

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	Technische Bedeutung	3
2.1.1	Sonderzeichen	3
2.1.2	Leerzeichen	4
2.1.3	Zeilenbrüche	6
2.1.4	Dokumentenbrüche	7
2.2	Sprachliche Semantik	8
2.2.1	Ausgangssituation	8
2.2.2	Exemplarische Beispiele	8
2.3	Spezifischer Technologien	9
2.3.1	Kommentare	9
2.3.2	Dilemmatische Makros	9
2.3.3	TikZ und Layouting	9
2.3.4	Quellmehrsprachigkeit	9
2.4	Weitere Schwierigkeiten	10
2.4.1	Glossare und Nomenklaturen	10
2.4.2	Weitere	10
3	Technologien	11
3.1	Anforderungen	11
3.2	Denkbare Ansätze	11
3.3	Existierende Ansätze	11
3.4	Tests	11
3.5	Grenzen der Lösungen	11
3.6	Takeaways	11
4	Ungelöst	12
4.1	Verfolgter Lösungsweg	12
4.2	Gelöste Probleme	12
4.3	Lessons Learned	12
4.4	Fazit	12
5	Fazit	13
5.1	Zusammenfassung	13
5.2	Ausblick	13
5.3	Weiteres	13


6	Eigenständigkeitserklärung	14
	Literatur	15
A	Anhänge	16
A.1	Fontskalierung auf Webseiten	16

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)“).

Software zur Übersetzung von menschlicher Sprache auf T_EX-Quellcode anzuwenden, kann Dokumente erzeugen, welche die T_EX-Syntax brechen oder nur zu Teilen und damit unvollständig übersetzt wurden. Mit Hilfe von Google Translate lassen sich wesentliche Gründe ermitteln und welche weitere Folgen aus einem Übersetzungsprozess entstehen können, der nicht von einem L^AT_EX-Dokument ausgeht (mit: T_EX, der Programmiersprache für *typesetting* (Zeichensetzung) von Knuth (1986) sowie der Erweiterung des funktionellen Umfangs durch L^Ampport (1994)). Bereits einfachste Funktionen dieses Systems können Übersetzungen einfacher Zeichenketten verhindern. Ein Übersetzen (via Google Translate) von `hello world` liefert nicht `Hallo Welt`, sondern `hallo world`. Abgesehen von der Frage, wo die kursive Hervorhebung im eigentlichen String erfolgen sollte, würden Leser eines kompilierten Dokumentes das Wort „Welt“ erkennen. Die beschriebene Zeichenkette wird von T_EX als „hello world“ dargestellt, in welcher das Wort „world“ für einen menschlichen Leser als das englische Wort für „Welt“ eindeutig erkennbar ist. Fehlt die Kenntnis über eine der Sprachen (DE, EN) oder ein natürliches Sprachverständnis verliert die Wortkette einen Teil ihrer Bedeutung. Besonders fatal wird 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 \TeX auf. Einzelne \LaTeX Makros zu übersetzen kann einen semantischen Verlust für einen \TeX Compiler mit sich führen. Als einfaches Beispiel zeigt sich hier die Möglichkeit in bestimmten Fällen eine Dateiendung auszulassen, sollte man ein \LaTeX Dokument in mehrere \TeX Dateien trennen wollen. $\text{\code{\include{clock}}}$ zu $\text{\code{\include{Uhr}}}$ zu übersetzen (wie bspw. Google Translate am 06.10.2025) würde nun nicht mehr als $\text{\code{\include{clock.tex}}}$ interpretiert werden, sondern als $\text{\code{\include{Uhr.tex}}}$ (dessen Existenz nicht garantierbar ist).

Daher muss nach einer Lösung gesucht werden, welche diese technischen und sprachlichen Hürden überwinden kann. Neben diesen gänzlich technischen Details, darf die Perspektive des Lesers (wörtlich) nicht missachtet bleiben und kein Übersetzungsprozess darf zu einem „verstecktem“ Inhalten führen. Diese Verbergungen von evtl. relevanten Informationen resultieren aus verschiedensten Layouting-Problemen, ähnlich wie bei der Skalierung von Boxen auf Webseiten (Anhang A.1) und ist von den jeweiligen Paaren an Sprachen abhängig (zwischen welchen übersetzt wird).¹Wünschenswert ist neben vorigen Aspekten auch Möglichkeiten für einen Endnutzer zu finden, welche die Möglichkeit bewahren manuelle Anpassungen vorzunehmen, insofern dies gewünscht ist. Außerdem sollte ein möglichst hoher Support für verschiedene menschliche Sprachen, aber auch verschiedene \LaTeX -Pakete gegeben sein, wobei Letzteres nur ein Bonus  ist, sollten Systeme wie TikZ, bzw. pgfplots oder Bib \TeX innerhalb des \LaTeX Dokumentes nutzbar bleiben. 

¹Zwei adjazente Textfelder müssen sich zwangsläufig überlagern, wenn eine dieser Flächen größer werden muss, da z.B. die Textgröße nicht verkleinert werden kann und das Wachstum dieses Textfeldes nur in das Gebiet einer Anderen stattfinden kann. Dies wäre z.B. mit Hilfe von Inhaltsangaben auf Lebensmitteln vorstellbar. Sollten die Wörter der ersten Spalte übersetzt werden und dadurch mehr Textfläche nach rechts benötigen, entstehen beschriebene Überlappungen.

2 Problemfälle

Sprachliche Uneindeutigkeiten können in vielen Sprachen auftreten und dadurch Missverständnisse produzieren (bspw. sarkastische Kommentare oder Mehrdeutigkeiten einzelner Worte). Besonders kritisch sind solche Uneindeutigkeiten jedoch für übersetzende Programme, welche Dokumente in eine andere, menschliche Sprache überführen zu suchen, da hier bereits einzelne missinterpretierte Wörter die T_EX Syntax brechen könnten, wodurch nur noch ein unzureichender Teil der Beschreibung des Dokumentes bestehen bleibt, aus welcher kein echtes Dokument entstehen kann. Das Einhalten der T_EX Syntax alleine genügt jedoch nicht, um erwähnte Mehrdeutigkeiten von einzelnen Wörtern zu verhindern, da sie abhängig ihres Kontexts eine andere Übersetzung verlangen. Weiterhin können innerhalb von verschiedenen, in Kombination mit T_EX genutzten Systeme einzelne „Fehler“ entstehen. Ansätze zur Behebung dieser sind jedoch in manchen Fällen bereits konzeptionell unmöglich. Hieran anknüpfend existieren einige sehr spezifische sprachliche

Aus kurzer Nachverfolgung und Einschätzung dieser Probleme werden anschließend konkrete Anforderungen gestellt. Eine Erläuterung dieser Art macht es allerdings unabdingbar eine Software zu nutzen, welche auf das Übersetzen spezialisiert ist und nicht von sich aus L^AT_EX oder T_EX-konforme Dokumente erwartet. Daher werden Beispiele anhand von Google's Web-Service „Translate“ aufgeführt.

2.1 Technische Bedeutung

2.1.1 Sonderzeichen

Manche menschliche Sprachen beinhalten gelegentlich Zeichen, welche keine direkte Bedeutung tragen und kein Teil auch nur eines Wortes der Sprache sind. Denkbare Beispiele hierfür sind Klammern, die (üblicherweise) für ein impliziert Erwähntes, allerdings nicht vorzeitig verwendetes Wort genutzt werden. Stilistische Mittel für Texte sind prinzipiell nicht an spezifische Zeichen gebunden und ein Programm kann immer davon ausgehen, dass ein „Tippfehler“ entstehen kann. Beispiel 1 zeigt allerdings, wie vereinzelte Zeichenketten produzieren können, welche fälschlich (in diesem Kontext: unerwünscht) übersetzt werden.

Vermutung Fraglich ist, warum `label` nicht erfasst werden sollte, obwohl die folgenden drei Wörter übersetzt werden. Ein String, welcher menschliche Sprache mit Sonderzeichen vermischt, kann dahingehend interpretiert werden, dass diese Sonderzeichen wie Klammern verwendet werden. Seinerseits könnte `:` also nicht als bekanntes „geteilt“ aufgefasst werden, sondern als Klammern. Ersetzt man diese Klammern mit Leerzeichen resultiert aus dem ersten Beispiel `\label problem encounter solve` und in zweitem Beispiel `\section example`. Zu sehen ist hier also bereits, dass Google Translate bei einer, wenn man es so interpretieren möchte, „Vernestung“ zweiten Grades scheitert, jedoch einfache Vernestungen noch erkennt².

²„Vernestung“ meint die Verschachtelung von Klammern

Original	Übersetzung
Korrekt	
<pre>1\label{problem:encounter:solve}</pre> <p>T_EX Code 1: Test</p>	<pre>1\label{Problem:Begegnung:Lösung}</pre> <p>T_EX Code 2: Test</p>
Unerwünscht	
<pre>1\section{example}</pre> <p>T_EX Code 3: Test</p>	<pre>1\Abschnitt{Beispiel}</pre> <p>T_EX Code 4: Test</p>

Tabelle 1: Der für einen T_EX Compiler relevante Befehl `label` bleibt unverändert, allerdings `section` wird fälschlicherweise als Abschnitt übersetzt

Takeaway Teile der T_EX-Syntax lassen sich anhand von `\`, `{`, `}`, `[`, `]`, `$`, `$$` oder `\%` erkennen und müssten daher ausgeschlossen werden. Anders als in mathematischen Formeln zeigen sich Sonderzeichen jedoch nicht paarweise auf, sodass sie nicht paarweise ignoriert werden können. 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 verstanden werden).

2.1.2 Leerzeichen

Dem vorangegangenen Beispiel (zunächst) widersprechend, offenbart ein Auslassen von Zeichen (und dadurch ein Trennen von Worten) eine Vielzahl anderer möglicher Fehler, welche sich glücklicherweise schnell einheitlich beschreiben lassen. Hierunter fallen meist freistehende Wörter, welche als Parameter für verschiedene L^AT_EX Umgebungen dienen. Die Übersetzung solcher Parameter kann schnell zu Fehlern in einem T_EX-Compiler führen.

Verdeutlichung Die Optionen innerhalb eckiger Klammern lassen Whitespace zu. Dies kann jedoch für die Nutzung einiger Funktionen in z.B. wichtigen Paketen wie `hyperref` dazu führen, dass falsche Wörter übersetzt werden, die ein Kompilieren des Dokumentes verhindern.

Original	Übersetzung
Korrekt	
<pre>1\usepackage[urlbordercolor=red]{ ↪hyperref}</pre> <p>T_EX Code 5: Test</p>	<pre>1\usepackage[urlbordercolor=red]{ ↪hyperref}</pre> <p>T_EX Code 6: Test</p>
Unerwünscht	
<pre>1\usepackage[urlbordercolor = red]{ ↪hyperref}</pre> <p>T_EX Code 7: Test</p>	<pre>1\usepackage[urlbordercolor = rot]{ ↪hyperref}</pre> <p>T_EX Code 8: Test</p>

Tabelle 2: Fehler in einem einzeiligen Dokument

Takeaway 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).

2.1.3 Zeilenbrüche

Abstrahiert man nun über einzelne Zeilen hinweg, so wird die folgende Art von Fehlerquelle direkt offensichtlich, sodass sie keiner detaillierten Schilderung mehr bedarf. Sollte man versuchen ein Dokument Zeile nach Zeile zu übersetzen und den Kontext der vorigen Zeile zu ignorieren, so werden schnellig Zeilen übersetzt (obwohl: diese Zeilen nicht übersetzt werden durften, da sie Befehle für T_EX beinhalten (siehe: 3)).

Original	Übersetzung
Korrekt	
<pre> 1This is all \texttt{some} text.\label{ ↳hello} 2The following will only work, if both ↳the label and reference values ↳remain the same~\ref{hello}. </pre> <p>T_EX Code 9: Test</p>	<pre> 1Dies ist alles \texttt{irgendein} Text ↳.\label{hallo} 2Das Folgende funktioniert nur, wenn ↳sowohl die Label- als auch die ↳Referenzwerte gleich bleiben~\ref{ ↳hallo}. </pre> <p>T_EX Code 10: Test</p>
Unerwünscht	
<pre> 1\hypersetup{ 2 urlcolor=red, 3 urlbordercolor={1 0 0}, 4} </pre> <p>T_EX Code 11: Test</p>	<pre> 1\hypersetup{ 2URL-Farbe=rot, 3URL-Rahmenfarbe={1 0 0}, 4} </pre> <p>T_EX Code 12: Test</p>

Tabelle 3: Das Übersetzen der hyperref Optionen würde den Kompilier-Prozess scheitern lassen

2.1.4 Dokumentenbrüche

Noch abstrakter wird hier die dritte Dimension erreicht, indem Teile von T_EX nicht in einer einzelnen Datei vorliegen müssen, sondern auch in anderen Dateien vorliegen könnten. Diese Tatsache wirft eine Vielzahl neuer (und teilweise system-abhängiger) Probleme auf. Hier wird sich zunächst jedoch nur auf die Fähigkeiten der unveränderten T_EX-Engine konzentriert und deren vorgesehene Primitiven (für diesen Zweck).

Original	Übersetzung
Unerwünscht	
<pre>1\input{file}</pre> <p>T_EX Code 13: Test</p>	<pre>1\input{Datei}</pre> <p>T_EX Code 14: Test</p>
Übersehen	
<pre>1I could'nt have been translated, as ↳google translate does not have ↳remote access to my local files.</pre> <p>T_EX Code 15: Test</p>	<pre>1I could'nt have been translated, as ↳google translate does not have ↳remote access to my local files.</pre> <p>T_EX Code 16: Test</p>

Tabelle 4: Übersetzung in einem include führt zum nicht-Übersetzen einer Datei. Ein zunächst nicht sonderlich „interessant“ wirkendes Beispiel

2.2 Sprachliche Semantik

2.2.1 Ausgangssituation

Probleme des vorherigen Teils 2.1 zu verhindern, stellt sowohl ein Kompilieren/Entstehen eines übersetzten Dokumentes sicher, als auch ein Erkennen aller Inhalte von diesem. Dies ist jedoch als minimale Anforderung zu sehen. Die Fähigkeit alle Inhalte eines \LaTeX Dokumentes lesen zu können allein, zeigt sich jedoch als unzureichend, da das Übersetzen zwischen menschlichen Sprachen kontextuell in verschiedenen Lexemen (Wörtern) enden sollte. Dieser Kontext kann einem Quellcode aus mehr als nur Wörtern entnommen werden. Das Bestimmen des Kontextes erfordert Kenntnis über *bestimmte Elemente* eines Dokumentes, welche nicht vorhersehbar sind und charakterisiert werden könnten. Elemente dieser Art können sich verschieden äußern und sind nicht zwangsweise standardisiert, sodass das Einhalten von eventuell etablierten Paradigma (hinsichtlich einiger institutionellen Kontexte) geprüft werden muss.

Vorangegangenes zeigt dementsprechend erneut eine üppige Menge an erwartbaren Problemen. Damit Grenzen der Realität erhalten bleiben, müssen sich denkbare Anwendungsfälle von \LaTeX auf (z.B.) wissenschaftliche Arbeiten oder Veröffentlichungen konzentrieren und die dafür benötigten Darstellungsmöglichkeiten in den Vordergrund rücken (bspw. Zitationen, Tabellen, Formeln, Graphiken, ...). Exemplarisch werden daher Beispiele aufgezeigt, deren Phrasen einen Kontext tragen, der nicht aus der sprachlichen Satzstellung allein hervorgeht.

2.2.2 Exemplarische Beispiele

Autoren/Titel Das Erwähnen des wortwörtlichen Namens oder Titels eines Dokumentes/Werkes wäre innerhalb dieses normalerweise nicht zu erwarten, aber aus diesem lassen sich unter Umständen mehr Informationen über den Kontext gewinnen (und: beide sind normalerweise in der Präambel eines Dokumentes vorhanden). Handelt es sich um einen bekannten Autoren könnte der Name einen Kontext liefern (welcher jedoch im Vorwege nur als eine Heuristik angesehen werden kann, da der Autor authentifiziert werden muss) oder der Titel trägt den Namen von einer bekannten Methode aus einem wissenschaftlichen Gebiet oder nennt ein solches Gebiet. Denkbare Beispiele umfassen: „[...] of Communication“, „Bayesian [...]“, „[...] inference [...]“ (entnommen aus einigen *subject areas* von ACM: Transactions on Probabilistic Machine Learning). Vergisst aber ein Übersetzendes Programm diese Wörter des Titels an einer kritischen Stelle im Text, so würden fälschliche Übersetzungen entstehen.

Dokumenten-interne Verweise Hyperreferenzen „erlauben“ zunächst (scheinbar) recht wenig, implizieren jedoch eine Giganz an Möglichkeiten. Es wäre zu erwarten, dass jegliche Information, welche für die Übersetzung eines Satzes erforderlich ist, textlich vorliegt. \TeX erlaubt es allerdings auf solch einen Kontext zu referenzieren (via: `ref` oder `hyperref`) und ihn dadurch zu *implizieren*, statt zu nennen.

Externe Referenzen In \LaTeX kann auf verschiedene Arten und Weisen auf anderweitige Quellen verwiesen sein.

Formeln, Tabellen

Graphiken

2.3 Spezifischer Technologien

Hier wenden wir uns von Problemen einer Übersetzung ab und widmen uns denen eines Lesers. Alle textlichen Inhalte eines Dokumentes zu übersetzen, als auch eine kontextuelle Fachsprache zu bewahren scheint aus abstrakterer Perspektive ausreichen, kann allerdings zu Situationen führen, in welchen Informationen verloren gehen, da diese vom Endnutzer nicht mehr gesehen werden können.

2.3.1 Kommentare

Beispiele

Beschreibungen Wohingegen sich 2.3.1 nicht mit anderen, in Kommentaren referenzierten, Dateien beschäftigt, soll sich hier auf solche Fälle konzentriert werde.

Abstrahierung Hier treffen technische Fehler aus den ersten drei Kategorien (in 2.1.1, 2.1.2 und 2.1.3 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 2.3.4 beinhalten.

2.3.2 Dilemmatische Makros

Beispiele

Beschreibungen

Abstrahierung

2.3.3 TikZ und Layouting

Beispiele

Beschreibungen

Abstrahierung

2.3.4 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 schnellig zu übersehen.

2.4 Weitere Schwierigkeiten

Kommentare

2.4.1 Glossare und Nomenklaturen

Beispiele

Beschreibungen

Abstrahierung

2.4.2 Weitere

Beispiele

Beschreibungen

Abstrahierung

3 Technologien

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.

3.4 Tests

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

3.5 Grenzen der Lösungen

3.6 Takeaways

4 Ungelöst

4.1 Verfolgter Lösungsweg

Ausgehend von den zuverigen Takeaways

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.

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).

A Anhänge

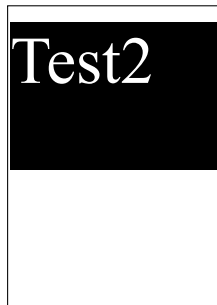
A.1 Fontskalierung auf Webseiten

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_EX geschehen, sollte

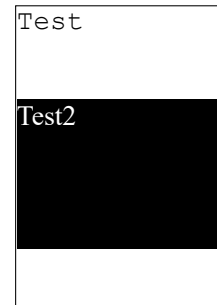
```
<html>
  <head>
    <title>Example</title>
    <style>
      /*formatting options are: none and black*/
      .t{
        font-size:13em;
        height:50%;
      }
      /*formatting option: none = no background, black, courier*/
      .t#none{
        font-family: 'Courier New', Courier, monospace;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