

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

Abstrakt

placeholder

Inhaltsverzeichnis

1	Einleitung	1
1.1	Hintergrund	1
1.2	Anforderungen	2
2	Problemfälle	3
2.1	Struktur	3
2.1.1	Klassifizierung	3
2.1.2	Beschreibung	3
2.2	Translative	4
2.2.1	Unbekannte Wörter	4
2.2.2	Auszulassende Wörter	4
2.2.3	Neuartige Satzarten	5
2.2.4	Größere Sprachstrukturen	6
2.3	Technische Probleme	6
2.3.1	Hilfsdateien	6
2.3.2	Unerreichbare Informationen	8
2.3.3	Figuren und Tabellen	9
2.3.4	Literaturverzeichnisse	9
2.3.5	Category Codes	9
2.4	Spezifischer Technologien	9
2.4.1	Kommentare	9
2.4.2	Dilemmatische Makros	10
2.4.3	TikZ und Layouting	10
2.4.4	Quellmehrsprachigkeit	10
2.5	Sprachliche Schwierigkeiten	10
2.5.1	Glossare und Nomenklaturen	10
2.5.2	Weitere	11
3	Technologien	12
3.1	Übersicht	12
3.1.1	Auflistung	12
3.1.2	Eingrenzung	12
3.1.3	Auswertung	12
3.2	Einschätzung	12
3.3	Fazit	12
4	Offene Problematiken	13
4.1	Verfolgte Ideen	13
4.2	Gelöste Probleme	13

4.3	Lessons Learned	13
4.4	Fazit	13
5	Fazit	14
5.1	Zusammenfassung	14
5.2	Ausblick	14
5.3	Weiteres	14
6	Eigenständigkeitserklärung	15
	Literatur	16
A	Anhänge	17
A.1	Fontskalierung auf Webseiten	17

1 Einleitung

1.1 Hintergrund

Der erste Satz dieses Werkes befindet sich noch in Arbeit. Vermutlich wird dieser mit einem Wort beginnen, welches seinerseits den Anfangsbuchstaben „a“ trägt (bsps. „Ausgehend (von)“ oder „Anders (als ... behaupten)“).

Herkömmliche Software zur Übersetzung von menschlicher Sprache auf T_EX-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 T_EX zu „hello world“ 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} probability 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 schnell 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 $\text{T}_{\text{E}}\text{X}$ auf. Einzelne \LaTeX Makros *nicht* zu übersetzen ist unbedeutend, da sie ihre Bedeutung für einen $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$ zurückführbar ist. Die Möglichkeit in bestimmten Fällen eine Dateiendung auszulassen, führt beim Einbinden von anderen `.tex` Dateien in einem $\text{T}_{\text{E}}\text{X}$ 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 verschiedensten 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.¹Wünschenswert 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 TikZ, bzw. pgfplots oder Bib $\text{T}_{\text{E}}\text{X}$ innerhalb \LaTeX (zusammen mit $\text{T}_{\text{E}}\text{X}$) nutzbar bleiben.

¹Zwei 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 T_EX-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 L^AT_EX (bzw. T_EX 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 Problems und in einer Reihenfolge, in welcher sie auch bei der Nutzung von T_EX 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 demonstriert², danach wörtlich in einer Beschreibung erläutert und wird abschließend abstrahiert und versucht auf konkrete T_EX Primitiven zurückgeführt zu werden. Die fortgeschritteneren Probleme bedürfen teilweise mehreren Beispielen zur Beschreibung, lassen sich jedoch auf ähnliche Äußerungen zurückführen. Einführende Beispiele werden in der Form ausführbarer Code wird zu ausführbarer Code übersetzt, aber ausführbarer Code zu fehlerhafter Code³ (1-dimensional). Einige Beispiele benötigen mitunter eine 2-dimensionale Darstellung, da sie mehrere Zeilen Quellcode umfassen und werden tabellarisch nebeneinander gestellt. Einige Probleme auf höherer Abstraktionsebene lassen sich aller-

²welche aus einem Test mit Hilfe von Google Translate in Firefox durchgeführt wurden (Oktober 2025)

³„Code“ bezieht sich in beiden Fällen auf T_EX-Quelltexte

dings nur schwierig veranschaulichen, wodurch Beispiele sich teils wieder auf Schilderungen von Situationen berufen, um diese Übersicht übersichtlich zu halten.

2.2 Translative

2.2.1 Unbekannte Wörter

`\label{problem:encounter:solve}` wird zu `\label{Problem:Begegnung:Lösung}` übersetzt, aber `\section{example}` zu `\Abschnitt{Beispiel}`.

Beschreibung

Abstrahierung Teile der T_EX-Syntax lassen sich anhand von `\`, `{`, `}`, `[`, `]`, `$`, `$$` oder `\%` erkennen und müssten daher ausgeschlossen werden. Man kann sich diese Art von Fehlern wie 0-dimensionale Fehler vorstellen, wobei die nullte Dimension hierbei bei einem einzelnen Wort beginnt (welche als Punkte zu verstehen sind).

2.2.2 Auszulassende Wörter

`\begin{myenvironment}[fontsize=red]` wird zu `\begin{myenvironment}[fontsize=red]` übersetzt, aber `\begin{myenvironment}[fontsize = red]` zu `\begin{myenvironment}[fontsize = rot]`.

Beschreibung

Abstrahierung Teile der T_EX-Syntax lassen sich nicht nur anhand der zuvor beschriebenen Zeichenketten erkennen, sondern lassen sich auch in Zeilen wiederfinden. Diese Art von Fehlern bahnt den Weg zu einer Dimension, wodurch nicht nur innerhalb eines Wortes (Punktes), sondern auch zwischen verschiedenen Punkten Fehler entstehen könnten (also innerhalb einer Zeile).

2.2.3 Neuartige Satzarten

Original	Übersetzung
Beispiel 1	
<pre>1 This is all \texttt{some} text. 2 \label{hello} 3 The following will only work, if both the label and reference values remain the same~\ref{hello}.</pre>	<pre>1 Dies ist alles \texttt{irgendein} Text. 2 \label{hallo} 3 Das Folgende funktioniert nur, wenn sowohl die Label- als auch die Referenzwerte gleich bleiben~\ref{hallo}.</pre>
Beispiel 2	
<pre>1 \hypersetup{ 2 urlcolor=red, 3 urlbordercolor={1 0 0}, 4 }</pre>	<pre>1 \hypersetup{ 2 URL-Farbe=rot, 3 URL-Rahmenfarbe={1 0 0}, 4 }</pre>

Tabelle 1: Beispiel für eine Zeile, welche übersetzt werden darf

Beschreibung

Abstrahierung Teile der $\text{T}_{\text{E}}\text{X}$ -Syntax lassen sich nicht nur anhand von einzelnen Zeilen oder Zeichenketten erkennen, sondern könnten sich auch in verschiedenen Zeilen wiederfinden lassen. Diese Art von Fehlern kann 2-dimensional betrachtet werden, wodurch Fehler auch zwischen Zeilen entstehen können.

2.2.4 Größere Sprachstrukturen

Original	Übersetzung
Datei <i>x</i>	
1 <code>\input{file}</code>	1 <code>\input{Datei}</code>
Datei <i>y</i>	
1 I could'nt have been translated, as google translate does not have remote access to my local files.	1 I could'nt have been translated, as google translate does not have remote access to my local files.
2 % This file doesn't need to be translated and couldn't have been, at least in the current way of demonstration.	2 % This file doesn't need to be translated and couldn't have been, at least in the current way of demonstration.
3 % Thus the left/right version of this text have to be the exact same.	3 % Thus the left/right version of this text have to be the exact same.

Tabelle 2: Beispiel für eine übersehene Datei

Beispiel

Beschreibung Die Übersetzung von Datei *x*, welche Datei *y* via `input` oder `include` führt zwar dazu, dass Datei *x* übersetzt wird, aber Datei *y* nicht.

Abstrahierung Teile der $\text{T}_{\text{E}}\text{X}$ -Syntax müssen nicht zwingend in einer Datei vorliegen, sondern könnten auch in verschiedenen Dateien integriert sein. Die Klassifizierung simpler Probleme gelangt in $\text{T}_{\text{E}}\text{X}$ hier bereits in der dritten Dimension an, weswegen sich fortan bereits mit „fortgeschrittenen“ Problemen beschäftigt werden muss (welche sich Teils über mehrere „Dimensionen“ erstrecken). Eine vierte Dimension existiert physikalisch nicht, ist jedoch mathematisch formulierbar⁴ und äußert sich in diesem Kontext auf eine Erhöhung von Laufzeitkomplexitäten.

2.3 Technische Probleme

2.3.1 Hilfsdateien

2.3.1.1 Struktur dieses Abschnittes Eine Datei nutzt ein Literaturverzeichnis (`BibTEX`), `pdfTEX` (`\pdfcomment{}`), `\footnote`, ein Inhaltsverzeichnis, ein Glossar,

⁴physikalisch: die vierte Dimension ist die Zeit, wenn man eine nicht-euklidische 3-dimensionale Bewegung verlangt (Teleportation)


2.3.1.2 Beispielliste

Inhaltsverzeichnisse, Abbildungsverzeichnisse, Tabellenlisten

Beispiel

Erläuterung Ändern wir den Titel eines Paragraphen oder Abschnittes, dann erfasst dies der T_EX Compiler beim ersten Durchlauf. Jedoch der String im Inhaltsverzeichnis kann nur verändert werden, sobald diese Information zu Beginn des nächsten Kompilierungsprozesses in der entsprechenden Hilfsdatei vorliegt (.toc).

2.3.1.2.1 Backrefs

Beispiel Bedarf evtl. einer bildlichen Veranschaulichung. Meint: Inhaltsverzeichnis, Tabelle, . . . vor einer *backwards reference* verschiebt die echte Position der (Phantom-) Sektion. Daher muss zunächst bestimmt werden, wo im Dokument eine Referenz auf ein existierendes Label stattfindet, welches vorherig bereits vergeben wurde 

Erläuterung Ein Verweis auf einen vorherigen Paragraphen kann nur klickbar verlinkt werden, wenn die Information, an welcher Stelle er sich im Dokument befindet, bereits klar ist. Da nach der Referenz weitere Abschnitte folgen können, welche vorherige Elemente mit variabler Größe verändern könnten, muss zunächst die Größe dieser bestimmt sein und gegenüber dieser kann dann der

Literaturverzeichnisse

Beispiel

Erläuterung

PDF Funktionen

Beispiel

Erläuterung

„footnotes“

Beispiel

Erläuterung

2.3.1.3 Abstrahiertes Problem Falls auf Hilfsdateien zugegriffen wird, ist das mehrfache Kompilieren eines \LaTeX Dokumentes unvermeidbar. Gegeben n_h womöglich existierenden Hilfsdateien, welche alle ihrerseits zu übersetzende Inhalte beinhalten können, folgt eine minimale Laufzeit $\mathcal{O}(n_h)$.

2.3.2 Unerreichbare Informationen

Original	Übersetzung
Dokument	
<pre> 1 \documentclass{standalone} 2 \usepackage{natbib} 3 \bibliographystyle{agsm} 4 \begin{document} 5 ~\cite{salomon_c} %Kapitel 5 6 refers to character arrays as a string. 7 \bibliography{example_original.bib} 8 \end{document} </pre>	<pre> 1 \documentclass{standalone} 2 \usepackage{natbib} 3 \bibliographystyle{agsm} 4 \begin{document} 5 ~\cite{salomon_c} %Kapitel 5 6 bezeichnet Zeichenarrays als Zeichenfolge. 7 \bibliography{example_original.bib} 8 \end{document} </pre>
Bibliothek	
<pre> 1 % Bad Citation, as the title is missing. All necessary information can be garnered by accessing the book behind the isbn. 2 @misc{salomon_c, 3 author={{Prof. Dr.-Ing. habil. Ralf Salomon}}, 4 year={2013}, 5 title={Siehe ISBN: 978-3-00-042684-1}, 6 } </pre>	<pre> 1 % Falsche Zitierung, da der Titel fehlt. Alle notwendigen Informationen finden Sie im Buch hinter der ISBN. 2 @misc{salomon_c, 3 author={{Prof. Dr.-Ing. habil. Ralf Salomon}}, 4 year={2013}, 5 title={Siehe ISBN: 978-3-00-042684-1}, 6 } </pre>

Tabelle 3: Beispiel für einen verpassten literarischen Kontext

Beispiele

Beschreibungen Ein Dokument erwähnt ein Werk, in welchem es um die C-Programmierung geht. Rein aus den im System vorliegenden Dateien ist kein Kontext für das Wort „String“ erkennbar, sodass ein Zugriff auf eine externe Ressource unabdingbar ist.

Abstrahierung Einfache Cloud-Architektur. Ein Client möchte auf ein beliebiges Wissen einer Webseite (bzw. dem Server und den beanspruchten Speicherplätzen in einem (beliebigen) Rechenzentrum⁵ zugreifen).

2.3.3 Figuren und Tabellen

Beispiele

Beschreibungen

Abstrahierung

2.3.4 Literaturverzeichnisse

Beispiele

Beschreibungen BibTeX erlaubt es an vielerlei Stelle eigene Strings in einer kompilierten TEX-Datei zu verbergen.

Abstrahierung

2.3.5 Category Codes

Beispiele

Beschreibungen

Abstrahierung

2.4 Spezifischer Technologien

Mitunter NP-schwer.

2.4.1 Kommentare

Beispiele

Beschreibungen Wohingegen sich 2.4.1 nicht mit anderen, in Kommentaren referenzierten, Dateien beschäftigt, soll sich hier auf solche Fälle konzentriert werde.

⁵Hierbei ist nicht von Festspeicher zu reden. Aus Sicherheitsgründen sei davon auszugehen, dass sich die physischen Adressen des wissensrepräsentierenden Speichers regelmäßig und unvorhersehbar ändern

Abstrahierung Hier treffen simple Fehler aus den ersten drei Kategorien (in 2.2.1, 2.2.2 und 2.2.3 geschildert) aufeinander. In die dritte Dimension, also in andere Dateien, wird jedoch (vorerst) nicht traversiert, da auskommentierte Datei-Einbindungen nicht erfasst werden dürften. Ausgehend von ?? wird nun erwartet, dass eine Referenzierung von Dateien erwartet wird, welche sich in Kommentaren verbergen. Dies kann jedoch 2.4.4 beinhalten.

2.4.2 Dilemmatische Makros

Beispiele

Beschreibungen

Abstrahierung

2.4.3 TikZ und Layouting

Beispiele

Beschreibungen

Abstrahierung

2.4.4 Quellmehrsprachigkeit

Beispiele

Beschreibungen

Abstrahierung Quelltexte anderer Quellsprachen (Programmiersprachen) können ihrerseits auf andere Dateien verweisen, oder andere Syntaktik tragen. Das Erkennen dieser ist theoretisch gesehen leicht, jedoch praktisch gesehen schnell zu übersehen.

2.5 Sprachliche Schwierigkeiten

2.5.1 Glossare und Nomenklaturen

Beispiele

Beschreibungen

Abstrahierung

2.5.2 Weitere

Beispiele

Beschreibungen

Abstrahierung

3 Technologien

3.1 Übersicht

3.1.1 Auflistung

3.1.2 Eingrenzung

3.1.3 Auswertung

3.2 Einschätzung

3.3 Fazit

4 Offene Problematiken

4.1 Verfolgte Ideen

4.2 Gelöste Probleme

4.3 Lessons Learned

4.4 Fazit

5 Fazit

5.1 Zusammenfassung

5.2 Ausblick

5.3 Weiteres

6 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

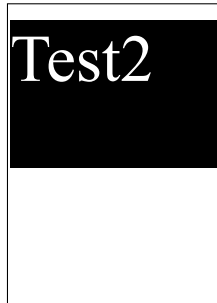
A.1 Fontskalierung auf Webseiten

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 $\text{T}_{\text{E}}\text{X}$ geschehen, sollte

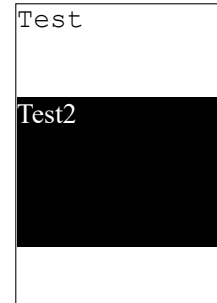
```
<html>
  <head>
    <title>Example</title>
    <style>
      /*formatting options are: none and black*/
      .t{
        font-size:13em;
        height:50%;
      }
      /*formatting option: none = no background, black, courier*/
      .t#none{
        font-family: 'Courier New', Courier, monospace;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