Bachelorarbeit

Extraktion von Diagrammen aus Texten und Auswertung von Liniendiagrammen mit Deep-Learning Methoden

Luzian Uihlein

Würzburg, 10. August 2024

Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik VI
Betreuer: Prof. Dr. Frank Puppe
Norbert Fischer
Alexander Hartelt

Abstract

Hallo. Ich bin ein kleiner Blindtext. Und zwar schon so lange ich denken kann. Es war nicht leicht zu verstehen, was es bedeutet, ein blinder Text zu sein: Man ergibt keinen Sinn. Wirklich keinen Sinn. Man wird zusammenhangslos eingeschoben und rumgedreht – und oftmals gar nicht erst gelesen. Aber bin ich allein deshalb ein schlechterer Text als andere? Na gut, ich werde nie in den Bestsellerlisten stehen. Aber andere Texte schaffen das auch nicht. Und darum stört es mich nicht besonders blind zu sein. Und sollten Sie diese Zeilen noch immer lesen, so habe ich als kleiner Blindtext etwas geschafft, wovon all die richtigen und wichtigen Texte meist nur träumen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Kapitel 2

Literaturübersicht

Die automatische Transkription von Liniendiagrammen ist weit weniger erforscht als die von Tabellen, z.B. gibt es auf den ICDAR-Konferenzen (International Conference on Document Analysis and Recognition) keine Wettbewerbe (Challenges) mit annotierten Datensätzen, im Gegensatz zu Tabellen und vielen anderen Bereichen. Es gibt nur wenige Publikationen, die sich mit diesem Problem beschäftigen, wobei aktuelle Ansätze [3, 4] Deep-Learning-Techniken verwenden, die mangels annotierter realer Daten überwiegend mit synthetischen Daten trainiert werden. In der Literatur wird die Erkennung von Liniendiagrammen meist in folgende Schritte unterteilt:

1. Erkennen und Klassifizieren des Diagramms
2. Erkennen der x- und y-Achse des Liniendiagramms
3. Erkennen der Linien
4. Erkennen der Beschriftungen
5. Extraktion der Datenpunkte auf den Linien
6. Zuordnung der Datenpunkte zu den semantischen x- und y-Werten
7. Darstellung des Ergebnisses als Tabelle.

Während einfache Linien gut erkannt werden, wird bei überlappenden Linien oft angenommen, dass diese farbig gezeichnet werden, um sie zu unterscheiden. Dies gilt jedoch nicht für historische Liniendiagramme, die in der Regel durch verschiedene gestrichelte Linien unterschieden werden, was automatisch schwer zu erkennen ist. Dafür eignen sich semiautomatische Ansätze wie z.B. in [5] beschrieben. Hierbei werden die automatischen Schritte von den Anwendern sofort manuell überprüft und korrigiert, was bei einer Massentranskription nicht praktikabel, aber bei einer begrenzten Anzahl von Diagrammen realistisch ist, zumal eine Qualitätskontrolle für die GT-Erstellung ohnehin notwendig ist. Erforschte Herangehensweisen [6] zur Linienerkennung und Datenextraktion bestehen unter anderem aus der Erkennung von Schlüsselpunkten (key point detection) der jeweili-

gen Wertelinien, welche hier durch Steigungsänderungen (pivot points) festgelegt werden. Nach deren Erkennung durch ein neurales Netzwerk werden diese mit Hilfe einer zusätzlichen Faltungsschicht (convolution layer) zu einzelnen Linieninstanzen gruppiert. Andere Linieninstanzgruppierungsalgorithmen [7] bestehen in der Optimierung einer Kostenfunktion mithilfe der linearen Programmierung über ein Minimum-Kosten-Fluss-Problem (minimum-cost-flow problem). Im Vergleich zu handgeschriebenen, historischen Liniendiagrammen allerdings, bestehen die Datensätze exklusiv aus computergenerierten Textbeschriftungen, sodass die optische Schriftzeichenerkennung (optical character recognition) erfolgreicher durchgeführt werden kann. Die Zuordnung der Datenpunkte zu den semantischen x- und y-Werten erfolgt dadurch fehlerfreier, was wie bei allen Zwischenschritten die Effizienz des Endergebnisses direkt beeinflusst.
Zur Evaluation werden die Linien als kontinuierliches Ähnlichkeitsproblem (continuous similarity problem) behandelt. Die Punktsequenz der Vorhersage des Modells und eine definierte Grundwahrheitsmenge werden verglichen, sodass Präzision (precision), Erinnerung (recall) und F1-Wert (F1-Score) berechnet werden können.

Kapitel 3

Methodik

3.1 Extraktion von Diagrammen aus Texten

Ziel des ersten Teils ist die Extraktion der Diagrammen aus den historischen Textscans, welche dann im folgenden Teil in eine gewünschte Form ausgewertet werden können.
Die Wesentlichen Schritte des Extraktionsteils beinhalten die Objekterkennung, also die Bestimmung des Begrenzungsrechtecks (bounding box) der Diagrammen innerhalb den vorliegenden Vollseitscans und deren Unterscheidung in verschiedene Diagrammtypen, beispielsweise Linien- und Balkendiagrammen. Die erkannten Liniendiagramme werden anschließend anhand ihrer Auswertungsschwierigkeit klassifiziert, etwa durch Kennzeichnung deren Diagrammen, welche kontextbedingt gruppiert wurden, zum Beispiel aufgrund gemeinsamer Graphsachsen.
Um mit Hilfe von Deep-Learning Modelle zu trainieren, werden annotierte Grundwahrheiten (ground truth) benötigt.

3.1.1 Datensatz DocBank zur Objekterkennung

Für die Erkennung von Diagrammen in Texten wurden DocBank [1] und ein Anteil der historischen Wirtschaftsscans verwendet. DocBank besteht aus wissenschaftliche Publikation mit computergenerierten Grafiken zusammengesetzt, weshalb DocBanks Dokumentenseiten lediglich zum Vortrainieren des Detektionsmodells gedacht sind. Beabsichtigt wurde dieser Prozess des Vortrainierens um das System schneller und algemeingültiger, also mit besseren Voraussagen, trainieren zu können. Spätere Experimente untersuchen diese Annahme.
An die Vorkommenshäufigkeit bei den historischen Scans angepasst, wurde die Differenzierung in fünf Objektklassen beschlossen: Linien (line), Balken (bar), Histogramm (histogram), Sonstige (other) und Gemischt (mixture). Aufgrund von Verwechslungen des Modells im Verlauf der Experimente zwischen Balkendiagrammen und Histogrammen

wurden die Datensätze auf vier Klassen reduziert, indem Balkendiagramme und Histogramme vereinigt wurden.
Die Schwierigkeit zwischen Balkendiagrammen und Histogrammen zu unterscheiden beruht darauf, dass Balkendiagramme kategorische Datenvergleiche anschaulich machen, bei denen die Balkenanordnung irrelevant ist, während Histogramme kontinuierliche, numerische Daten darstellen. Die Differenz liegt lediglich an der Achsenbeschreibung und nicht an visuellen Hinweisen, oftmals werden Balkendiagramme jedoch mit Lücken zwischen den Balken dargestellt, während Histogramme lückenlos abgebildet werden; dies ist allerdings nicht ausschlaggebend zur Bestimmung des Diagrammtyps.
Für die manuell GT-Annotation der DocBank Dokumentenseiten, sowie folgender anderer Datensätze, wurde die Annotationssoftware CVAT [2] verwendet.

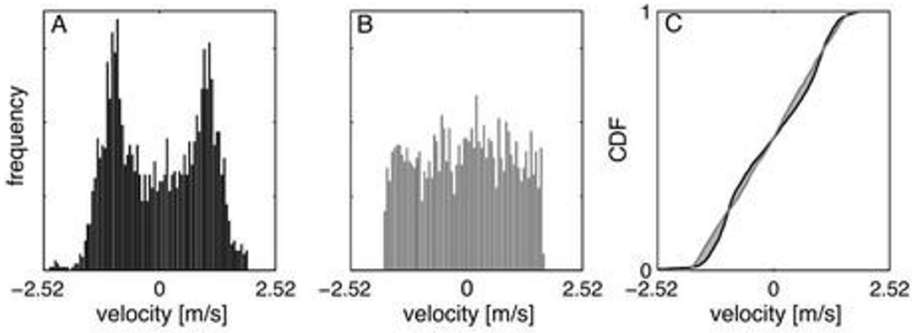

Figure 5: Panel A shows a velocity profile - histogram (h_A; black) of the

Abbildung 3.1: Beispiel kontextbedingter Gruppierung wegen gemeinsamer Y-Achsenbeschreibung eines gemischten Diagrammtyps (Histogramm und Liniendiagramm)

Da der Datensatz aus einer beträchtlich diversen Menge verschiedener wissenschaftlichen Publikationen besteht, beinhalten diese auch zahlreich verschiedene Diagrammlayouts. Um eine bestmögliche Konsitenz und Nützlichkeit in der Handannotation zu gewährleisten wurden einige Überlegungen gemacht: Da einige Abbildungen als Gruppe von Diagrammen fungieren, siehe Abbildung 3.1, muss die generelle Entscheidung getroffen werden, jedes Diagramm der Gruppe einzeln zu annotieren oder lediglich die gesamte Gruppe zusammen. Beide Möglichkeiten liefern Vor- und Nachteile; beim getrennten Annotieren muss die Gruppe in einem späteren Schritt nicht mehr in die einzelnen Diagramme aufgeteilt werden, jedoch können auch kontextbedingte Informationen verloren gehen, wie in dem abgebildeten Beispiel die Y-Achsenbeschreibung des mittleren Diagramms (B), welches sich eine gemeinsame Y-Achsenbeschriftung mit dem linken Diagram (A) teilt. Ebenfalls können Diagrammgruppen aus verschiedenen Diagrammtypen bestehen, etwa Histogramme und Liniendiagramme beieinander, weswegen dementsprechend für genau diesen Fall die gemischte Diagrammklasse eingeführt wurde. Bei weiteren Unklarheiten

des Gruppenumfangs wurde sich sonst immer an die darunterliegenden Abbildungsunterschrift gehalten.
Insgesamt wurden 321 Seiten annotiert, beinhaltend aus 105 Liniendiagrammen, 115 Balkendiagrammen (vereinigt mit Histogrammen), 79 sonstige und 66 gemischte Diagrammen.

3.1.2 Datensatz historischer Wirtschaftsscans zur Objekterkennung

Die Scans der geschichtlichen Wirtschaftsmagazine wurden mit ähnlichen Überlegungen annotiert. Hier befinden sich ebenfalls Diagrammgruppen, teils auch mit mehreren verschiedenen Diagrammtypen, siehe Abbildung 3.2, welche alle wieder als gesamte Gruppe annotiert wurden. Bis auf sehr wenigen Ausnahmen, befinden sich alle Abbildungen in den Scans visuell eingerahmt. Da die Ausrichtung derer jedoch nie wirklich perfekt gerade dargestellt wurde, und somit, der Ausrichtung verschuldet, kein Annotationsrechteck mit ausgeschlossenem Abbildungsrahmen gezeichnet werden kann wurde die Entscheidung getroffen, jede Annotation mit allen Ecken der Diagrammrahmen zu beinhalten. Grunsätzlich wurden alle Abbildungen, Diagramme oder nicht, wie etwa vereinzelte Karikaturen oder Landedskarten mit in die Klasse der sonstigen Diagramme eingeschlossen um so die allgemeine Erkennung von seltenen Diagrammtypen zu verstärken. Es wurden insgesamt 2391 zufällige Seiten ausgewählt und manuell annotiert, woraus sich 343 Liniendiagramme, 102 Balkendiagramme, 77 sonstige und 52 gemischte Diagramme ergebten.

Abbildung 3.2: Diagrammbeispiel historischer Scans

3.2 Schwierigkeitsklassifizierung von Liniendiagrammen

Aufgrund der überwiegenden Liniendiagrammen in den historischer Wirtschaftsscans, wurde sich im Folgenden primär auf die Auswertung der Liniendiagrammen fokusiert. Für genau diese Auswertung wurde der Vorverarbeitungsschritt überlegt, die extrahierten Liniendiagramme in verschiedene Untergruppen zu unterteilen. Es wurden vier Klassifikationen gewählt; Liniendiagramme mit nur einer Wertelinie, aus zusammengesetzten Diagrammen, also Liniendiagrammsgruppen, sich nicht überlappenden Wertelinien und sich überlappenden Wertelinien.

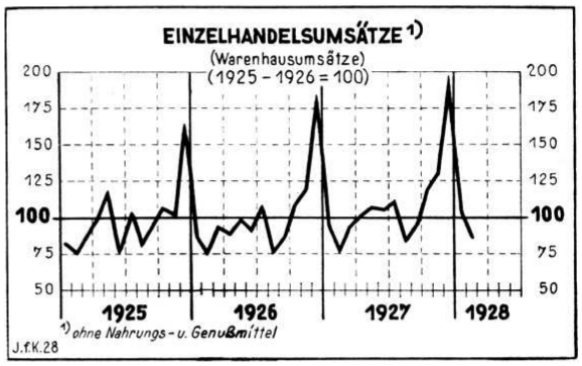

Abbildung 3.3: Liniendiagramm mit einer Wertelinie

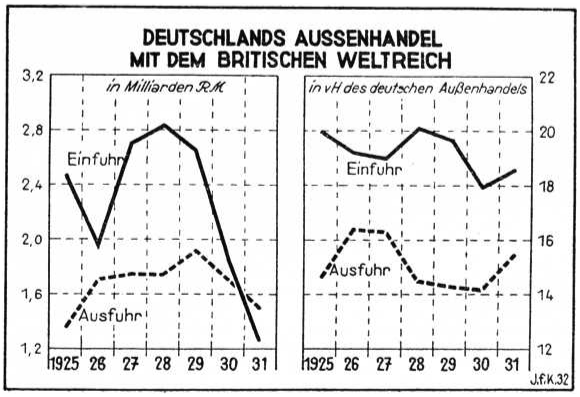

Abbildung 3.4: Zusammengesetzte Liniendiagrammsgruppe

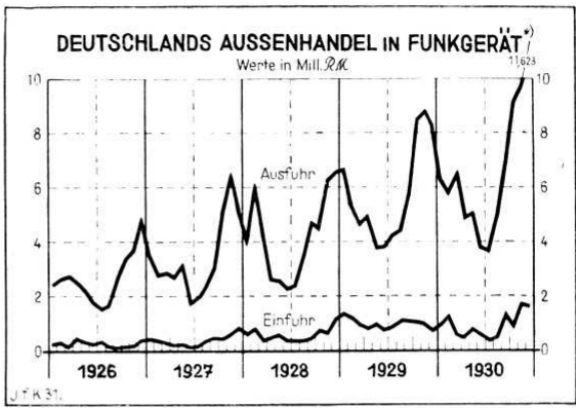

Abbildung 3.5: Liniendiagramm mit sich nicht überlappenden Wertelinie

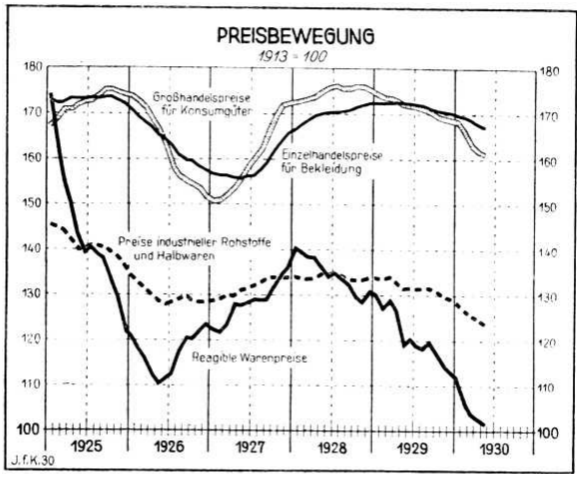

Abbildung 3.6: Liniendiagramm mit sich überlappenden Wertelinie

Es existieren nämlich Liniendiagrammsgruppen mit mehreren eigenständigen Unterdiagrammen, welche möglicherweise jeweils ihre eigene Achsenbeschreibung haben, oder auch sich kontextbedingt diese Achsenbeschriftungen teilen. Diese müssen also im Vergleich zu

einfachen Liniendiagrammen speziell behandelt werden. Aber auch wenn für Liniendiagramme mit nur einer oder sich nicht überschneidenden Wertelinien ein eher primitiver Extraktionsalgorithmus ausreichen würde, tritt bei komplexeren, sich überlappenden oder überschneidenden Wertelinien schnell das Problem der Linientrennung bzw. Liniengruppierung auf.

3.3 Auswertung von Liniendiagrammen

Für die Auswertung der historischen Liniendiagramme wurden Überlegungen gemacht, Beschriftungen und vorallem das Hintergrundgitter, welches sich in jedem Diagramm zu finden lässt, zu entfernen, jedoch wurde schnell klar, dass diese primitive Herangehensweise grundsätzlich eher impraktibel ist. Zum einen führen die nicht genau senkrecht und waagerecht verlaufenden Gitterlinien die korrekte Erkennung dieser zu einem nichttrivialem Erkennungsproblem und zum andern überlappen und verlaufen viele Wertelinien auf dem Gitter, sodass die einfache Entfernung der Gitterlinienpixel das Diagramm mit unzähligen Lücken verbleiben lässt. Dementsprechend wurde beschlossen, statt aus dem Diagramm alles bis auf die Wertelinien zu entfernen, die Wertelinien selbst zu extrahieren, also sie durch Segmentation vom Hintergrundgitter und allen anderen Elementen zu trennen.
Die manuelle Erstellung der Grundwahrheiten für die Werteliniensegmentation ist allerdings recht arbeitsaufwendig, weswegen zusätzlich ein synthetisch erstellter Datensatz generiert wurde, bei dem die Erstellung von Binärmasken der Wertelinien trivial ausfällt.

3.3.1 Datensatz von synthetischen Liniendiagrammen zur Segmentation

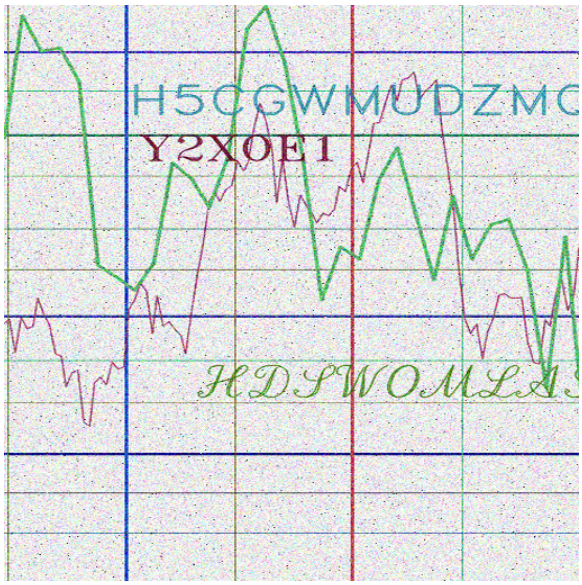

Abbildung 3.7: Synthetisch erstelltes Liniendiagramm

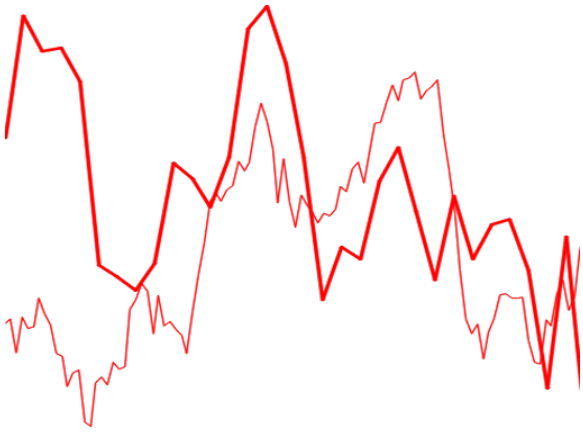

Abbildung 3.8: Zugehörige generierte Binärmaske der Wertelinien

Der synthetische Datensatz besteht aus 2000 verschiedenen, zufällig generierten Liniendiagrammen. Diese beinhalten zufällige Wertelinien und Gitterlinien, sowohl in Position als auch in Liniendicke und zufällige, teils den Wertelinien überlappenden, Textbeschriftun-

gen vielfältiger Schrifgrößen und Schriftarten. Nachbearbeitet wurden sie mit unterschiedlichem Bildrauschen, um so näher an die Scanqualität und Diversität der historischen Diagramme heranzukommen.

3.3.2 Datensatz von historischen Liniendiagrammen zur Segmentation

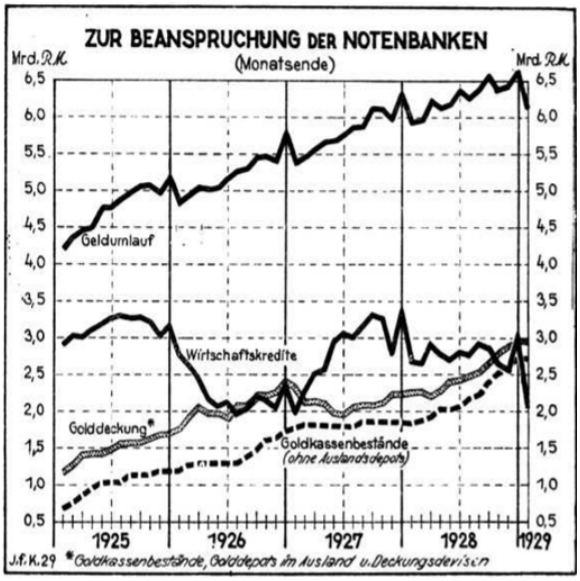

Abbildung 3.9: Historisches Liniendiagramm

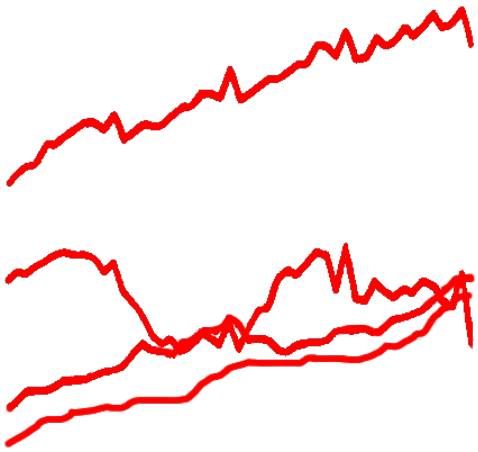

Abbildung 3.10: Zugehörige manuell erstellte Binärmaske der Wertelinien

Kapitel 4

Implementation

Kapitel 5

Experimente

Kapitel 6

Zusammenfassung

Declaration of originality

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Würzburg, 10. August 2024 Name Name .....

Literaturverzeichnis

[1] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020.

[2] CVAT.ai Corporation. Computer Vision Annotation Tool (CVAT), November 2023.

[3] Shivasankaran V P, Muhammad Yusuf Hassan, and Mayank Singh. Lineex: Data extraction from scientific line charts. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6202–6210, 2023.

[4] Jaewoong Lee, Wonseok Lee, and Jihan Kim. Matgd: Materials graph digitizer, 2023.

[5] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. Chartsense: Interactive data extraction from chart images. pages 6706–6717, 05 2017.

[6] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1916–1924, 2021.

[7] Mateusz Kozinski and Renaud Marlet. Image parsing with graph grammars and markov random fields applied to facade analysis. pages 729–736, 03 2014.