

# Automatische Sprachübersetzung von $\text{\LaTeX}$ -Dokumenten

Name: Hendrik Theede

Matrikelnummer: 221201256

Abgabedatum: 02.12.2025

Betreuer und Gutachter: Prof. Dr. rer. nat. habil. Clemens H. Cap  
Universität Rostock  
Fakultät für Elektrotechnik und Informatik

# Abstrakt


placeholder

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Hintergrund . . . . .	1
1.2	Thematische Einordnung . . . . .	1
<b>2</b>	<b>Problemfälle</b>	<b>2</b>
2.1	Hinweise für Leser . . . . .	2
2.2	T <sub>E</sub> X's Syntax . . . . .	4
2.2.1	Befehle . . . . .	4
2.2.2	Umgebungen . . . . .	5
2.2.3	Dateien . . . . .	5
2.2.4	Definitionen (Makros) . . . . .	5
2.3	L <sup>A</sup> T <sub>E</sub> X Dokumente . . . . .	6
2.3.1	Erneuerungen . . . . .	6
<b>3</b>	<b>Stand der Technik</b>	<b>8</b>
3.1	Anforderungen . . . . .	8
3.2	Denkbare Ansätze . . . . .	8
3.3	Existierende Ansätze . . . . .	8
3.3.1	Testverfahren . . . . .	8
3.3.2	Durchführung . . . . .	8
3.3.3	Auswertung . . . . .	8
3.4	Grenzen der Lösungen . . . . .	8
3.5	Takeaways . . . . .	8
<b>4</b>	<b>Eigenständigkeitserklärung</b>	<b>9</b>
	<b>Literatur</b>	<b>10</b>
<b>A</b>	<b>Anhänge</b>	<b>11</b>
A.1	Fontskalierung auf Webseiten . . . . .	11



# 1 Einleitung

## 1.1 Hintergrund

 Die schnellstmögliche und einfache Erstellung von Dokumenten beliebiger Natur (formlos) wird heutzutage oftmals über Produkte bekannter Anbieter abgewickelt (bspw. Microsoft's Word, PowerPoint, ... und die vergleichbaren „LibreOffice“-Software, sowie die korrespondierenden Apple-Produkte). Unterliegen Dokumente allerdings strengeren stilistischen Vorgaben (bspw. bei wissenschaftlichen Veröffentlichungen) entsteht der Vorteil, dass sich diese Vorgaben wie ein Regelsatz behandeln lässt, aus welchem sich bestimmte, vorprogrammierte Dokumentenstrukturen definieren lassen. Hierzu existiert bereits ein geläufiges System welches den Namen  $\text{\LaTeX}$  trägt (mit dem *La* nach einem der ursprünglichen Entwickler Lamport (1994)), welches selbst auf dem von Knuth (1986) entwickelten Zeichensetzungs-System und der verbundenen Programmiersprache  $\text{\TeX}$  basiert.

Die  $\text{\TeX}$ -Syntax selbst basiert auf englischen Begriffen, allerdings ist nicht davon auszugehen, dass nur englischsprachige Menschen  $\text{\LaTeX}$  und  $\text{\TeX}$  nutzen werden. Quelltexte und Dokumentenbeschreibungen werden also nicht immer in einer rein englischsprachigen Form vorliegen (z.B. `\chapter{Erstes Kapitel: Einleitung}` oder der Quelltext dieses Werkes). Ein Zurückführen solcher Dokumente in die englische Sprache ist einfach, insofern ein Verständnis der deutschen Sprache besteht (`\chapter{First Chapter: Introduction}`). Die Rückrichtung zeigt sich allerdings genau dann problematisch, sollte ohne Vorkenntnisse von  $\text{\TeX}$  (bzw. dessen syntaktische Elemente) bestehen. In genanntem Beispiel würde dann das Wort `chapter` aufgegriffen werden und die Zeichenkette `\Kapitel{Erstes Kapitel: Einleitung}` entstehen (ohne weitere, mögliche Anpassungen entsteht hier keine Kapitelüberschrift mehr. Das erstere „Kapitel“ würde ignoriert werden und die innerhalb der Klammern stehende Zeichenkette als einfacher Fließtext gedruckt werden).

## 1.2 Thematische Einordnung

 Bereits Shannon (1948) beschäftigte sich mit theoretischen Grundlagen der Darstellung und Übertragung von Informationen, insbesondere der menschlicher Sprache und Kommunikation. Heutige maschinelle Systeme zu diesem Zweck (bspw. ChatGPT, DeepL, Gemini und co.) wirken zunächst wie „magische“ Blackboxen, arbeiten jedoch auf Grundlage von statistischen Modellen. Spricht man hier von Magie, dann ist jeder Mitarbeiter eines Wetterdienstes oder der Klimaforschung „bezaubernd“. Das zugrundeliegende Konzept kann jedoch sehr schnell auf den Punkt gebracht werden: Eine KI erhält einen Input, für welchen ein bestimmter Output erwartet wird (Beispiel: „Einfügen“ als nächstes Wort eines zu übersetzenden Satzes und „insertion“ im Kontext innerhalb des Satzes als Erwartung (substantiviertes Verb)), und produziert einen Output, welcher mit dem Erwarteten abgeglichen wird. Sollte das Resultat von der Erwartung abweichen (z.B. „insert“ entstehen), so kann dieser Fehler erkannt werden und im Modell dazu beitragen, dass (gegeben einer bestimmten, sequentiellen Folge von Wörtern (Satzstruktur)) dieser „Fehler“ von nun an seltener passiert. Allerdings wird klar, dass eine KI unabdingbar Fehler machen muss, denn nur so kann diese lernen. Dies gilt es stets zu berücksichtigen, wodurch theoretische gesehen immer mehrere Permutationen betrachtet werden müssen, sollten Probleme entstehen, in welchen ein Input mehrere Ausgaben erzeugen könnte. Angemerkt sei zu dem Vorherigen, dass bekannte und bereits rein in den menschlichen Sprachen auftretende Problem (bzw. mögliche Missverständnisse bereits im Verstehen  ner Sprache, ausgelöst durch Mehrdeutigkeiten von Wörtern) in dieser Arbeit nicht näher verfolgt werden.

## 2 Problemfälle

### 2.1 Hinweise für Leser

#### Herangehensweise

Die nachfolgende Auflistung an verschiedenen Fällen, welche Probleme gegenüber der  $\text{T}_{\text{E}}\text{X}$ -Syntax, bzw. innerhalb von  $\text{\LaTeX}$ -Dokumenten hervorrufen könnten, benötigt per se keine Reihenfolge, da sie möglichst alle behoben sein sollen. Ein zufälliges sequenzielles Nennen dieser würde jedoch die Lesbarkeit und Nachvollziehbarkeit dieser Arbeit mindern und eventuell auftretende, aber übersehene Fälle für Andere nicht schnellig ersichtlich machen. Deshalb wird eine Reihenfolge gewählt, welche nicht auf  $\text{\LaTeX}$ -Ebene beginnt, sondern sich so weit wie möglich dem Ursprung dieses Systemes nähert und von diesem ausgehend zunächst alleine in diesem erdenkbare Probleme beschreibt und darauf aufbauend neue Systeme (bzw. benötigte Systeme für bestimmte Zwecke) benennt und die damit verbundenen Probleme schildert. Für jegliche Beispiele sei bemerkt, dass kein „Übersetzer“ im herkömmlichen Sinne (menschlich, maschinell, . . . ) herangezogen wurde, sondern ein Konzeptioneller. Dieser erkennt verschiedene Zeichen und weiß, dass einige Teil der  $\text{T}_{\text{E}}\text{X}$ -Syntax sein *könnten*, aber betrachtet immer nur alleine-stehende Wörter, die er kennt und versucht möglichst alle Gefundenen zu übersetzen, als auch eine alternative Zeichenkette zu liefern, in welcher dieses Wort nicht übersetzt wurde, falls es fälschlich aufgegriffen war. Abhängig von den bestimmten Zeichenketten könnten hierbei eine Vielzahl an Permutationen entstehen, welche möglichst versucht wurden in folgender Struktur einzuordnen.

#### Struktur eines Beispiels

Die Darstellung einzelner Beispiele erfolgt tabellarisch und demonstriert zunächst gewünschte (bzw. zulässige) Verhalten und danach Unerwünschte (bzw. Fehlerhafte). Untige 1 dient hierbei als einfaches Beispiel und zeigt, wie das Übersetzen von Zeichenketten, welche z.B. Tags enthalten, theoretisch unterschiedliche Permutationen erzeugen können. Inwiefern sich die einzelnen zuvor unterschiedenen, möglichen Verhalten unterschiedlich äußern, soll für jedes Beispiel konkretisiert werden. Dieses einleitende Beispiel nutzt daher (zunächst) den rein imaginären Befehl `ink` mit einer auswählbaren Farbe, die auf einen String angewendet wird. Wünschenswert ist, dass nur der in geschwungenen Klammern stehende String übersetzt wird, aber ein Auslassen dieses wäre in erster Betrachtung nicht  $\text{T}_{\text{E}}\text{X}$ -Syntax brechend und daher zulässig. Unerwünscht wäre das Übersetzen der zusätzlichen Option in eckigen Klammern, wobei man davon ausgehen würde, dass für diese Option auf einen *default*-Wert zurückgegriffen wird. Dies ist zwar ein syntaktischer Fehler, würde jedoch den Befehl selbst ausführbar lassen. Als „Fälschlich“ werden demnach Fehler betrachtet, welche die  $\text{T}_{\text{E}}\text{X}$ -Syntax insofern brechen, dass der Kompilierprozess unterbrochen/abgebrochen wird.

English	Mögliche Übersetzung
Erwünscht	
<code>1\ink[red]{word}</code>	
Zulässig	
<code>1\ink[red]{word}</code>	
Unerwünscht	
<code>1\ink[red]{word}</code>	
Falsch	
<code>1\ink[red]{word}</code>	<code>1\ink[red]{word}</code>

Tabelle 1: Abstrakte Struktur der folgenden Beispiele

## 2.2 T<sub>E</sub>X's Syntax

### 2.2.1 Befehle

**Kommandos** 2 zeigt, dass selbst das Übersetzen einzelner Wörter zu Problemen führen kann, da die  $\TeX$  interne Variable des Autoren, welche in der Präambel gesetzt wird, durch die gezeigte Übersetzung nun nicht mehr passend referenziert ist.

English	Mögliche Übersetzung
Erwünscht <div></div> <div></div> <div></div> <div></div>	<div></div>
<div></div> <div>1\@author</div>	
Falsch <div></div> <div></div> <div></div> <div></div>	<div></div>
<div></div> <div>1\@author</div>	

Tabelle 2: Das Fehlerhafte Beispiel meint nun nicht mehr die author-Variable, sondern eine (wohlmöglich) nicht existierende Autor-Variable

## Auf Zeichenketten angewendete Kommandos

Kommandos mit Optionen

Befehle mit spezifischen Eingaben

2.2.2 Umgebungen

2.2.3 Dateien

includes und inputs

Referenzen

2.2.4 Definitionen (Makros)



## 2.3 L<sup>A</sup>T<sub>E</sub>X Dokumente

### 2.3.1 Erneuerungen

English	Mögliche Übersetzung
---------	----------------------

Erwünscht	
<div> <div></div> <div></div> <div></div> <div>1\section{example}</div> <div></div> <div></div> </div>	<div></div>

Zulässig	
<div> <div></div> <div></div> <div></div> <div>1\section{example}</div> <div></div> <div></div> </div>	<div></div>

Unerwünscht	
<div> <div></div> <div></div> <div></div> <div>1\section{example}</div> <div></div> <div></div> </div>	<div></div>

Falsch	
--------	--

## **3 Stand der Technik**

### **3.1 Anforderungen**

Abgelitten aus der Problemliste werden hier die Probleme umformuliert als Anforderungen dargestellt und in absteigender Reihenfolge nach Relevanz in Bezug auf die gegebene Aufgabenstellung aufgeführt.

Die Technologien dienen den Anforderungen, sollten sie:

1. kompilierbare Dokumente erzeugen
2. alle Abschnitte in Dokumenten übersetzen
3. kontextuell terminologisch richtige Übersetzungen wählen (die richtigen Lexeme/Wörter treffen)
4. den Kontext selbstständig aus den wörtlichen und erreichbaren (lokalen) Informationen (Dateien) ablesen können
5. den Kontext aus den mathematischen, graphischen, tabellarischen, ... Inhalten einer Datei ablesen können
6. den Kontext aus externen Verweisen (Links) erfassen können (Lokal, als auch Web)
7. ...

### **3.2 Denkbare Ansätze**

Alle Lösungswege und Workflows, die ich mir vorstellen kann und denken konnte. Definiert evtl. Rollen,

### **3.3 Existierende Ansätze**

Alle Technologien, die diese Rolle (n) in den entsprechenden Ansätzen füllen könnten.

#### **3.3.1 Testverfahren**

logischerweise: In den denkbaren Ansätzen schon gegenargumentieren, was unsinnig ist und warum. Reduziert die Menge an zu testenden Lösungen.

#### **3.3.2 Durchführung**

#### **3.3.3 Auswertung**

### **3.4 Grenzen der Lösungen**

### **3.5 Takeaways**

## 4 Eigenständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt und ohne fremde Hilfe verfasst habe. Dazu habe ich keine außer den von mir angegebenen Hilfsmitteln und Quellen verwendet und die den benutzten Werken inhaltlich und wörtlich entnommenen Stellen habe ich als solche kenntlich gemacht. Ich versichere, dass die eingereichte elektronische Fassung mit den gedruckten Exemplaren übereinstimmt.

Rostock, den 02.12.2025

---

Hendrik Theede

## Literatur

Knuth, D. E. (1986), *The TeXbook*, ISBN: 9780201134476, Addison-Wesley Professional.

Lamport, L. (1994), *LaTeX: A Document Preparation System, 2nd Edition*, ISBN: 9780201529838, Addison-Wesley Professional. available at: <https://www.latex-project.org/help/books/tlc3-digital-chapter-samples.pdf> (last Access: 04.10.2025).

Shannon, C. E. (1948), 'The mathematical theory of communication', *The Bell System Technical Journal*, ISSN: 0343-6993 (vol. 27). Harvard Reprint available at <https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf> (last Access: 16.10.2025).

## A Anhänge

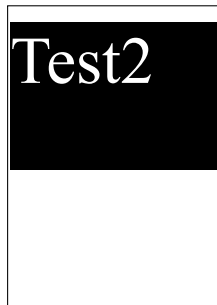
### A.1 Fontskalierung auf Webseiten

Beispielsweise produziert die folgende HTML-Notation bei einer Skalierung im Browser von 120 Prozent (Abbildung 2a) und 50 Prozent (Abbildung 2b) jeweilig zwei verschiedene PDF (unter welchen nur Zweitere alle textlichen Inhalte offenbart). Ähnliches kann auch innerhalb  $\text{T}_{\text{E}}\text{X}$  geschehen, sollte

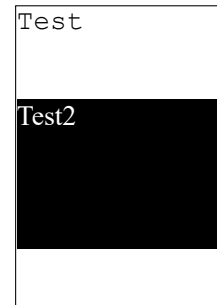
```
<html>
  <head>
    <title>Example</title>
    <style>
      /*formatting options are: none and black*/
      .t{
        font-size:13em;
        height:50%;
      }
      /*formatting option: none = no background, black, courier*/
      .t#none{
        font-family: 'Courier New', Courier, monospace;
      }
      /*formatting option: black = black background, white, serif*/
      .t#black{
        background-color:black;
        color:white;
        margin-top: -2em;
      }
    </style>
  </head>
  <body>
    <div class="t" id="none">Test</div>
    <div class="t" id="black">Test2</div>
  </body>
</html>
```

Abbildung 1: HTML-Beschreibung einer Webseite mit zwei Textflächen

Abbildung 2: Um die Dokumente von der restlichen Papierfläche abzugrenzen wurden schwarze Rahmen mittels TikZ hinzugefügt.



(a) Zuvorige HTML-Beschreibung liefert bei einer Browser-Skalierung von 120% obige Graphik



(b) Zuvorige HTML-Beschreibung liefert bei einer Browser-Skalierung von 50% obige Graphik