

# 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 Abgrenzung . . . . .	1
<b>2</b>	<b>Problemfälle</b>	<b>3</b>
2.1	Hinweise für Leser . . . . .	3
2.2	Elemente eines Dokumentes . . . . .	5
2.2.1	Paragraphen und Abschnitte . . . . .	5
<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 Abgrenzung

 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.

Aufgrund bekannter Technologien, wie z.B. Google Translate, DeepL und co. sollten solche Fehler innerhalb einzelner Wörter als ein „gelöstes Problem“ betrachtbar sein. Sprachliche Missverständnisse könnten aber immer dann entstehen, wenn sich mehrere Sprachen miteinander vermischen, welche sich ein Lexikon teilen (bspw. hier:  $\text{T}_{\text{E}}\text{X}$ ,  $\text{\LaTeX}$ , ... und die, zunächst, englische Sprache). Die Frage ob ein Wort übersetzt werden darf oder nicht, unterscheidet einzelne Problemarten (Fälle). Die Hoffnung besteht, dass diese Probleme bereits gelöst sind, allerdings soll *ein Übersetzer* in der folgenden Schilderung alle Fehler machen *dürfen*.

## 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 unbedachtes, zufälliges sequenzielles Nennen dieser könnte logische Lücken produzieren und damit potentielle Fehler produzieren. Deshalb wird eine Reihenfolge gewählt, welche nicht auf  $\text{\LaTeX}$ -Ebene beginnt, sondern sich so weit wie möglich dem Ursprung dieses Systemes nähert. Von den bereits in  $\text{T}_{\text{E}}\text{X}$  auftretenden Problemen muss ein Weg in Richtung der auf dieser Software aufbauenden Technologien gebahnt werden. Da es sich der Gesamtheit der Quelltextdateien, auf welche ein Kompilervorgang von  $\text{T}_{\text{E}}\text{X}$  zugreifen kann, immer um reine Textdateien handelt, werden spätere Beispiele nicht nach einzelnen Technologien betrachtet, sondern nach ihren Use-Cases (als relevante Systeme kämen hier zunächst  $\text{\LaTeX}$ ,  $\text{BibT}_{\text{E}}\text{X}$ ,  $\text{TikZ}$  und Weiterführende in Frage, welche teils andere Probleme nach sich ziehen). Eine Fehlerunterscheidung findet nach Funktionalität in einem reellen Dokument (das nach dem Kompilieren entstehende) und einem virtuellen Dokument (dessen Inhalte abhängig von anderen Dateien und Technologien sind). Beispiele in reellen Dokumenten sind nach den Strukturen sortiert, welche man in Dokumenten beliebiger Natur wiederfinden kann. Als kleinste Struktur würde man hier (abgesehen von einzelnen Worten und Sätzen) Paragraphen sehen, aus welchen sich Abschnitte eines Dokumentes zusammensetzen. Mehrere dieser Abschnitte ergeben einen größeren Abschnitt. Statt von einem „Überabschnitt“ zu sprechen, wird daher die Formulierung umgekehrt. Ein Dokument ist daher (zunächst) ein großer Abschnitt, welcher sich in verschiedene Unterabschnitte teilt, deren Namensgebung individuell sein kann. Hier wird (ähnlich wie bei: Knuth (1986)) von Kapiteln, Abschnitten, Unterabschnitten und Paragraphen gesprochen, jedoch mögliche „Unterunterabschnitte“ nicht als einzelne logische Struktur betrachtet, sondern als Unterabschnitt eines Unterabschnittes. Die möglicherweise entstehenden Fehler in einem „virtuellen“ Dokument werden nach den Teilen des Dokumentes klassifiziert, welche sie verändern (sollen). Hierbei können entweder einzelne Paragraphen angepasst werden (reiner Fließtext ohne vorgegebene Struktur), Literaturverzeichnisse oder Glossare entstehen (reiner Fließtext mit vorgegebener Struktur), Bilder und Graphiken eingebunden werden oder erstellt werden (Fließtexte innerhalb einer vorgegebenen Struktur), als auch Graphiken innerhalb eines Dokumentes an vorgesehenen Stellen beschreiben sein (Fließtexte in einer losen Struktur) und insbesondere letzteres zu diversen Verschachtelungen führen. Abschließend muss dann allerdings ein Unterpunkt, welcher nach der beschriebenen Reihenfolge in einem der ersten Abschnitte zu erwarten wäre, an das Ende gestellt werden, da aus diesem zu viele neue, eigene Probleme entstehen könnten. Zudem sind ein paar zusätzliche, sprachliche und teils unlösbare Probleme gelistet, welche nicht unbedingt als Anforderungen der gegebenen Problemstellung zu verstehen sind und daher als „abweichend“ zu verstehen sind, aus welchen sich aber spätere Erweiterungspotentiale zeigen könnten.

#### Struktur eines Beispiels

Die Darstellung einzelner Beispiele erfolgt tabellarisch und demonstriert erst richtiges (bzw. zulässige) Verhalten und danach Fehlerhaftes (bzw. Unerwünschtes). Untiges Beispiel (Tabelle 1) dient hierbei als einfaches Bei-

spiel für Fehler, die sich bei einem imaginären Befehl `ink` zeigen könnten. Dieser soll einen String mit einer bestimmten Farbe hinterlegen und besitzt einen zusätzlichen optionalen Farbparameter. „Richtig“ wäre es im originalen String nur das Wort in den geschweiften Klammern zu übersetzen, da hierbei an keiner Stelle Information verloren geht und das Wort, nachdem es vom Deutschen ins Englische übersetzt wurde, weiterhin so wie vorgesehen hervorgehoben wird. „Zulässige“ Übersetzungen treten dann auf, wenn nur für die Formatierung (insofern hieraus keine weiteren Probleme entstehen) verloren geht. Im gegebenen Beispiel würde dann zwar die farbige Hinterlegung verloren gehen, das Wort allerdings trotzdem übersetzt werden und würde den Weg in ein Dokument finden, ohne einen sprachlichen Informationsverlust zu riskieren (für den Endnutzer/Leser).<sup>1</sup> „Unerwünscht“ sind Fälle, in denen ein Übersetzen Fehler für die T<sub>E</sub>X-Engine produziert. Übersetzt man hier z.B. `ink` nach `Tinte` könnte es sich bei Zweitem wiederum um einen anderen Befehl handeln, das Wort *Wort* einliest, aber eigentlich den alphanumerischen Wert von *word* erwartet hätte.<sup>2</sup> Ein Fehlschlagen des Befehls `Tinte` würde zwar einen Fehler für den T<sub>E</sub>X-Parser produzieren, dieser wüsste dann aber, dass dieser Befehl bereits einmal fehlgeschlagen ist und eine neues Kompilieren verlangen, in welchem dieser Befehl und seine Optionen ignoriert werden, wodurch das Wort auch hier im Dokument landen würde. Man kann allerdings nicht bei jedem beliebigen T<sub>E</sub>X-Befehl davon ausgehen, dass dieses Verhalten einheitlich auftreten wird. Hierbei existieren Fälle, welche dafür sorgen könnten, dass andere Wörter nun nicht mehr Teil eines Dokumentes werden könnten ?? „Fehlerhaftes“ Verhalten beim Übersetzen von T<sub>E</sub>X-Quelltextdateien führt zu einem Informationsverlust, da das zu übersetzende Wort entweder nicht mehr übersetzt wird oder nicht mehr im Dokument wiederzufinden ist. Sobald man beginnt mit mehreren Dateien ein einziges Dokument zu beschreiben, riskiert ein naïves Übersetzen nur von einem Quelltext ausgehend, dass aus unerwünschten Fehlern innerhalb von einem Dokument fehlerhaftes Verhalten für das entstehende Produkt (meint: die kompilierte PDF) entsteht.

Abstrahiert man von diesem detaillierterem Beispiel, so sind „richtige“ Übersetzungen frei von Informationsverlust, „zulässige“ Übersetzungen nur dazu fähig Informationen für die graphische Aufbereitung (allerdings nicht den sprachlichen Inhalten) zu entwenden, „unerwünschte“ Übersetzungen dazu in Lage Informationen verbergen können und „falsche“ Übersetzungen fehlende sprachliche Inhalte innerhalb eines Dokumentes, sowie fehlende Übersetzungen dieser. Nicht jede Gruppe von Beispielen führt dazu, dass alle benannten Kategorien auftreten.

---

<sup>1</sup>Selbst bei weißer Schriftfarbe kann das Wort in einem PDF-Reader markiert und kopiert werden.

<sup>2</sup> $57_{16} + 6f_{16} + 72_{16} + 74_{16} = 25 \times 16^1 + 28 = 400$  statt:  $77_{16} + 6f_{16} + 72_{16} + 64_{16} = 26 \times 16^1 + 28 = 416$ . Wofür der Befehl `Tinte` einen/den Integer 416 benötigt, kann ich Ihnen allerdings nicht erläutern.

## **2.2 Elemente eines Dokumentes**

### **2.2.1 Paragraphen und Abschnitte**

Der zuvor etablierten logischen Struktur der Beispiellistung folgend, würde man zunächst einen Abschnitt oder Paragraphen erwarten. Technisch gesehen äußern sich die Beispiele allerdings alle sehr ähnlich. Hier erwartet man immer einen Befehl, welcher nicht übersetzt werden darf (da er Teil der logischen Struktur des Dokumentes ist) und einen Wert, welcher diesem Befehl übergeben ist und übersetzt werden soll. Das präsentierte Beispiel 2 zeigt hier wohlmöglich auftretende Fälle.



English	Mögliche Übersetzung
Richtiges Verhalten	
<div> <code>1\ink[red]{word}</code> </div> <div> <math>\text{\TeX}</math> Code 1: Original </div>	<div> <code>1\ink[red]{Wort}</code> </div> <div> <math>\text{\TeX}</math> Code 2: Beispielübersetzung </div>
Zulässiges Verhalten	
<div> <code>1\ink[red]{word}</code> </div> <div> <math>\text{\TeX}</math> Code 3: Original </div>	<div> <code>1\ink[Rot]{Wort}</code> </div> <div> <math>\text{\TeX}</math> Code 4: Beispielübersetzung </div>
Unerwünschtes Verhalten	
<div> <code>1\ink[red]{word}</code> </div> <div> <math>\text{\TeX}</math> Code 5: Original </div>	<div> <code>1\Tinte[Rot]{Wort}</code> </div> <div> <math>\text{\TeX}</math> Code 6: Beispielübersetzung </div>
Falsches Verhalten	
<div> <code>1\ink[red]{word}</code> </div> <div> <math>\text{\TeX}</math> Code 7: Original </div>	<div> <code>1\Tinte[Rot]{word}</code> </div> <div> <math>\text{\TeX}</math> Code 8: Beispielübersetzung </div>

Tabelle 1: Abstrakte Struktur der folgenden Beispiele

English	Mögliche Übersetzung
Richtiges Verhalten	
<div> <pre>1\paragraph{uninteresting}</pre> </div> <div> TeX Code 9: Original </div>	<div> <pre>1\paragraph{Uninteressant}</pre> </div> <div> TeX Code 10: Beispielübersetzung </div>
Unerwünschtes Verhalten	
<div> <pre>1\paragraph{uninteresting}</pre> </div> <div> TeX Code 11: Original </div>	<div> <pre>1\Paragraph{Uninteressant}</pre> </div> <div> TeX Code 12: Beispielübersetzung </div>
Falsches Verhalten	
<div> <pre>1\paragraph{uninteresting}</pre> </div> <div> TeX Code 13: Original </div>	<div> <pre>1\Paragraph{uninteresting}</pre> </div> <div> TeX Code 14: Beispielübersetzung </div>

Tabelle 2: Abstrakte Struktur der folgenden Beispiele

## **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

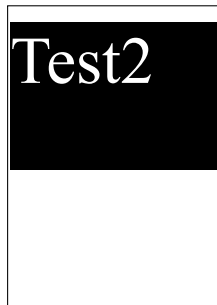
### A.1 Fontskalierung auf Webseiten

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 T<sub>E</sub>X geschehen, sollte

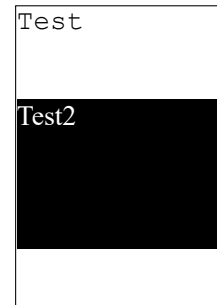
```
<html>
  <head>
    <title>Example</title>
    <style>
      /*formatting options are: none and black*/
      .t{
        font-size:13em;
        height:50%;
      }
      /*formatting option: none = no background, black, courier*/
      .t#none{
        font-family: 'Courier New', Courier, monospace;
      }
      /*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