

Vergleich zwischen \TeX und herkömmlichen Textverarbeitungen

Ein sachlicher Vergleich

Gregor Barth

v2.06 vom 28. Februar 2019

Inhaltsverzeichnis

1	Vorwort	2
2	Konzeptbedingte Unterschiede in der Arbeitsweise	2
3	Anschaffungskosten	5
4	Technisches	5
4.1	Speicherplatz	5
4.2	Prozessor, Speicher, Grafik	6
4.3	Betriebssystem	7
4.4	Dateiformat	7
4.5	Systemschriftarten	8
4.6	Konfigurationsdateien	8
4.7	Die Sache mit den Bugs	9
5	Layout und Typographisches	9
5.1	Ligaturen, Alternativ-Buchstaben, Ziffern	11
5.2	Kapitälchen und Unterstreichungen	11
5.3	Rechtschreibprüfung	12
5.4	Formelsatz und Sonderzeichen	12
5.5	Zurichtung (Kerning)	13
5.6	Umfang der Dokumente	13
5.7	Verlinkte Abbildungen	14
6	Was \TeX kann, und Textverarbeitungen nicht	15

7 Was Textverarbeitungen können, und \TeX nicht	18
8 Was soll ich nun nehmen?	20
8.1 Nimm \LaTeX ...	20
8.2 Nimm eine herkömmliche Textverarbeitung ...	20

1 Vorwort

Obwohl ich unzweifelhaft Verfechter von \LaTeX und seinen Varianten bin, hat die Arbeit mit einer herkömmliche Textverarbeitung wie LibreOffice (LO) Writer doch einige Vorzüge, die im folgenden Vergleich zwischen beiden Systemen sachlich dargestellt werden sollen. Gerade unerfahrene Computer-Nutzer greifen für jede kleine Textarbeit gerne zur Textverarbeitung wie MS Word, ohne Alternativen zu bedenken, die einer solchen Textverarbeitung, insbesondere hinsichtlich Textsatz und Qualitätskontrolle, überlegen sind.

Im Folgenden wir von \TeX gesprochen, meint aber \LaTeX , $X_{\text{Y}}\TeX$, $\text{Lua}\TeX$ und andere Varianten, d. h. ein System, das aus einer \TeX -Distribution (Pakete-Sammlung, z. B. TeXLive) und einem beliebigen Text-Editor, sinnvollerweise einem speziellen \TeX -Editor besteht; wo der Unterschied von Bedeutung ist, wird darauf hingewiesen. Auf der anderen Seite wird mit dem Begriff »herkömmliche Textverarbeitung« diejenige Software gemeint, die man allgemein in jeder Büro-Software-Suite findet: LO Writer, MS Word, Calligra Words, Abiword etc. Wo der Unterschied von Bedeutung ist, wird darauf hingewiesen. Es werden in diesen Vergleich keine DTP-Programme wie Adobe InDesign, Quark Xpress oder Scribus hinzugezogen, da das den Vergleich sprengen würde.

2 Konzeptbedingte Unterschiede in der Arbeitsweise

Für die meisten Computer-Nutzer wird die herkömmliche Textverarbeitung (MS Word, LO Writer) einen eingängigen Arbeitsplatz bereitstellen: Nach Programmstart sieht man den Seitenumriss, und sobald man auf der Tastatur tippt, erscheinen Buchstaben der voreingestellten Schriftart dort, wo sie später auch gedruckt werden. Buttons für Farben, Schriftgröße, Zeilenabstand und Formatierungen (Kursivierung,

Fette, Unterstreichung etc.) befinden sich stets in Sichtweite und sind rasch bedient, um sein Dokument zu strukturieren. – Hier findet sich die erste Ursache, wie Nutzer unbewusst zu einer schlechten Typografie kommen: Sie stellen Seitenformate selbst ein, strukturieren ihren Text mit Auszeichnungen, ohne eigentlich zu wissen, was sie tun. Obwohl letztlich die ästhetische Empfindung des Auges darüber entscheidet, ob ein Text gut gesetzt ist, gibt es gewisse Regeln, die eingehalten werden sollten, um guter Lesbarkeit zu entsprechen:

- harmonische Seitenränder (z. B. nach dem Goldenem Schnitt oder ganzzahligen Seitenverhältnissen) und eine dazu passende Schriftgröße, sodass nicht zu viele Wörter pro Zeile gesetzt werden
- zur Schriftgröße passender Durchschuss
- eine für Mengentext angenehm lesbare Schriftart und ggf. dazu passende Schrift für Überschriften, Bildunterschriften usw.
- Anordnung von Bildern immer in Rückwirkung mit dem umgebenden Text (anstatt manuelle Anordnung nach Gutdünken)
- konsequente Anwendung von Absatzvorlagen etc.

Der große Nachteil bei dieser Arbeitsweise (WYSIWYG = *What you see is what you get*) ist, dass der Nutzer permanent, d. h. noch während des Schaffensprozesses, in Layout und Formatierung seines Textes eingreift, anstatt sich, wie es sein sollte, auf den bloßen Inhalt zu konzentrieren.

Dies ist der wesentliche Unterschied zur Arbeit mit \TeX : Hier tippt man seinen Text in einem Texteditor, meist in einer charakterlosen Monospace-Schriftart, und setzt lediglich Kommandos zur Auszeichnung (Kursiv, Fett) und Strukturierung (Überschriften) seines Textes. Von der Arbeit selbst lenkt das nicht ab, denn man sieht nicht sofort das Resultat. Erst nach dem Setzen (Kompilieren) kommt das Ausgabedokument (meist ein PDF) zum Vorschein, worauf der Text so erscheint, wie er nach allen Regeln der Setzerkunst gesetzt sein sollte: eine gut lesbare Schriftart, in passender Schriftgröße, harmonisierend mit dem Satzspiegel und sonstigen Elementen (Fußnoten, Bilder etc.). Ohne, dass man von Hand eingreifen muss (es aber natürlich kann).

Da der Anwender in einer herkömmlichen Textverarbeitung permanent das Layout mit den Augen prüft und verändert, erhöht sich auch der Fehler, dass Absatzvorlagen nicht konsequent eingehalten werden: Wie auch unter T_EX bietet eine herkömmliche Textverarbeitung wie LO Writer die Möglichkeit, den Text mithilfe von sog. Absatzvorlagen zu strukturieren. So werden alle Überschriften derselben Ebene und auch der gesamte Brotttext gleichartig (Schriftgröße, Schriftart, Farbe etc.) formatiert und wahlweise durchgängig nummeriert. Problematisch wird es, wenn Absatzvorlagen ungenutzt bleiben oder das Fehlen ihrer Anwendung nicht bemerkt wird. Was sich im Fließtext vielleicht noch als unproblematisch erweist, könnte aber zu einer fehlerhaften Nummerierung (auch im Inhaltsverzeichnis) der Überschriften führen. Auch können verwaiste, also leere Absatzvorlagen zwischen den Absätzen stecken, z. B. eine Überschrift ohne Inhalt, sodass im Inhaltsverzeichnis eine leere Zeile generiert wird. Konzeptbedingt hat sich der Anwender in einer herkömmlichen Textverarbeitung um all diese Dinge selbst zu kümmern¹.

Im Unterschied dazu sind derartige Fehler bei T_EX ausgeschlossen, oder zumindest nur mutwillig herbeizuführen: Allein durch die Arbeitsweise der »Kommandoklammerung« wird spätestens beim Kompilieren deutlich, wo Fehler gemacht worden sind. Unvollständig angewendete Absatzvorlagen gibt es daher nicht; auch bei der Nummerierung von Überschriften, Fußnoten, Abbildungen usf. wird niemals etwas durcheinander kommen (können).

Wenn sowohl in T_EX als auch in einer herkömmlichen Textverarbeitung Absatzvorlagen konsequent angewendet worden sind, ist es problemlos möglich, Formatierungsänderungen vorzunehmen: Alle Überschriften der 2. Ebene doch lieber in 16 pt, kursiv und in Grün? Kein Problem. – Das wäre nicht möglich, wenn der Anwender sich sein Layout selbst bauen würde, indem er beispielsweise manuell seine Überschriften nummeriert und direkt beim Eingeben derselben eine Schriftgröße und -art festlegt.

¹Kleine Erweiterungen wie der Pepito Cleaner (als Erweiterung für LibreOffice) checken abschließend das Dokument auf solche Unstimmigkeiten und geben eine Fehlerliste aus, die abgearbeitet werden kann.

3 Anschaffungskosten

Unter den herkömmlichen Textverarbeitungen gibt es kommerzielle und freie. Unter den kommerziellen ist MS Word sicherlich die bekannteste. Natürlich zahlt man auch für jedes Upgrade und ggf. auch für jeden Arbeitsplatz. Außerdem gibt es Unterschiede im Umfang, je nach Anwender (Studenten-Lizenz, Professionell). Neuerdings gibt es von MS Office auch ein Abo-Modell, was wiederum vom Internet abhängig macht (sofern man die angebundenen Cloud-Dienste nutzen möchte).

Freie (quelloffene) Textverarbeitungen (LibreOffice, Calligra-Suite, Abiword) und auch $\text{T}_{\text{E}}\text{X}$ sind dagegen frei von Kosten zu beziehen, natürlich auch alle Updates. Ebenso gibt es keine Einschränkungen hinsichtlich Anzahl der Arbeitsplätze oder ähnliches². Auch unter den $\text{T}_{\text{E}}\text{X}$ -Editoren gibt es freie und kommerzielle.

4 Technisches

In dieser Kategorie wird angenommen, dass das Textverarbeitungssystem auf dem heimischen Computer betrieben werden soll. Für beide Lager (herkömmliche Textverarbeitungsprogramme und $\text{T}_{\text{E}}\text{X}$) gibt es nämlich auch reine Online-Anwendungen, bei denen der Nutzer allein über den Browser auf seine Dokumente zugreift, und eine lokale Installation inklusive aller eventuellen Hardware-Fragen entfällt.

4.1 Speicherplatz

Diese Kategorie ist bei heutigen Festplattengrößen eigentlich kein Thema mehr. Dennoch eine kurze Ausführung:

MS Office belegt in seiner aktuellen Version nicht unter 1 GB Speicherplatz; andere Textverarbeitungen wie LibreOffice sind da genügsamer. Beiden ist gemein, dass sie sich modular installieren lassen, d. h. beim Installieren kann man die Komponenten der jeweiligen Office-Suite auswählen. Auch von einer aktuellen $\text{T}_{\text{E}}\text{X}$ -Distribution wie TeXLive wird wenigstens ein Gigabyte Speicherplatz belegt, je nachdem, wie umfangreich sie installiert wird.

²Nicht nur aus diesem Grund wird in immer mehr Behörden und Schulen (weltweit) MS Office durch LibreOffice ersetzt.

Hier unterscheidet sich eine T_EX-Distribution von einem herkömmlichen Office-Programm, denn sie lässt sich wesentlich modularer installieren; genau genommen kann man sich ein System mit nur denjenigen Komponenten zusammenstellen, die man wirklich benötigt. T_EX lässt sich also besser an die Bedürfnisse des Nutzers anpassen, und belegt damit nicht unnötig Speicherplatz (insbesondere auf älteren Systemen!).

Bei T_EX kommt weiterhin ein Texteditor hinzu. Da es sich bei T_EX-Quellcode um reine Textdateien handelt, kommt jeder beliebige Texteditor infrage, sodass dem Anwender diejenige Freiheit gegeben wird, eine Arbeitsumgebung seiner Wahl zu nutzen. Professionelle Editoren oder gar ein reiner T_EX-Editor haben den Vorzug, dass sie die Dokumentstruktur und die T_EX-Syntax anzeigen und andere T_EX-bezogene Schalter zur Verfügung stellen können. In vielen T_EX-Editoren ist eine Vorschau-Ansicht generiert, die das kompilierte Dokument synchron anzeigt. Für einen Editor fallen selten mehr als 50 MB Speicherplatz an.

4.2 Prozessor, Speicher, Grafik

Jede herkömmliche Textverarbeitung wie LO Writer oder MS Word hat den Nachteil, dass sie eine grafische Oberfläche darstellt, und nicht, wie T_EX, auch im reinen Textmodus, z. B. im Terminal-Fenster, betrieben werden kann. Dementsprechend benötigen Erstgenannte einen modernen Prozessor, nicht zu wenig RAM und manchmal auch etwas Unterstützung von der Grafikkarte, wenn alles flüssig laufen soll. Das trifft besonders auf MS Office zu, das von Version zu Version optisch aufgeblasener zu werden scheint (Transparenz, Schatten, Animationen). Je mehr dieser optischen Spielereien verbaut sind, desto mehr werden Systemressourcen verbraucht und desto träger wird die Eingabe oder Verarbeitung von Funktionen sein.

Da, konzeptbedingt (Abschnitt 5), Wortumbrüche und Layout bei herkömmlichen Textverarbeitungen während der Eingabe berechnet werden³ (und nicht erst, wie bei T_EX, beim Kompilieren), wird ebenfalls vermehrt Prozessorleistung beansprucht als bei T_EX.

T_EX läuft auch auf uralten PCs zuverlässig, notfalls nur im reinen Textmodus. Weder zum Eingeben von Text noch zum Kompilieren ist eine grafische Oberfläche notwen-

³»Live-Text«

dig. Selbst bei umfangreichen Dokumenten würde der Quellcode sauber, wenn auch länger kompiliert werden.

4.3 Betriebssystem

Bekannt ist, dass die Abwärtskompatibilität bei MS Word keine Priorität zu haben scheint, weder was das Dateiformat (Abschnitt 4.4) betrifft noch die Anwendung selbst. Ein modernes MS Office auf einem Windows 98 zu betreiben, ist unmöglich. Außerdem ist MS Office nicht für GNU/Linux verfügbar (kann aber emuliert werden). LibreOffice ist da einsichtiger und lässt sich auf nahezu jedem System mit grafischer Oberfläche installieren, ob nun Windows (ab XP), klassischem Desktop-Linux, Mac, Android, Solaris, FreeBSD oder sonst was. T_EX hängt ebenfalls weniger vom Betriebssystem ab (Windows, Apple, GNU-Linux, OS/2, MS-DOS, OpenVMS, BeOS, Amiga u. a.

4.4 Dateiformat

Das von MS Word verwendete Format `.doc` (und neuerdings sein XML-Luftballon `.docx`) gelten weithin als Standard-Dateiformat zum Austausch von Textdokumenten. Nur sind sie kein Standard und sollten niemals einer sein! Bekannt ist, dass Word-Formate gar nicht gut auf Abwärtskompatibilität zu sprechen sind; dass sich unter Word 95 erstellte Dateien mit modernen Word-Versionen mitunter gar nicht mehr öffnen lassen. Das Word-Format ist kein Format für Dokumente, die auch in ferner Zukunft noch lesbar sein sollen!

Ein Standard ist dagegen das OpenDocument-Format, bezüglich der Textdokumente heißt es `.odt`: Es handelt sich um ein XML-basiertes und mit ZIP komprimiertes, offen dokumentiertes Format, das von einem internationalen Gremium (und nicht einem einzelnen Konzern) betreut wird. Es gilt mittlerweile als (berechtigter) Standard für die An- und Ablage von Textdateien. Im Gegensatz zum `.docx`-Format belegt es sogar geringfügig weniger Speicherplatz, da im `.docx` (das ebenfalls auf XML basiert) die XML-Syntax zwar technisch korrekt geschrieben, aber jedes einzelne Wort mit XML-Tags umklammert wird, anstatt, wie bei `.odt`, nur die Absätze. Das bläht natürlich die Dateigröße auf.

Das Dateiformat von $\text{T}_{\text{E}}\text{X}$, also eine reine Textdatei mit der Endung `.tex`, ist das in diesem Vergleich vermutlich einfachste Dateiformat. Es kann mit jedem beliebigen Texteditor geöffnet und bearbeitet werden⁴. Dabei ist zu bedenken, dass reine Textformate mitunter die ältesten Austauschformate für digitale Texte sind, d. h. ein in den 80er Jahren erstellter `.tex`-Code lässt sich (mit wenigen Anpassungen, ggf. einer korrigierten Zeichencodierung) auch heute noch lesen, und er wird auch noch in Jahrzehnten lesbar sein. Hinzu kommen zwei weitere Vorzüge reiner Textdateien:

- Da in `.tex` nur der reine Inhalt gespeichert wird, und nicht etwa Software-spezifische Informationen, bleibt die Größe der Datei, selbst unkomprimiert, sehr klein. Bei Word-Dateien werden dagegen ggf. auch Vorgängerversionen des gleichen Dokuments gespeichert, Hardware-spezifische Infos im Header u. a.
- Bei reinen Textdateien kann der Editor gelegentlich abstürzen, ohne dass gleich der Inhalt des gesamten Dokuments beschädigt werden muss. Sollte etwas bei komplexen Formaten wie `.odt` oder `.docx` beschädigt werden, kann die Datei ggf. nicht mehr eingelesen werden (bei `.odt` sind die Chancen durch Datenrettung aber erheblich größer).

4.5 Systemschriftarten

Prinzipiell können sowohl herkömmliche Textverarbeitungen als auch $\text{T}_{\text{E}}\text{X}$ die auf dem System installierten Schriftarten verwenden. $\text{T}_{\text{E}}\text{X}$ selbst bringt Dutzende weitere Schriftarten mit, die über entsprechende Pakete eingebunden werden. Externe Schriftarten lassen sich mithilfe von $\text{X}_{\text{Y}}\text{T}_{\text{E}}\text{X}$ oder $\text{LuaT}_{\text{E}}\text{X}$ verwenden.

4.6 Konfigurationsdateien

Die das Dokument betreffenden Einstellungen werden im Quellcode selbst hinterlegt; die Konfiguration des Texteditors wird oftmals in einer einfachen Text-Datei gespeichert. Große Textverarbeitungen wie MS Word oder LO Writer verteilen ihre Programm-internen Einstellungen meist etwas unübersichtlich auf der Festplatte.

⁴OK, das geht auch mit dem im `.odt` eingebetteten XML, wenn man es findet.

Soll für eine Anzahl von Dokumenten dieselbe Konfiguration zur Verfügung stehen, kann die $\text{T}_{\text{E}}\text{X}$ -Präambel in eine separate Datei ausgelagert und in allen Dokumenten verlinkt werden. So modifiziert man sie nur noch an einer Stelle.

4.7 Die Sache mit den Bugs

Komplexe, Software wie LibreOffice oder MS Office enthalten natürlich immer wieder Bugs, nicht umsonst gibt es Updates und ferner: mit jeder neuen Version ebenfalls Bugs und Updates. Dieses ewige Hin und Her kennt jede Software. $\text{T}_{\text{E}}\text{X}$ dagegen hat den Ruf, dass Bugs nur selten gefunden werden. Natürlich besteht eine $\text{T}_{\text{E}}\text{X}$ -Distribution aus vielen Paketen, die, selbstverständlich, auch hin und wieder von Bugs befreit oder mit neuen Funktionen ausgestattet werden. Sie entwickeln sich weiter oder veralten und sind dann nicht mehr mit anderen, aktuellen Paketen problemlos nutzbar. Wieder andere Pakete werden seit Jahren nicht mehr weiterentwickelt, ganz einfach, weil sie ausgereift sind und keine Bugs bekannt sind. Anwender, die einer Flut von Warnungen und Fehlern nach dem Kompilieren ihrer Dokumente gegenüberstehen, neigen nur zu oft dazu, auf Programmierfehler zu schließen. Üblicherweise handelt es sich aber um unachtsam gesetzten Code, d. h. Syntax-Fehler.

5 Layout und Typographisches

Wie in Abschnitt 2 bereits angedeutet, kann man mit $\text{T}_{\text{E}}\text{X}$ ohne viel Eingreifen einen Text erzeugen, der absolut harmonisch und gut lesbar ist. Das hat nicht nur damit zu tun, dass $\text{T}_{\text{E}}\text{X}$ die meisten Layout-Fragen selbst übernimmt (Satzspiegel-Größe etc.), sondern auch der Umbruch-Algorithmus anders funktioniert als bei einer Textverarbeitung wie MS Word oder LO Writer: Hier werden die Wörter nach Sprachvorgaben umgebrochen, die Zusammenstellung der Wörter und ihr Umbruch erfolgt meist auf Zeilenbasis, d. h. sobald ein Wort nicht mehr in die Zeile passt, wird es umgebrochen oder auf die nächste Zeile verwiesen. Beim resultierenden Blocksatz können dann sehr unterschiedliche Ergebnisse herauskommen: Lücken reißende Wortabstände oder wenigstens ein unausgeglichener Grauwert der Seite sind keine Seltenheit. Bei $\text{T}_{\text{E}}\text{X}$ dagegen geschieht der Wort-Umbruch auf Absatzbasis (was auch nicht weiter verwundert, denn schließlich schreibt man einen Absatz erst zu Ende, ehe man ihn

setzt): Während des Kompilierens wird der Absatz »analysiert« und der Absatz so gesetzt (und umgebrochen), dass alle Wortabstände gleichmäßig (über den gesamten Absatz) verteilt sind. Daraus resultiert ein einheitlicher Grauwert und der Effekt, dass Testpersonen denselben Text schöner und harmonischer finden, wenn er mit $\text{T}_{\text{E}}\text{X}$ gesetzt worden ist.

Wer ein paar optische Vergleiche zwischen Word und $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ vorzieht, kann sich beispielsweise **hier** und **hier** umsehen. Dort gibt's Beispiele für Ligaturen, Kerning, Kapitälchen und mehr, jeweils im Vergleich zu Word.

Als zusätzliche Spielerei lässt sich auf einen sog. optischen Randausgleich zurückgreifen, bei dem Zeichen mit viel Fleisch (V, W) oder Bindestriche, sofern sie am Zeilenende stehen, ein wenig nach außerhalb vom Satzblocks verschoben werden. So werden auch hier »lückige Stellen« aufgefüllt, was den Grauwert ebenfalls erhöht. Sowohl in $\text{T}_{\text{E}}\text{X}$ als auch mit einer herkömmlichen Textverarbeitung sind diese Spielereien möglich. In einer herkömmlichen Textverarbeitung muss allerdings eine Schriftart verwendet werden, die den optischen Randausgleich unterstützt, z. B. die Libertine (Graphite-Version). Typisch sind solche Möglichkeiten für DTP-Programme, etwa das hz-Programm, das heute in Adobe InDesign integriert ist.

Ein bedeutender Unterschied findet sich, wenn man die Stabilität des Layouts betrachtet: Ein mit $\text{T}_{\text{E}}\text{X}$ -Quellcode geschriebenes Dokument wird, unabhängig vom Betriebssystem, der Hardware oder dem Drucker, zu jeder Zeit genau gleich aussehen. Das lässt sich von herkömmlicher Textverarbeitung nicht zwingend behaupten: Selbst das offene, gut dokumentierte OpenDocument-Format kann ein unterschiedliches Ergebnis liefern, je nachdem ob man dieselbe Datei mit LibreOffice, MS Office oder Calligra öffnet. Prinzipiell sind OpenDocument-Dateien »Layout-stabiler« als Word-Dateien. Bei MS Word kommt dazu, dass das Layout des Dokuments vom installierten Druckertreiber abhängig sein kann, d. h. das gedruckte Dokument sieht unterschiedlich aus, je nachdem, auf welchem Drucker es gedruckt wurde. Durch Kompatibilitätsprobleme kann es beim Öffnen mit anderen Textverarbeitung zum Verschieben von Bildern und Bildunterschriften oder einem anderen Wortumbruch kommen.

5.1 Ligaturen, Alternativ-Buchstaben, Ziffern

Durch Ligaturen an entsprechender Stelle erhöht sich die Lesbarkeit eines Textes. Sofern Ligaturen in der betreffenden Schriftart enthalten sind, werden Standard-Ligaturen wie `fl` und `fi` von $\text{T}_{\text{E}}\text{X}$ und herkömmlichen Textverarbeitungen (bei MS Word auch nur manchmal, siehe Times New Roman) automatisch gesetzt. Sofern über OpenType-Features noch weitere Ligaturen-Paare, alternative Buchstabenformen, kontextsensitive Ersetzungen oder sogar Ziffernsätze (Versalziffern vs. Mediävalziffern, Tabellenziffern vs. Proportionalziffern) enthalten sind, müssen sie erst separat aktiviert werden. Unter $\text{T}_{\text{E}}\text{X}$ wird das mit $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ und $\text{LuaT}_{\text{E}}\text{X}$ ermöglicht; bei MS Word findet sich ab Version 2013 eine entsprechende Funktionserweiterung im Zeichen-Dialog. Unter LibreOffice können derartige Features aktiviert werden ([Anleitung](#)), indem man die entsprechenden OpenType-Kürzel an die Schriftart setzt.

Für LO Writer gibt es eine Erweiterung namens `Typography toolbar`, welche eine Werkzeugleiste einblendet, über die sämtliche OpenType-Funktionalitäten einer Schriftart verfügbar werden⁵: Aktivieren/Deaktivieren von Ligaturen jeder Art, Proportional- und Tabellenziffern, Minuskel- und Majuskelziffern, kontextsensitive Ersetzungen, durchgestrichene Null, korrektes Minuszeichen und viele andere Dinge. So können ebenfalls für einen Text nachträglich alle Ligaturen zu- und abgeschaltet werden; das ersetzt natürlich nicht die manuelle Überprüfung, ob eine Ligatur an der betreffenden Stelle auch erlaubt ist!

5.2 Kapitälchen und Unterstreichungen

Je nach Schriftart sind im Glyphensatz echte Kapitälchen enthalten. (Kapitälchen werden häufig und typografisch falsch durch Verkleinerung von Versalbuchstaben generiert: herkömmliche Textverarbeitungen stellen diese Möglichkeit durch die Kapitälchen-Formatierung bereit, ohne darauf hinzuweisen, dass die verwendete Schriftart vielleicht gar keine Kapitälchen enthält! Werden Kapitälchen von $\text{T}_{\text{E}}\text{X}$ angefordert, die in der Schriftart nicht enthalten sind, wird dies beim Kompilieren mit einer Warnung bemerkt.)

⁵Diese LO-Erweiterung arbeitet nur mit der Graphite-Version einer Schriftart zusammen, z. B. Libertine!

Was Unterstreichungen angeht, werden diese bei herkömmlichen Textverarbeitungen wie LO Writer durch die Unterlängen der Buchstaben gezogen. Obwohl Unterstreichungen für sich allein typografisch fragwürdig sind, ist die Durchschneidung der Unterlängen typografisch doppelt falsch. Je nach Paket werden Unterstreichungen bei T_EX dagegen unterhalb der Unterlängen der Buchstaben gesetzt.

5.3 Rechtschreibprüfung

Die »Rechtschreibprüfung« (Orthografie) entspricht sowohl bei herkömmlichen Textverarbeitungen als auch bei T_EX-Editoren (sofern sie diese Funktionalität enthalten) einfach einem Abgleich der Wörter mit einer *Whitelist*. Ist das betreffende Wort nicht enthalten, wird es rot unterkringelt oder kann hinzugefügt werden. Der Umfang dieser *Whitelist* ist wenig von der Software abhängig; im Internet können derartige Wörterbücher (*dictionaries*, Datei-Endung `.dic`) gedownloadet und in den Editor oder die Textverarbeitung einbezogen werden.

Was die Grammatik-Prüfung angeht, kenne ich keinen T_EX-Editor, der das kann. MS Word und auch LO (mit Plugin) können zwar die Wortstellung und -auswahl auf grammatikalische Richtigkeit prüfen, aber auf eventuelle Beanstandungen ist kritisch einzugehen.

5.4 Formelsatz und Sonderzeichen

T_EX ist bekannt für seinen exzellenten Formelsatz beliebig umfangreicher und komplexer Formeln; spezielle auf unterschiedliche Formelebenen abgestimmte Glyphengrößen helfen beim Setzen einer insgesamt harmonisch aussehenden Formel-Struktur. MS Word und LO Writer haben zwar auch eigenständige Formelsatz-Module, die aber v. a. bei großen und komplexen Formeln schnell an die Grenzen der ästhetischen Darstellung stoßen. Meistens werden für Hoch- und Tiefstellungen aller Art einfach die Glyphen des derzeitigen Font skaliert, ohne dass sie auf gute Lesbarkeit in kleinen Größen optimiert sind; außerdem werden Zeichen oftmals enger zusammengepresst als sie sollten. Da ich normalerweise keine komplexeren Formeln setze (außer durch Bruchstrich getrennte Terme), kann ich die Leistungsfähigkeit nur ungenau beurteilen. Viele Anwender kritisieren auch eine auf amerikanische Maßstäbe getrimmte Formeldarstellung, die nicht immer allen Ansprüchen genügt (vielleicht durch Er-

weiterungen zu beheben?). Nicht unerwähnt soll sein, dass hochkomplexe Formeln im $\text{T}_{\text{E}}\text{X}$ -Quellcode zu einem Gewirr aus Buchstaben und Zahlen ausarten können, in denen schnell die Übersicht verloren geht. Spezielle Formelmodule wie das von LibreOffice zeigen dagegen direkt das Endergebnis (oder wahlweise den dahinterliegenden Code, der dann $\text{T}_{\text{E}}\text{X}$ ähnelt), sodass man stets die Übersicht behält und auf Veränderungen leichter einwirken kann.

Was die Symbolvielfalt angeht, sind $\text{T}_{\text{E}}\text{X}$ und herkömmliche Textverarbeitungen vergleichbar. Beide greifen zurück auf einen Fundus von speziellen Symbol-Schriftarten (oder, im Falle von $\text{T}_{\text{E}}\text{X}$, auf entsprechende Symbol-Pakete), sodass sich jedes nur erdenkliche Zeichen darstellen lassen kann.

$\text{T}_{\text{E}}\text{X}$ ist darüber hinaus in der Lage, mithilfe von Paketen musikalische Noten oder chemische Strukturformeln (mit optimalen Abständen) zu setzen; etwas, das ich von keiner herkömmlichen Textverarbeitung kenne.

5.5 Zurichtung (Kerning)

Viele professionelle Schriftarten sind mit Tausenden Kerning-Paaren ausgestattet, die den Textfluss bestimmter Buchstaben-Paare optimieren (z. B. die Unterschneidung der Buchstabenpaare T-a und V-o). In den meisten Textverarbeitungsprogrammen (LO, MS Office älter als 2010) sind nur ein Teil dieser Kerning-Tabellen ansprechbar. Mit der Nutzung von $\text{X}_{\text{Y}}\text{T}_{\text{E}}\text{X}$ oder $\text{LuaT}_{\text{E}}\text{X}$ kann dagegen der gesamte Umfang der OpenType-Schrift, darunter die vollständigen Kerning-Tabellen, ausgereizt werden. Eigene Tests (Kerning-Vergleiche) zeigen aber, dass der Satz derselben Paare zuweilen besser in $\text{T}_{\text{E}}\text{X}$ als LO Writer oder MS Word aussieht.

5.6 Umfang der Dokumente

Beispiele für umfangreiche Dokumente gibt es viele: Handbücher, Abschlussarbeiten, Bücher, Zeitschriften mit separaten Beiträgen etc. Hier ist sinnvoll, das Dokument in einzelne Kapitel aufzutrennen, was unter herkömmlichen Textverarbeitungen gerne einfach durch die Anlage einzelner Textdokumente geschieht. Das Endergebnis wird dann in einer gemeinsamen PDF zusammengeführt. Handelt es sich aber um Abschnitte, die durch Querverweise, gemeinsame Literatur- und Indexlisten oder ein Inhaltsverzeichnis aufeinander aufbauen und voneinander abhängen, muss sich

der Anwender um deren Konsistenz und Korrektheit selbst bemühen. Gleiches gilt für eine evtl. durchgängige Nummerierung von Bildunterschriften, Seitenzahlen, Überschriften-Nummern usw. Mit LibreOffice kann man dieser Problematik mit dem wenig bekannten »Globaldokument« (Dateiendung `.odm`) entgegnen, das als Header für beliebig viele Unterdateien fungiert, die z. B. als einzelne `.odt`-Dateien (Kapitel) in einem Ordner liegen. Jede der Dateien lässt sich einzeln bearbeiten, aber im Globaldokument sind alle zusammen sichtbar, mit einheitlicher Nummerierung, Seitenzahl, gemeinsamen Inhaltsverzeichnis usw.

Mit $\text{T}_{\text{E}}\text{X}$ können auf dieselbe Weise Dokumente untereinander verkettet werden (Kommando *input*); sie werden beim Kompilieren zusammengefügt, Seitenzahlen, Abbildungsnummerierungen etc. korrekt gezählt und gesetzt. Die $\text{T}_{\text{E}}\text{X}$ -internen Zähler und Automatismen arbeiten dabei so effizient, dass man im Endergebnis sicher sein kann, dass alle Querverweise, Literaturzitate und sonstiges, das dynamische Felder gebraucht, fehlerfrei sind. Und bei Unstimmigkeiten würde man während des Kompilierens darauf hingewiesen.

5.7 Verlinkte Abbildungen

Einfacherweise werden bei illustrierten Textdokumenten die Bilder in die Datei eingebettet. Dadurch können die Dateien rasch eine Dateigröße annehmen, bei denen der Programmabsturz vorprogrammiert ist. Zumindest dürfte das Arbeiten mit dem Dokument immer träger reagieren. Hinzu kommt die Dauer für das Öffnen und Speichern der Datei. Sinnvoller ist hier das Verlinken der Bilder.

In $\text{T}_{\text{E}}\text{X}$ -Quellcode können, konzeptbedingt, Bilder nur verlinkt werden, denn es handelt sich um reine Textdateien, die keine Bilder aufnehmen können.

In beiden Fällen ist die Verlinkung eines Bildes sinnvoll, denn sie bewahrt die volle Kontrolle über die Bild-Dateien: Obwohl auch innerhalb der Textverarbeitungen wie MS Word oder LO Writer eine nachträgliche Korrektur (Farbstich, Kontrast, Zuschnitt etc.) möglich ist, sollten sie extern verwaltet und bearbeitet werden. Da sie verlinkt sind, werden sie im (schlankeren) Dokument aktualisiert, sobald sie verändert wurden.

6 Was \TeX kann, und Textverarbeitungen nicht

- Möglichkeit, auch mit transparenten und übereinander angeordneten Schriftblöcken zu arbeiten.
- Manuelle Trennstellen (weiche Trenner) für falsch umgebrochene Wörter lassen sich wesentlich einfacher setzen. Auch kann eine eine Dokument-weit geltende Wortliste für häufig gebrauchte, problematisch umgebrochene Wörter angelegt werden.
- Effizienter und verlässlicher Automatismus zum Erzeugen von Inhaltsverzeichnissen, Literaturverzeichnissen, bibliografischen Zitaten, Nummerierung von Bildunterschriften, Indizes, Kopfzeilen etc. Derartigen Automatismen in einer herkömmlichen Textverarbeitung zu vertrauen, setzt eine absolut einwandfreie Anwendung von Absatzvorlagen voraus. Ebenso verlässlich ist die Änderung von Absatzvorlagen mit \TeX .
- Ansehnliche Vorlagen für Kapitelköpfe.
- Die Eingabe von halben Leerzeichen, Geviertstrichen und anderen typografischen Kleinigkeiten erfordert nicht den Wechsel in die Glyphentabelle (und die Suche nach dem Zeichen), sondern ist direkt im Quellcode möglich, und zwar mit allen auf der Tastatur sichtbaren Zeichen.
- Programmierbarkeit nach eigenen Bedürfnissen, z. B.:
 - Einrichtung einer Dokumentklasse, speziell für Geschäftsbriefe: Festlegung der Position von Brief-Inhalt, Anhangsliste, Absender-Feld, Firmen-Logo, Falz- und Lochermarken. Einiges davon geht über die vertrauten Dokument-Vorlagen hinaus, die man von MS Word und LO Writer kennt.
 - Definition von Umgebungen: ein Bereich, in dem der gesamte Text grün und kursiv ist, sowie eine andere Schriftart als der Brottext annimmt. Die selbst benannte Umgebung wird im Quellcode, wie üblich, mit `begin` und `end` eingeklammert.
- Kontrollgewinn/Transparenz des Dokuments: durch das reine Textformat sehe ich stets, warum ein bestimmter Text diese oder jene Formatierung angenom-

men hat. Eine herkömmliche Textverarbeitung zeigt mir in der Tat nur das Ergebnis und verbirgt im Unsichtbaren, im Hintergrund, alle Informationen, die zu diesem Ergebnis geführt haben. Es ist also möglich, dass in dem Dokument Informationen stehen, die ich am Bildschirm gar nicht sehen kann. \TeX -Quellcode zeigt mir dagegen immer an, was wirklich in der Datei enthalten ist: Ich weiß genau, mit welchem Zeichen die Datei beginnt, und mit welchem sie endet. Das gibt mir auch die Kontrolle, dass die Datei bei Weitergabe nur das enthält, was sie enthalten soll.

- Vermeidung typografischer Fehler, z. B.:
 - Im Quellcode doppelt (oder beliebig mehrfach!) eingegebene Leerzeichen werden nach dem Kompilieren auf ein einziges reduziert. Im fertigen PDF stößt man also nie auf zwei Wörter, die aus Versehen durch zwei statt einem Leerzeichen spationiert worden sind (außer man forciert das mittels Befehlen). In LibreOffice kann man in den Autokorrektur-Optionen auch einstellen, dass doppelt eingegebene Leerzeichen ignoriert werden. Der bereits erwähnte Pepito Cleaner (als Erweiterung für LibreOffice) kann das Dokument am Ende auch nach doppelten Leerzeichen durchsuchen, oder man tut das mit der Suchen/Ersetzen-Funktion der Textverarbeitung.
 - Die Nutzung des `csquotes`-Pakets vorausgesetzt, werden Anführungszeichen nicht mehr manuell eingegeben, sondern das Wort wird mit einem Befehl eingeklammert. Nur am Dokumentanfang wird definiert, welche Form von Anführungszeichen verwendet werden sollen (deutsch/englisch/französisch, einfach/doppelt etc.). Beim Kompilieren werden diese nach Vorgabe gesetzt, und können ebenso einfach, Dokument-weit!, ausgetauscht werden. Durch die Einklammerung ist außerdem ausgeschlossen, dass man aus Versehen nur das eine Anführungszeichen setzt und das zweite vergisst.
 - Literaturverwaltung zusammen mit $\text{Bib}\text{\TeX}$ bzw. $\text{Bib}\text{\LaTeX}$: Literatur in einer Datenbank, eindeutige Schlüssel werden im Quellcode zitiert; verwaiste Zitate im Mengentext oder überflüssige Einträge im Literaturverzeichnis sind gar nicht möglich! Korrekte Sortierung der Einträge und

einheitliche Formatierung werden durch Automatismen und einen ausgewählten, eindeutigen Zitierstil übernommen.

- $\text{T}_{\text{E}}\text{X}$ stürzt so gut wie niemals ab, egal wie groß das Dokument wird oder wie alt das System ist.
- Mit $\text{T}_{\text{E}}\text{X}$ erzeugter Text sieht immer identisch aus, egal auf welchem System man ihn kompiliert.
- Der Editor zum Bearbeiten des $\text{T}_{\text{E}}\text{X}$ -Codes ist nach Vorlieben, Hardware und Gewöhnung frei wählbar. Dazu zählt auch, dass ich mir ein (augenschonendes) dunkles Farbthema einrichten kann, während ich bei herkömmlichen Textverarbeitungen auf vorinstallierte Designs angewiesen bin. Sie lassen bestenfalls eine Auswahl von Farbschattierungen der Buttons oder den Austausch der Symbole zu. Gleichwohl bleibt mir das weiße, grelle Blatt als Eingabefeld erhalten.
- Die verwendeten Kommandos zur Formatierung eines Textes sind seit Jahrzehnten weitgehend die gleichen. Daran werden auch neuere Versionen eines Editors nichts ändern. Hin und wieder bringt eine Paket-Aktualisierung auch einige neue Befehle mit sich und markiert andere als obsolet. Das ist aber in der Dokumentation ausführlich beschrieben und kann mit Alternativen entsprechend ausgeglichen werden.
- Man kann auch ungewöhnliche Strukturen setzen, z. B. Musiknoten, Sternenkarten oder Schachbrett-Konstellationen.
- $\text{T}_{\text{E}}\text{X}$ -Code kann kommentiert werden, d. h. das Kommentar-Zeichen für diese Sprache (das Prozent-Zeichen) leitet Text ein, der nur im Editor sichtbar ist, nicht aber im kompilierten Ergebnis erscheint. Das eröffnet die Möglichkeit durch Hinterlegung von Bemerkungen (beliebiger Länge), wie bestimmte Befehle funktionieren oder gedacht sind; oder man »versioniert« seinen Text, indem man veraltete Passagen auskommentiert und so unsichtbar erhält. Oder man nutzt den Kommentar zum Ein-/Ausschalten alternativer Präambel-Befehle für Testzwecke. In einer herkömmlichen Textverarbeitung kann man natürlich auch Kommentare anbringen, die dann als schwebende Farbboxen am Seitenrand erscheinen (sie erhalten sogar automatisch Erstellungszeit und Namen

des Erzeugers). Nachteilig bei diesen Boxen ist, dass sie nur begrenzt Text aufnehmen können.

7 Was Textverarbeitungen können, und $\text{T}_{\text{E}}\text{X}$ nicht

- Soll das Textdokument (nicht als PDF) weitergegeben werden, ist die Arbeit mit einer herkömmlichen Textverarbeitung einfacher: Über den Speichern-Unterdialg kann man eines von vielen Dateiformaten auswählen. Das `.tex`-Format sollte man dagegen nur an Leute weitergeben, die damit auch arbeiten können. Für die Konvertierung eines $\text{T}_{\text{E}}\text{X}$ -Quellcodes zu einem Dateiformat wie `.docx` oder `.odt` existieren diverse Programme (z. B. der Parser Pandoc), denen gemein ist, dass das Ergebnis fast immer einer Nachbearbeitung bedarf. Manchmal ist der Zwischenschritt über HTML besser, das wiederum von MS Word oder LO Writer eingelesen wird. Für den umgekehrten Weg gibt es beispielsweise die LibreOffice-Erweiterung `Writer2LaTeX`, über die ein mit einer herkömmlichen Textverarbeitung erstelltes Dokument für $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ aufbereitet werden kann.
- Herkömmliche Textverarbeitung legen nur eine Datei pro Dokument an. $\text{T}_{\text{E}}\text{X}$ -Quellcode liegt zwar ebenfalls nur in einer einzigen Datei vor (sofern nicht aufgeteilt), aber beim Kompilieren entstehen daraus zahlreiche Metadateien, die Dokumentstruktur, Bibliografie etc. aufnehmen. Das kann den Arbeitsordner unordentlich werden lassen.
- Herkömmliche Textverarbeitungen bringen direkt eine Verschlüsselungsfunktion für ihre Dateien, mit denen die Berechtigung zum Lesen und/oder Schreiben geregelt werden kann. $\text{T}_{\text{E}}\text{X}$ -Quellcode ist zunächst nicht verschlüsselt, sondern muss nachträglich über ein Drittprogramm verschlüsselt werden (z. B. verschlüsseltes ZIP-Archiv).
- Wortzählung ist bei MS Word und LO Writer einfacher; hier genügt ein Klick auf den entsprechenden Button. Die Wort-Zählung von $\text{T}_{\text{E}}\text{X}$ -Quellcode ist nicht so trivial, denn Kommando-Befehle, auch die Präambel zählen nicht dazu. Verschiedene Editoren bieten eine Wortzählung an, die genau diese Befehlsstruktur »herausrechnet«, allerdings alle mit unterschiedlichem Ergebnis bei demsel-

ben Text. Bei Online-Editoren wird die Wortanzahl häufig in einer Statusleiste direkt angezeigt.

- Absatzvorlagen oder individuelle Listen-Umgebungen sind für TeX-Anfänger nicht leicht programmierbar; das geht in einer herkömmlichen Textverarbeitung einfacher.
- In MS Word/LO Writer ist die Anbindung einer Datenbank, z. B. für Serienbriefe, einfacher.
- Das Einfügen von Tabellen, deren Ausrichtung, Sortierung, Gliederung, Formatierung (auch Textrichtung) ist in einer herkömmlichen Textverarbeitung wesentlich einfacher als in TeX. Dort wird eine komplexe Tabelle rasch zum undurchsichtigen Gewirr aus Befehlen und Text, und die Orientierung, in welcher Zelle man gerade etwas verändert, ist kaum nachvollziehbar⁶. Gleichwohl können sich Tabellen aus einer herkömmlichen Textverarbeitung mit solchen aus einem TeX-Quellcode qualitativ und ästhetisch nicht messen.
- Obwohl über ein Zusatzpaket auch in TeX-Quellcode das beliebte »Änderungen nachverfolgen«, d. h. die farblich dokumentierte Veränderung des Textes durch Zweit- und Dritt-Autoren, möglich ist, sollte man das besser in einer herkömmlichen Textverarbeitung machen. Andererseits bieten manche Online-L^AT_EX-Editoren Funktionen zur Zusammenarbeit an: Dokumente könnten geteilt, d. h. gleichzeitig daran gearbeitet werden; Kommentarboxen und vollzogene Änderungen lassen sich für andere hinterlegen usf.
- Da man sich am fertigen Seitenlayout orientieren kann, ist das schnelle Korrigieren kleiner Textfehler einfacher. In Quellcode sucht man erfahrungsgemäß länger, um die korrigierende Stelle zu finden, insbesondere, wenn man sich an dem dazugehörigen PDF orientiert.

⁶Als Einstieg könnte man die Tabelle mit der Tabellenverarbeitung Gnumeric vorbereiten und dann als TeX-Quellcode exportieren.

8 Was soll ich nun nehmen?

8.1 Nimm \LaTeX ...

- wenn es um typografisch und ästhetisch einwandfreien Textsatz geht.
- wenn du komplexe Formeln setzen musst.
- wenn du dir keine Gedanken um die korrekte Nummerierung (Überschriften, Bilder, Seitenzahlen) und Zuordnung von Verweisen (Bibliografie-Zitate!) machen willst. Manch einer wird schon Stunden in den manuellen Abgleich gesteckt haben, ob Textzitate zu den Einträgen der Literaturliste passen!

8.2 Nimm eine herkömmliche Textverarbeitung...

- wenn dir die Arbeit mit Quellcode-Syntax zu abgehoben ist.
- wenn du dir keinen Überblick über die Funktionalität der einzelnen \TeX -Pakete verschaffen willst/kannst.
- wenn du sofort optisches Feedback für deine Eingaben brauchst, z. B. weiche Trennstellen.
- wenn du viel mit Schriftarten experimentieren willst.
- Wenn du komplexe Tabellen setzen musst, die nicht in einem Tabellendokument abgelegt werden können.
- wenn an deinem Dokument noch intensiv gearbeitet wird (wissenschaftliches Dokument mit vielen Co-Autoren), und man sich intensiv auf »Änderungen nachverfolgen«, Kommentarboxen und dergleichen verlässt. Obwohl gerade dann die konsequente Anwendung von Absatzformatierungen kompliziert ist, wenn ständig Absätze und Bilder eingefügt und verschoben werden, und auch das endgültige Layout noch gar nicht klar ist (oder bestimmt werden darf, siehe Weitergabe des Manuskripts an den Verlag).