P(\geq 1\ \text{Leo})$	63,1%	71,9%
Mehrzeichen / Satz	0,908	1,089
Zeicheninflation	0,91%	1,06%

Der Startbuchstabe wird aus der über alle Folgebuchstaben summierten Zeilenhäufigkeit gezogen. Ein Aho–Corasick-artiger Musterautomat protokolliert, ob `leo`, eines der drei Zweiermotive oder `l?o` bisher auftrat. Dynamische Programmierung über Wortlängen n liefert disjunkte Wahrscheinlichkeiten p_0 (schon `leo`), p_+ (Einfügung) und $p_\sim$ (nur Ersetzung):

$$p_{\text{Leo}} = p_0 + p_+ + p_\sim. \tag{4}$$

Ein möglicher Löschfall ist wegen Gl. (2) niemals bevorzugt. Das Modell ignoriert tatsächliche Wortanfänge, Morphemgrenzen und Korpus-Wortlängenverteilungen; hierzu Abschnitt 6.

3 Ergebnisse auf Wortebene

Insbesondere `l?o` ist *nicht* übergangen: Sein zusätzlicher Anteil nach Ausschöpfung sämtlicher Einfügungsgelegenheiten beträgt $0,3748\,\%$ (DE) beziehungsweise $0,7358\,\%$ (EN). Man darf $P(\texttt{le})$, $P(\texttt{lo})$, $P(\texttt{eo})$ und $P(\texttt{l?o})$ nicht naiv addieren, da die Treffer überlappen und ein Wort nur einmal verändert wird.

4 Vom Wort zum Durchschnittssatz

Für geschriebene deutsche Korpustexte nennt der Duden eine mittlere Wortlänge von $\bar{w}_{\text{DE}} = 5{,}97$ Buchstaben [3]; für englisches Buchmaterial berichtet Norvig $\bar{w}_{\text{EN}} = 4{,}79$ [4]. Die Vergleichsstudie von Zhou, Jiang und Liu ermittelt in Nachrichtenkorpora $S_{\text{DE}} = 14{,}4224$ und $S_{\text{EN}} = 17{,}8598$ Wörter je Satz [5].

Für die Wortwahrscheinlichkeiten verwenden wir vereinfachend $n_{\text{DE}} = 6$ und $n_{\text{EN}} = 5$. Dann sind

$$\mathbb{E}[N_{\text{Leo}}] = S\,p_{\text{Leo}}, \tag{5}$$

$$P(N_{\text{Leo}} \geq 1) \approx 1 - (1 - p_{\text{Leo}})^{S}. \tag{6}$$

Die zweite Gleichung setzt unabhängig verteilte Worttreffer und eine deterministische mittlere Satzlänge voraus; bei echten Sätzen trifft beides nicht exakt zu.

Die mittlere zusätzliche Zeichenzahl lautet $\mathbb{E}[\Delta C] = S\,p_+$. Die Ursprungslänge wird durch $C_0 \approx S(\bar{w}+1) - 1$ angenähert, inklusive je eines Leerzeichens zwischen den Wörtern. Daraus folgt die dimensionslose *Leoinflationszahl*

$$\boxed{\mathcal{I}_{\text{Leo}} = \frac{S\,p_+}{S(\bar{w}+1) - 1}.} \tag{7}$$

Für Deutsch sind dies $0{,}9077$ zusätzliche Zeichen auf geschätzte $99{,}52$ Ausgangszeichen, also $0{,}91\,\%$. Englisch liefert $1{,}0890$ auf $102{,}41$ Zeichen, also $1{,}06\,\%$. Auf 1 000 gleich lange Modellwörter kommen rund 66,8 (DE) bzw. 68,6 (EN) Leo-Treffer.

5 Diskussion: Reicht ein Prozent?

Ja: Die Leoinflation der Zeichenlänge ist bescheiden, die der Semantik dagegen erheblich. Ein Modellwort mit sechs deutschen Buchstaben trägt nach einem Durchlauf mit Wahrscheinlichkeit von etwa $6{,}7\,\%$ ein `leo`. Damit ist im Durchschnitt fast jeder deutsche Satz betroffen, im Englischen sogar mehr als ein Leo pro Satz zu erwarten.

Die bereits ohne Eingriff enthaltene Zeichenfolge `leo` ist ausgesprochen selten: $0{,}0152\,\%$ (DE) bzw. $0{,}0298\,\%$ (EN) bei den gewählten Wortlängen. Das Verfahren erhöht die Vorkommensrate modellhaft ungefähr um Faktor 440 bzw. 230.

Die Zahlen sind bewusst nicht als Effektgröße für beliebige echte Korpora zu interpretieren.

5.1 Der vernachlässigte Freiraum `l?o`

Im expliziten Wortautomaten umfasst `?` alle 26 ASCII-Buchstaben. Bei `leo` liegt schon ein Treffer vor; 25 andere Mittelzeichen sind mögliche Ersetzungen. Die Wörter `also` und `Alto` illustrieren die Regel. Umlaute, Eszett und Ziffern sind in diesem *statistischen* Modell nicht erfasst; eine Unicode-fähige JavaScript-Anwendung kann sie durchaus als editierbare Zeichen behandeln.

5.2 Wortgrenzen als Forschungsprogramm

Die Varianten `l o`→`leo` (Leerzeichen ersetzen) und `le o`→`leo` (Leerzeichen entfernen) sind im Ein-Wort-Modell nicht enthalten. Hierfür braucht man Häufigkeiten von Wortend- und Wortanfangsbuchstaben sowie eine explizite Politik zur Zusammenfassung zweier Wörter. Ohne diese zusätzlichen Daten wäre eine Zahl für wortgrenzenüberschreitende Leoereignisse erfunden.

6 Grenzen der Aussagekraft

- **Bigramm-Markowmodell:** Es modelliert nur den jeweils vorherigen Buchstaben, nicht Silben, Wortgrenzen, Flexion oder die Verteilung echter Wörter.
- **Heterogene Quellen:** Zeichenpaare aus OCR-Trainingsmaterial werden mit Wortlängen aus Literatur/Korpora und Satzlängen aus Nachrichtentexten kombiniert.
- **Punktlängen:** $n = 6$ bzw. $n = 5$ ersetzen die volle Verteilung der Wortlängen. Das kann die Satzprognosen verschieben.
- **Unabhängigkeit:** Der Satzwert $P(\geq 1)$ berücksichtigt weder syntaktische Abhängigkeiten noch die natürliche Verteilung der Satzlängen.
- **Keine Korpusvalidierung:** Die Modellvorhersagen wurden hier nicht an vollständig durchlaufenen Texten gegengeprüft. Nachkommastellen sind rechnerisch, nicht inferenzstatistisch präzise.

Eine saubere Folgestudie würde dieselben dokumentierten Korpora mit der realen Transformationsfunktion verarbeiten und die gemessenen Trefferquoten, zusätzliche Zeichen sowie Konfidenzintervalle gegen das Markowmodell ausweisen.

7 Conclusio: das erste Leogesetz

Das Leomotiv besitzt eine knappe formale Beschreibung, seine Einfügungsgelegenheiten sind in beiden Sprachen statistisch nicht selten, und bei Einfügepriorität ist die Transformation niemals verkürzend. Ein Durchschnittssatz erhält ungefähr ein Leo bei nur einem Prozent Zeichenwachstum. Oder wissenschaftlicher formuliert:

$$\boxed{\text{Mehr Text} \;\Rightarrow\; \text{mehr Gelegenheit für Leo.}}$$

Quod erat leonstrandum.

Abkürzungsverzeichnis

ASCII	7-Bit-Zeichencodierung
DE / EN	Deutsch / Englisch
DP	dynamische Programmierung
LFI	Leofizierungsinflation
MK	Markowkette erster Ordnung
OCR	optische Zeichenerkennung
P, $\mathbb{E}$	Wahrscheinlichkeit, Erwartungswert
S, $\bar{w}$	Wörter pro Satz, mittlere Wortlänge

Abbildungsverzeichnis

1 Die vier lokalen Leomotive. Die Fälle `le`, `lo`, `eo` liefern Einfügungen; `l?o` erlaubt eine Ersetzung. 1
2 Empirische Einfügungsbigramme. `lo` ist im Englischen deutlich häufiger. 2
3 Modellierte Wort-Leofizierbarkeit nach Wortlänge. Schon ein einziger erlaubter Edit erzeugt messbare Trefferquoten. . 2

Literatur

[1] Tesseract OCR. *langdata*: Bigramm-Dateien für `deu` und `eng`. Quellrepository, Zugriff 14.11.2026. GitHub-Repository

[2] Tesseract OCR. *Training Tesseract: Langdata Files*. Beschreibung der Trainingsdaten und der `bigram_freqs`. Dokumentation

[3] Dudenredaktion. *Durchschnittliche Länge eines deutschen Wortes*. Online-Sprachwissen, Zugriff 14.11.2026. Duden online

[4] Norvig, P. (2012). *English Letter Frequency Counts: Mayzner Revisited*. Onlineanalyse

[5] Zhou, Y.; Jiang, J.; Liu, H. (2025). Language Universals in Sentence Length: Comparing Sentence Length Distributions of 10 Languages. *Cognitive Science* 49(9), e70115. doi:10.1111/cogs.70115