Universität Rostock

Traditio et Innovatio

Automatische Sprachübersetzung von 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

Bachelorarbeit am Lehrstuhl für Informations- und Kommunikationsdienste

Fakultät für Elektrotechnik und Informatik

Abstrakt

placeholder

Inhaltsverzeichnis

1 Einleitung

1.1 Hintergrund

Herkömmliche Software zur Übersetzung von menschlicher Sprache auf TEX-Quellcode anzuwenden, erzeugt schnell Dokumente, welche entweder nicht vollständig übersetzt wurden oder sich nicht mehr kompilieren lassen. Mit Hilfe von Google Translate lassen sich wesentliche Gründe hierfür finden und wie sich diese äußern. Beispielsweise führt eine Übersetzung von `hello wor\textit{ld}` nicht zu `Hallo We\textit{lt}`, sondern zu `hallo wor\textit{ld}`. Abgesehen von der Frage, wo die kursive Hervorhebung im eigentlichen String erfolgen soll, werden Leser eines kompilierten Dokumentes das Wort „Welt" erkennen. Zuvor beschriebene Zeichenkette wird von TEX zu „hello wor*ld*" aufgelöst, in welcher das Wort „world" für einen menschlichen Leser als das englische Wort für „Welt" erkenntlich bleibt. Fehlt die Kenntnis über eine der Sprachen (DE,EN), würde einem monolingualen Leser Teil der Wortkette geraubt werden. Selbstverständlich sind die Wörter „world" und „Welt" einander sehr nahe und auch eine Formulierung der Art „Hallo Welt" lässt Vermutungen gegenüber eines größeren Kontexts zu.Anders wäre dies, wenn das Auslassen von auch nur einem Wort keine Rückschlüsse mehr auf einen größeren Kontext mehr zulässt. `$\mathbb{P}$robability density function` wäre ein denkbarer stilistischer Weg bereits in z.B. einem Folientitel bereits eine Notation für eine Wahrscheinlichkeitsdichtefunktion einzuführen. Hierbei würde der Verlust des Wortes „probability" den stochastischen Kontext aufheben. Der Verlust des Wortes „density" würde einen Kontext innerhalb der Stochastik verändern und ohne das Wort „function" ist fraglich, wovon die Rede ist. Vor allem in größeren Dokumenten könnten hierdurch Logikbrüche entstehen.

Eine Betrachtung eines „Übersetzers" als Konzept veranschaulicht die Problematik auf abstrakterer Ebene. Sollte der Kontext des Dokumentes unbekannt sein, werden sich unausweichlich semantische Fehler einschleichen. Bereits das gezeigte Beispiel könnte für z.B. eine Folie einer Lesung den restlichen Kontext der Seite entfernen und dadurch die Möglichkeit bieten umgangssprachliche Bedeutungen in Wörter zu interpretieren, anstatt einer Mathematischen (bspw. „ungerade" könnte im Englischen „crooked", statt „odd" produzieren). Noch weitere sprachliche Beispiele finden sich schnellig durch Wörter mit zeitlichem/räumlichen Bezug. Der Satz `Morgen wird es regnen.` könnte ohne das Wort „morgen" als Frage mit unzureichend eingehaltener deutscher Grammatik interpretiert werden. (*Wird es regnen?*). Hierbei verliert man eine getroffene Aussage über das Wetter, welches bekanntlicherweise nur schwer vorhergesagt werden kann.

1.2 Anforderungen

Genauso wie das Fehlen einzelner Wörter die sprachliche Bedeutung für einen Menschen brechen kann, treten ähnliche Probleme auch in TEX auf. Einzelne LATEX Makros *nicht* zu übersetzen ist unbedeutend, da sie ihre Bedeutung für einen TEX Compiler behalten. Alleine einzelne Wörter eines Makros zu übersetzen kann dazu führen, dass größere Inhalte (im Sinne: Menge an Worten) nicht mehr in einem kompilierten Dokument vorzufinden sind, was auf eine Fähigkeit von TEX zurückführbar ist. Die Möglichkeit in bestimmten Fällen eine Dateiendung auszulassen, führt beim Einbinden von anderen `.tex` Dateien in einem TEX Dokument zu fehlerhaften/fehlenden Ressourcenangaben. `\include{clock}` zu `\include{Uhr}` zu übersetzen (wie bspw. Google Translate am 06.10.2025) würde nun nicht mehr zu `\include{clock.tex}` aufgelöst werden, sondern zu `\include{Uhr.tex}` (bei welchem nicht davon auszugehen ist, dass diese Datei zur Kompilierzeit im System zwingend vorliegt).

Daher muss nach einer Lösung gesucht werden, welche diese technischen und sprachlichen Hürden überwinden kann. Neben solchen rein technischen Details, darf die Perspektive des Lesers (wörtlich) nicht missachtet bleiben und keine Übersetzungsprozesse dürfen zu versteckten Inhalten im Dokument führen. Diese Verbergung resultiert aus verschiedenensten Layouting-Problemen, ähnlich wie bei der Skalierung von Boxen auf Webseiten (Anhang A.1) und ist abhängig von einzelnen Sprachen dazu in der Lage unbemerkt verdeckte textliche Inhalte zu provozieren.[1]Wünschenswert ist neben vorigen Aspekten auch Möglichkeiten für den Endnutzer zu erlauben, sollte dieser spezielle Übersetzungen oder Kontexte für einige Wörter wünschen, welche jedoch nicht aus dem Dokument selbst hervorgehen. Außerdem sollte ein möglichst hoher Support für sowohl verschiedene menschliche Sprachen, aber auch verschiedene LATEX-Pakete gegeben sein, wobei Letzteres nur ein Bonus ist, sollten Systeme wie Ti*k*Z, bzw. `pgfplots` oder BibTEX innerhalb LATEX (zusammen mit TEX) nutzbar bleiben.

[1]Zwei adjazente Textfelder müssen sich zwangsläufig überlagern, wenn eines unabdingbar größer werden muss, da z.B. die Textgröße nicht verkleinert werden kann und das Wachstum eines Textfeldes nur in das Gebiet eines Anderen stattfinden kann. Dies wäre z.B. mit Hilfe von Inhaltsangaben auf Lebensmitteln vorstellbar. Sollten diese Wörter übersetzt werden und dadurch mehr Textfläche nach rechts benötigen, würden sie in den tabellarischen Bereich der eigentlichen quantitativen Angaben des z.B. Brennwertes, der Makro- sowie der Mikronährstoffe, hereinragen, wodurch das Risiko besteht, dass diese verdeckt werden.

2 Problemfälle

Eine einzige, feste TEX-Syntax existiert theoretisch gesehen nicht, wie ein späterer Paragraph aufzeigen wird.Die Fähigkeit jegliche erdenkliche Zeichenkette (gegeben: diese ist auf einem Rechner darstellbar, siehe: Unicode Consortium (2025)) sorgt zunächst für eine unendliche Menge an testbaren Problemen. Da es unmöglich ist eine unendliche Menge an Testfällen abzudecken, wird zunächst nur die vorgesehene LATEX (bzw. TEX Syntax nach Knuth (1986)) betrachtet und die bereits rein innerhalb dieser schnellig auffallenden Fehler aufgezeigt, welche durch fälschlich übersetzte Zeichenketten entstehen könnten.

2.1 Struktur

2.1.1 Klassifizierung

Für spätere Testzwecke wurden die verschiedenen Problemfälle in einzelne Kategorien getrennt. Eine solche Kategorisierung ist technisch gesehen nicht zwingend, soll allerdings zur Verbesserung einer späteren Übersicht dienen. Die Einteilung konzentriert sich vorrangig auf die Komplexität des Problemes und in einer Reihenfolge, in welcher sie auch bei der Nutzung von TEX für ein beliebiges Dokument auftreten könnten. Hierbei seien *direkte Probleme* zunächst eher simple und technisch leicht zu behebende Probleme, welche sich durch ein einheitliches Vorgehen beheben lassen könnten.Zudem beschränkt man sich hier nur auf den benötigten Zugriff auf eine einzelne Datei. *Indirekte Probleme* formen potentielle Schwierigkeiten, da sie einen Zugriff auf weitere Dateien benötigen, da sie von einer Ausgangsdatei referenziert werden. Technisch gesehen sind sie jedoch auch durch einheitliche Paradigma zu bewältigen. Daher werden sie im Kapitel 2.2 zusammengefasst und fortan als „simple Probleme“ bezeichnet. „Fortgeschrittene Probleme“ beziehen sich auf Probleme, welche sich paradigmatisch lösen lassen, aber zusätzliches Vorwissen verlangen. Dieses Vorwissen lässt sich bei diesen Problemfällen jedoch ermitteln, da der Entstehungsweg dieses Wissens nachvollziehbar ist.„Speziellere Probleme“ umfassen Probleme, welche sich nicht ohne Vorkenntnis des tatsächlichen Dokumentes beheben lassen. Nach diesen wird zudem noch auf ein paar rein sprachliche Schwierigkeiten eingegangen, welche unter Anderem zu solchen speziellen Problemen führen könnten.

2.1.2 Beschreibung

Jeder Problemfall wird zunächst durch ein Beispiel demonstriert, danach wörtlich erläutert und wird abschließend abstrahiert anhand konkreter TEX Primitiven zusammengefasst.

2.2 Simple Probleme

2.3 Fortgeschrittenere Probleme

Placeholder. . .

2.4 Spezielle Probleme

3 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.

Unicode Consortium (2025), 'Unicode: The world standard for text and emoji', https://home.unicode.org/ (last Access: 06.10.2025).

A Anhänge

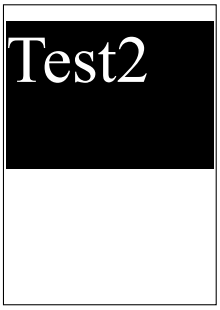
A.1 Fontskalierung auf Webseiten

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 TEX geschehen, sollte

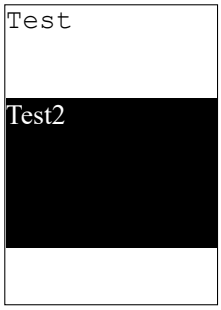
```
<html>
    <head>
        <title>Example</title>
        <style>
            /*formatting options are: none and black*/
            .t{
                font-size:13em;
                height:50%;
            }
            /*formatting option: none = no background, black, courier*/
            .t#none{
                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 Ti*k*Z 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