

To:

Subject

mit freundlichen Grüssen

John Doe

firstname / lastname

Grafik Designer

job title

Design Agency GmbH

design-agency.com

website

Hauptstraße 13

street

DE-12103 Berlin

location

Phone: +18 2767 9470 1808

phone

Mobile: +18 9153 3990 0008

phone


john.doe@design-agency.com


email


B


I


U














 Templates ▾

 Signature

 Attach

Saved

Discard

Send

Abschlussbericht

Automatisierte Pflege von Kundenkontaktangaben

Author: Tobias Weissert

Betreuer: Prof. Dr. Jürgen Vogel

Experte: Xavier Monnat

Berner Fachhochschule
Departement Technik und Informatik
BSc in Informatik | Vertiefung : Data Engineering

Erklärung der Diplomandinnen und Diplomanden

Selbständige Arbeit

Ich bestätige mit meiner Unterschrift, dass ich meine vorliegende Bachelor-Thesis selbständig durch geführt habe. Alle Informationsquellen (Fachliteratur, Besprechungen mit Fachleuten, usw.) und anderen Hilfsmittel, die wesentlich zu meiner Arbeit beigetragen haben, sind in meinem Arbeitsbericht im Anhang vollständig aufgeführt. Sämtliche Inhalte, die nicht von mir stammen, sind mit dem genauen Hinweis auf ihre Quelle gekennzeichnet.

Name, Vorname

Weissert Tobias.....

Ort, Datum

11.1.2021

Unterschrift



Inhaltsverzeichnis

polypoint

Abstract	2
Ausgangslage	2
Ergebnisse	2
Ausblick	2
1 Einleitung	3
2 Ähnliche Projekte	3
2.1 Apple Mail	3
2.2 Contacts+	3
2.3 SigParser	3
3 Material und Methoden	4
3.1 Material	4
3.2 Vorarbeit	4
3.3 Methodik	5
3.4 Programmiersprache	5
3.5 Umgebung	6
4 Ergebnisse	7
4.1 Projektverlauf	7
4.2 Namens Erkennung	8
4.3 Jobtitel Erkennung	8
4.3.1 Adresserkennung	9
4.3.2 Telefonerkennung	9
4.3.3 Durchsuchen der Webseite nach weiteren Informationen	10
4.3.3.1 Crawling	10
4.3.3.2 Parsing	10
4.3.3.3 Weitere Quellen	10
4.4 Aufruf der API	11
4.5 Resultate	12
4.5.1 Signatur parsen	12
4.5.2 Webscraping	12
5 Diskussion	13
5.1 Rechtliches	13
5.1.1 Verarbeiten von personenbezogenen Daten	13
5.1.2 Speichern von personenbezogenen Daten	13
5.1.3 In der Praxis	13
5.2 Wirtschaftliches	14
6 Folgerungen	15
6.1 Entwicklung	15
6.2 Projektverlauf	15
Glossar	16
Tabellenverzeichnis	17
Abbildungsverzeichnis	17

Abstract

Ausgangslage

Im heutig schnelllebenden Arbeitsmarkt wechseln Arbeitnehmer innert weniger Jahre ihren Arbeitgeber oft mehrfach. Das CRM (Customer Relation Management) System von Lieferanten oder Partnern hat dadurch schnell falsche oder veraltete Daten. Dies führt zu Mehraufwand beim Kontaktieren der verantwortlichen Person sowie Mehrkosten durch unzustellbaren Postversand. Ausserdem wird in CRM Systemen jeweils die Kontaktperson gepflegt, die im Entscheidungsprozess bei Neuanschaffungen involvierten Personen sind dem Lieferanten oft gar nicht bekannt. Ziel dieser Arbeit ist die Abklärung zur technischen Machbarkeit eines Tools zur Lösung dieses Problems.

Ergebnisse

Das Ziel wurde erreicht und es konnte ein Proof of concept für ein Tool zur automatischen Stammdatenpflege erstellt werden. Das Tool wurde gegen ein Datenset mit 9860 Kontakten getestet und dabei sind folgende Ergebnisse entstanden:

Analyse E-mail-Signatur

Von 93% der E-Mails konnte die Information in der Signatur richtig erkannt und richtig gelabelt werden. Auch wenn nicht alle Informationen in der Signatur erkannt wurden, konnten die wichtigsten Attribute wie Name, Telefon, E-Mail und Adresse extrahiert werden und können nun in einem weiteren Schritt gegenüber dem aktuellen Stand im CRM verglichen werden. Allerdings bestand das Datenset vor allem aus europäischen Kontakten. Was nicht getestet wurde, sind E-Mails aus anderen Ländern, welche vermutlich ein etwas anderes Format aufweisen: Anderes Postleitzahlensystem oder andere Labels für Telefon oder Adresse.

Analyse der Firmenwebseite

Obwohl nicht jede Webseite gleich aufgebaut ist, konnten bei 86% der Kontakte mittels Webcrawling zusätzliche

Informationen gewonnen werden, falls diese Person auf der Webseite erwähnt wird. Hierzu wird nach Elementen in optischer Nähe zum Namen gesucht. Während der Entwicklung hat sich aber herausgestellt, dass besonders bei grösseren Unternehmen oft nur die Geschäftsleitung auf der Webseite abgebildet ist. So konnte nur bei 36% der Personen im Testset wirklich einen Eintrag auf der Firmenwebseite gefunden werden. Oftmals waren sogar noch weniger Informationen verfügbar als in der E-Mail-Signatur, einzig interessante weitere Information war oft nur ein Foto der Person.

Ausblick

Die entwickelte API zur Analyse der E-Mail-Signatur kann nun in unser firmeninternes CRM System eingebunden werden. So werden automatisch eingehende E-Mails nach neuen oder abweichenden Kontaktangaben durchsucht und entsprechend angepasst. Die Methode, mittels Webcrawling, die Kontakte weiter zu pflegen, benötigt allerdings noch weitere Aufmerksamkeit.

1 Einleitung

Die Idee für diese Arbeit ist kurz vor Weihnachten bei meinem Arbeitgeber entstanden. Die Idee war es damals, nach Rolle personalisierte Weihnachtskarten zu versenden. Heisst, der CEO erhält auf seiner Karte eine andere Botschaft wie der CFO. Um dies zu ermöglichen, wurden circa 10'000 Kontakte manuell klassifiziert und auf deren Aktualität geprüft. Dabei flossen mehrere Mannwochen in diese Aufgabe und dabei kam aus, wie veraltet unsere CRM Daten oft sind. Trotz sorgfältiger Prüfung kamen circa 5% der Weihnachtskarten unzustellbar zurück, einige weitere konnten zwar zugestellt werden, wir wurden aber informiert, dass die entsprechende Person nicht mehr im Unternehmen tätig ist. Dies brachte uns auf den Gedanken, ob es nicht möglich wäre, diese Aufgabe zu automatisieren, um so das ganze Jahr über den Stand des CRMs aktuell zu halten und Aus- und Neueintritte frühzeitig zu erkennen. Daraus entstand das Konzept für diese Arbeit. Diese Arbeit ist ein Proof-of-Concept ob es technisch möglich ist, diese Arbeit zu automatisieren und so die manuelle Pflege zu vereinfachen und somit schlussendlich Kosten einzusparen.

2 Ähnliche Projekte

2.1 Apple Mail

Apple Mail erlaubt es, die Informationen in der Signatur direkt in das Adressbuch zu importieren. Allerdings beschränkt es sich dabei auf das lokale Adressbuch und dieser Schritt muss vom User jeweils manuell vorgenommen werden.

2.2 Contacts+

Contacts+ bietet einen Assistenten, welcher immer wieder nach Updates für die gegebenen Kontakte sucht. Woher die Daten allerdings genau stammen, ist leider unklar. Vermutlich handelt es sich um Twitter, LinkedIn sowie Bouncelisten von diversen E-Maildiensten.

2.3 SigParser

Dieses Tool erlaubt es, E-Mails und deren Signaturen zu parsen und direkt in ein CRM System zu importieren. Dieses Projekt wurde erst kurz vor der Fertigstellung der Arbeit entdeckt und macht, was das Parsen der Signatur angeht, etwas sehr ähnliches wie diese Arbeit.

3 Material und Methoden

3.1 Material

Mein Arbeitgeber hat freundlicherweise die Daten zur Evaluation dieser Arbeit zur Verfügung gestellt. Es handelt sich dabei um knapp 10'000 Kontakte. Und zusätzlich über 100'000 E-Mails, davon wurden 1496 kontrolliert und klassifiziert und schlussendlich zur Kontrolle verwendet. Diesen Daten durften freundlicherweise verwendet werden, unter dem Vorwand, dass diese nicht veröffentlicht werden und das interne Netzwerk nicht verlassen.

Die E-Mails werden bereits vom CRM vorverarbeitet und sowohl mit HTML Formatierung sowie textbasiert abgespeichert. Für diese Arbeit wurde auf die textbasierte Variante zurückgegriffen. Hier werden auch schon die Antworten der vorhergegangenen Konversation herausgefiltert.

3.2 Vorarbeit

Im Rahmen des Modules Projekt 2 wurden bereits einige Vorarbeiten und technische Abklärungen für diese Arbeit gemacht. Diese werden der Vollständigkeit halber hier trotzdem noch einmal erwähnt, wurden aber ausserhalb der vorgegebenen Zeit erarbeitet.

3.3 Methodik

Für dieses Projekt wird die Projektmethodik Scrum gewählt. Zum einen, um die administrative Arbeiten möglichst gering zu halten, ganz nach dem agilen Manifesto: "Individuals and interactions over processes and tools"¹ zum andern, da ich bei vorhergehenden Projekten sowie in meinem Arbeitsalltag bereits sehr gute Erfahrung mit dieser Methode sammeln durfte. Für die Arbeit stehen 16 Semesterwochen zur Verfügung. Ich habe mich für zweiwöchige Sprints entschieden. Das heisst, nach jeweils zwei Wochen wird der Fortschritt mit dem Betreuer besprochen und geschaut ob das Ziel so weiterhin erreicht werden kann. Dies erlaubt es auch, Änderungen am Vorgehen vorzunehmen und auf allfällige Ressourcenengpässe einzugehen. Somit wird die ganze Arbeit über 8 Sprints verteilt mit einem längeren Unterbruch in den Weihnachtsferien über die Festtage. Der letzte Sprint wurde als Reservesprint gewählt, um allfällige Verspätungen abfedern zu können. Somit bleiben 7 effektive Arbeitssprints übrig.

Die Arbeit ist zweigeteilt und beinhaltet einerseits die Analyse von E-Mail-Signaturen und andererseits das Webscraping der Firmendomain. Der erste Teil wird besonders in Sprint 2 & 3 behandelt. Der Schwerpunkt Webscraping liegt in Sprint 4 & 5. In Sprint 6 werden die Resultate dann mittels eines grösseren Datensets analysiert und die Präzision des Systems ermittelt. Sprint 7 ist für den schriftlichen Teil der Arbeit vorgesehen.

Die Arbeit wird so aufgeteilt, dass sich der erste Sprint, also Sprint 2 & 4 jeweils um die Datenbeschaffung und Aufarbeitung kümmert und im zweiten Teil, Sprint 3 & 5 werden die Daten analysiert.

3.4 Programmiersprache

Um einen schnellen Fortschritt zu erzielen, soll die Arbeit in einer Sprache umgesetzt werden, die möglichst einfach zu hosten ist und in welcher bereits schon Erfahrung vorhanden sind. Der Wahl stand hier zwischen PHP und Python. Schlussendlich wurde zur Umsetzung der API für dieses Proof-of-Concept PHP verwendet, da dies ohne grössere Aufwände auf einem Webserver gehostet werden kann und ich bereits mehrere Jahre Erfahrung damit habe.

¹ Zitat: Agile Manifesto 2001 <https://agilemanifesto.org/>

3.5 Umgebung

Auf folgendem Diagramm ist ersichtlich, wie sich diese Arbeit in die Gesamtumgebung eingliedert.

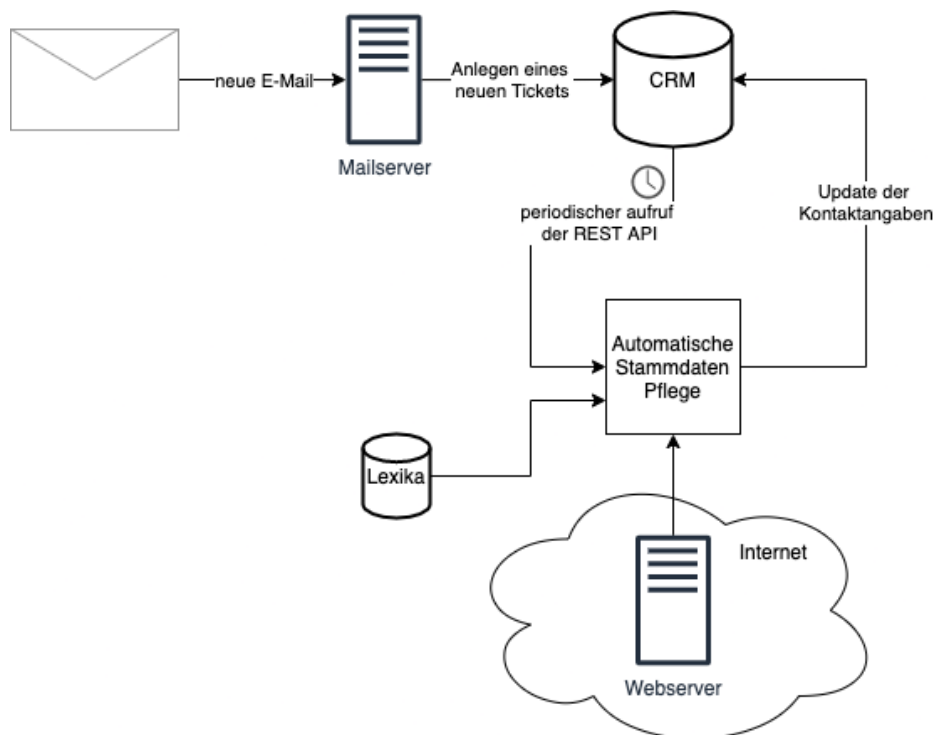


Abbildung 1 Einbettung der Arbeit

Trifft eine neue E-Mail auf dem Mailserver ein, wird diese automatisch in unserem Ticketsystem, das gleichzeitig auch unser CRM ist, abgespeichert. Periodisch werden nun alle E-Mails der API dieser Arbeit übergeben und diese extrahiert daraus jeweils die Kontaktinformationen, welche dann direkt wider im CRM importiert werden.

4 Ergebnisse

4.1 Projektverlauf

Auf der folgenden Grafik ist der Verlauf der Arbeit zu sehen. Die rote Linie beschreibt den Idealverlauf, wobei die grüne Linie den tatsächlichen Verlauf darstellt. Zu sehen ist in der Grafik, dass nicht an allen Tagen gearbeitet wurde, dies, da die Arbeit berufsbegleitend ausgeführt wurde. In Sprint 5 wurde leider keine Arbeit geleistet, da hier ein geschäftliches Projekt kurzfristig zu einem Ressourcenengpass führte. Die verlorene Zeit wurde über die Festtage durch einen zusätzlichen Sprint wieder aufgeholt. Wie es für solche Arbeiten wohl üblich ist, gab es zum Schluss doch noch einen kurzen Endspurt, die Arbeit konnte aber relativ gut aufgeteilt werden, sodass nur wenige Extrastunden notwendig waren.

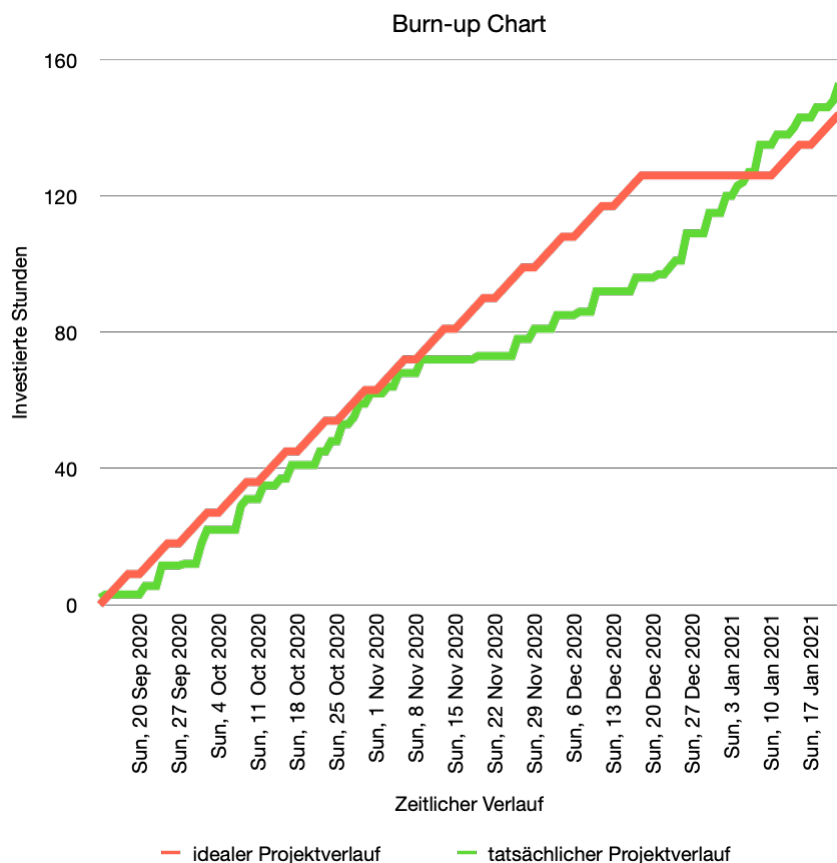


Abbildung 2 Burn-up Chart

4.2 Namens Erkennung

Die Namenserkennung wurde bereits im Vorprojekt umgesetzt. Da in einer Signatur nicht aus dem Wortkontext geschlossen werden kann, dass es sich bei einem Wort um einen Namen handelt, wie dies viele andere Wortklassifizierungsalgorithmen tun. Zudem folgen Namen auch nicht einem bestimmten Muster. Darum wurde dies über ein Lexikon zur Erkennung gelöst, in dem alle Vor- und Nachnamen hinterlegt sind. Hierzu wurde wikidata.org als Datenquelle verwendet. Wikidata ist eine Graphdatenbank, welche Wikipedia unterstützen soll und so das dort gesammelte Wissen maschinenlesbar bereitstellt. Um alle Namen herauszulesen, wurden folgende Sparql Skripte verwendet.

Familienname	Männliche Voramen	Weibliche Voramen
<pre>SELECT ?NAME WHERE { ?n wdt:P31 wd:Q101352. ?n wdt:P1705 ?NAME. ?n wdt:P282 wd:Q8229 }</pre>	<pre>SELECT ?NAME WHERE { ?n wdt:P31 wd:Q12308941; wdt:P1705 ?NAME; wdt:P282 wd:Q8229. }</pre>	<pre>SELECT ?NAME WHERE { ?n wdt:P31 wd:Q11879590; wdt:P1705 ?NAME; wdt:P282 wd:Q8229. }</pre>

Abbildung 3 Wikidata Export Scripts

Dabei steht P: jeweils für die Beziehung und Q für ein Objekt. In diesem Falle P31 instance of und z.B. Q101352 für familyname. Diese Daten wurden danach als CSV exportiert und in einer MYSQL Datenbank abgespeichert. Danach konnte über SQL darauf zugegriffen werden. Dabei konnten zu diesem Zeitpunkt 19964 männliche und 12083 weibliche Vornamen sowie 276544 Nachnamen ins Lexikon aufgenommen werden. Nach dem Abschluss des Projekt 2 wurden nicht erkannte Namen aus dem Testset Manuel hinzugefügt, was allerdings dazu führte, dass 100% der Namen im Testset auch im lexikon vorhanden sind.

4.3 Jobtitel Erkennung

Um die Funktion der Person in dem Betrieb zu erkennen, musste auch ein Lexikon aufgebaut werden. Hier zu diene zusätzlich jobs.ch als Datenquelle. Da es sich bei jobs.ch um eine single Page-Applikation handelt, hat die Seite eine API, welche die Jobs aus dem Backend holt. Diese API kann aufgerufen werden, um alle Jobtitel herunter zu laden und danach zu speichern. Die API verfügt über ein Paging und erlaubt es, maximal 100 Jobs gleichzeitig abzufragen, danach muss ein neuer Request abgesendet werden. Deshalb wurde im Abstand von jeweils einer Sekunde, um den Server nicht zu stark zu belassen, alle Jobtitel abgefragt und abgespeichert. Dies waren zu diesem Zeitpunkt 65714 verschiedene Jobs. Danach wurde mit einer Regular expression alle Jobs in einer männlichen und in einer weiblichen Variante getrennt und erneut wider in einer MYSQL Datenbank abgespeichert. Nicht zu parsende Jobs wurden ignoriert. Damit konnte ein Lexikon mit 13788 Berufen erstellt werden. Dieses kann nun verwendet werden, um die Funktion in der Signatur oder auf der Webseite zu erkennen.

4.3.1 Adresserkennung

Nach Analyse der Daten hat sich herausgestellt, dass die Adressen immer das gleiche Format haben. Diese setzen sich aus einer 4-5 Stellen Postleitzahl und einer Ortschaft, einer Strasse und einer Hausnummer oder alternativ dem Begriff Postfach zusammen. Dabei war in den Daten die Hausnummer nie grösser als 3 Ziffern. Auch die Recherche online hat ergeben, dass solche grossen Hausnummern, besonders in Industriegebiete, nicht existieren. Ausserhalb der DACH Region existieren weitere Formate wie Postleitzahlen mit Buchstaben oder das Amerikanische System mit Street und Avenue. Da mein Arbeitgeber ausschliesslich in dieser Region tätig ist, wird ausschliesslich dieses Format unterstützt.

4.3.2 Telefonerkennung

Eine Telefonnummer ist rein optisch und auch technisch relativ einfach zu erkennen. Sie besteht ausschliesslich aus Zahlen oder Zeichen wie ()/+. Oft sind in einer Signatur allerdings auch mehrere Telefonnummern für Direktwahl, mobile oder fax vorhanden. Hier wird aus Praktikabilität jeweils einfach die erste Nummer genommen und die weiteren Nummern als Alternative mitgegeben.

4.3.3 Durchsuchen der Webseite nach weiteren Informationen

4.3.3.1 Crawling

In einem ersten Schritt war der Plan, die ganze Domain, durch folgen aller Links, nach der Person zu durchsuchen. Doch gerade bei grösseren Firmen wie z.B. der BFH war dies aus Performancegründen nicht möglich. Bei folgen jedes Linkes ist die Anzahl Seiten exponentiell angestiegen und selbst nach drei Sprüngen wurde die Person nicht immer gefunden.

Die zweite Idee war ein Datenset zu übernehmen, dass die gesamte Seite bereits vorcrawled und welches dann durchsucht werden kann. Dabei habe ich mir das Projekt [commoncrawl](https://commoncrawl.org/)² angeschaut, welches ein grosses Datenset an vorgecrawelten Seiten bereits zur Verfügung stellt. Allerdings enthielt dieses nur sehr wenige schweizer Seiten und wenn oft auch nicht die komplette Seite. Zudem war die Datenmenge enorm gross um diese auf einem kleinen Webserver, der mir zur Verfügung stand abzuspeichern.

Deshalb habe ich mich entschieden das crawling nicht selbst zu machen, sondern von Google übernehmen zu lassen. Mittels der Google API können Suchanfragen an Google gestellt werden und über das keyword site : kann die Domain auf der gesucht werden soll eingeschränkt werden. Somit erhält man alle Seiten, auf der die Person erwähnt wird und muss dann nur noch diese parsen.

4.3.3.2 Parsing

Von dieser Seite wird dann der HTML Sourcecode heruntergeladen und darin nach dem Namen der Person gesucht. Von dort aus wird dann solange das Elternelement dazugenommen, bis sich der Inhalt darin verändert. Somit erhält man die optisch nahen Elemente, diese werden dann mit den gleichen Methoden wie auch bei der E-Mailsignatur nach Inhalten durchsucht.

4.3.3.3 Weitere Quellen

Der Plan war es auch weitere Quellen nach Informationen zu durchsuchen. Dafür hat sich in einem ersten Schritt die Plattform LinkedIn angeboten. Hier kann genau gleich wie auch beim crawlen der Domain mittels der Google API und der einschränkung auf eine Domain nach der Person gesucht werden. Allerdings zeigt einem hier LinkedIn jeweils nur ein einziges mal die entsprechende Seite an und beim zweiten mal wird die Registrierungsseite angezeigt. Ich hab mehrere Methoden probiert um dies zu umgehen, löschen der Cookies, aufrufen der Seite mittels Proxy, allerdings war die Abfrage jeweils nur einmal pro IP Adresse möglich. Einzige Methode war mittels VPN die IP Adresse zu verschleiern, allerdings hätten mir hier die Anzahl IP Adressen gefehlt um für das ganze Datenset nach Informationen zu suchen. Bei Recherchen bin ich auf den LinkedIn Sales Navigator gestossen, was wohl die offizielle Lösung für so etwas von LinkedIn ist. Dieser Zugriff kostet aber mehrere tausend Dollar pro Jahr, weshalb ich hier nicht weitergesucht habe. Mir ist allerdings aufgefallen, dass in den Vorschautexten von Google jeweils bereits einige Informationen enthalten waren. Google kann also LinkedIn crawlen. Aber auch mit dem User Agent und einer IP Adresse aus der Google Cloud Platform war ich nicht erfolgreich. Leider enthält dieser Text nur wenige zeichen und der Jobtitel wird oft abgeschnitten. Aus zeitlichen Gründen, habe ich aber hier nicht mehr weiter geforscht.

² <https://commoncrawl.org/>

4.4 Aufruf der API

Die API kann mittels http GET aufgerufen werden. Dabei können 3 verschiedene Varianten gewählt werden. Die Antwort wird JSON encodiert

Requestparameter :

body : E-Mail-Text (parst die E-Mail-Signatur)

email : E-Mail-Adresse (analysiert die E-Mail-Adresse sucht auf der domain nach der Person)

string : Name (analysiert lediglich den Namen)

Response :

Domain : Domain der Firma

Company

Name : Name der Firma

Segment : Segment in der die Firma tätig ist

Email : E-Mail-Adresse

Phone : Liste aller Telefonnummern

Address : Adresse im Format (Strasse Nr, PLZ Ort)

Firstname

Gender : Geschlecht anhand des Vornamens

Value : Vorname

Language : Sprache aus der der Name stammt (experimental)

Lastname

Value : Nachname

Language : Sprache aus der der Name stammt (experimental)

Image

Age : Jahrgang der Person anhand des Bildes (geschätzter Wert)

Gender : Geschlecht anhand des Bildes

Src : URL des Bildes

More : URL auf der die Person erwähnt wird

Job : Funktion der Person im Unternehmen

4.5 Resultate

4.5.1 Signatur parsen

Gesamthaft wurden 1496 E-Mails analysiert. Dabei sind folgende Ergebnisse entstanden.

Vorname richtig erkannt	92.51%
Nachname richtig erkannt	92.51%
Geschlecht richtig erkannt	86.96%
Adresse richtig erkannt	91.37%
E-Mail richtig klassifiziert	100%
Telefon richtig erkannt	98.39%
Funktion	36.49%

Tabelle 1 Resultate Signatur parsen

4.5.2 Webscraping

Gesamthaft wurden 9860 Kontakte analysiert. Dabei sind folgende Ergebnisse entstanden.

Name auf der Domain gefunden	85.99%
Bild der Person gefunden	35.99%
Funktion der Person gefunden	30.1%
Segment des Arbeitgebers	93.99%

Tabelle 2 Resultate Webscraping

5 Diskussion

5.1 Rechtliches

Bei den verarbeiteten Daten handelt es sich um personenbezogene Daten, dass heisst, diese Daten können jeweils einer Person zugeordnet werden und geben Rückschlüsse über diese Person. Obwohl die Schweiz nicht Teil der EU ist, verarbeitet unser Unternehmen auch Daten von europäischen Staatsbürgern, weshalb in diesem Falle die vor zwei Jahren eingeführte Datenschutzgrundverordnung zum Zuge kommt.

5.1.1 Verarbeiten von personenbezogenen Daten

Das Herunterladen und Verarbeiten von öffentlichen Daten ist grundsätzlich kein Problem. Wäre das ein Problem, hätte auch Google hier ein Problem, resp. jeder Webbrowser, der Daten aus dem Internet lädt.

5.1.2 Speichern von personenbezogenen Daten

Grundsätzlich gilt, dass sobald Daten einer Person abgespeichert werden, sei dies Manuel oder automatisch muss grundsätzlich das Einverständnis der Person eingeholt werden und die Person kann jederzeit fordern alle gespeicherten Daten vorzuweisen und auf Wunsch auch löschen lassen.

5.1.3 In der Praxis

Das Einverständnis einzuholen ist selbst im aktuellen manuellen Kontext strikt unmöglich, weshalb dies in den Datenschutzrichtlinien auf unserer Webseite geregelt ist. Wer mit der Firma Kontakt aufnimmt bestätigt damit automatisch auch, dass Personenbezogene Daten abgespeichert werden. Wir handhaben es dann so, dass wir jeweils einmal im Jahr die Person darüber informieren, welche Informationen über sie gespeichert sind und ihr die Möglichkeit geben diese zu korrigieren oder löschen zu lassen.

5.2 Wirtschaftliches

Während meiner Arbeit habe ich auch einige Gespräche mit Unternehmern und Beratern aus dem Bereich CRM geführt. Dabei stoss meine Arbeit auf grosses Interesse. Deshalb habe ich auch einmal nachgefragt, was denn Ihnen ein solcher Dienst Wert sein würde. Dabei wurden mir folgende Zahlen genannt.

Dienstleistung	Preis
Kontrolle eines Kontaktes	10rp/ Konakt:
Update eines Kontaktes	20rp/ Konakt
Neuanlegen eines Kontaktes	50rp/ Konakt

Tabelle 3 Preisliste

Würde diese Dienstleistung nun in einem mittleren bis grösseren KMUs, wie meines Arbeitgebers verwendet, würde dies nach meiner Schätzung nach pro Jahr pro Firma circa folgenden Umsatz generieren :

$$8000 \times 10\text{Rp} + 1200 \times 20\text{Rp} + 800 \times 50\text{Rp} = \mathbf{1440.- CHF}$$

Würde man die gleiche Rechnung bei einem Grosskonzern machen mit oftmals mehreren Millionen Kontakten, kähme dabei natürlich auch der 10 bis 100 fache Betrag heraus. Würde man einen solchen Grosskonzern oder einige kleinere KMU als Kunden gewinnen, könnte dieses Produkt durchaus rentabel betrieben werden. Ob der Kunde allerdings wirklich bereit ist, diese Preise pro Kontakt zu bezahlen, kann leider noch nicht gesagt werden, ein Test würde sich aber durchaus lohnen.

6 Folgerungen

6.1 Entwicklung

Aufgrund der Umstellung des aktuellen CRM bei meinem Arbeitgeber hat die Bedeutung für diese Arbeit intern leider etwas an Dringlichkeit verloren. Da ich meinen aktuellen Arbeitgeber in naher Zukunft verlassen werde, wird dieses Projekt vermutlich nicht mehr in Betrieb genommen. Allerdings bin ich auf grosses Interesse bei einem Berater für das neue CRM gestossen, welcher Interesse hat, das Projekt als eigenes Produkt zu vertreiben. Aktuell bin ich darum in Gesprächen wie dies von statten gehen könnte. Vermutlich müsste die Codebasis neu geschrieben werden um dies in ihr aktuelles Produkt zu integrieren und langfristig wartbar zu machen. Die erarbeiteten Lexika könnten aber durchaus weiter verwendet werden.

6.2 Projektverlauf

Durch die Agile vorgehensweise konnte sehr gut auf Änderungen eingegangen werden. Besondere Herausforderung war für mich vorallem die Disziplin im Homeoffice auf Grund der COVID19 Situation so wie eine solche Arbeit berufsbegleitend zu schreiben. Was meiner Meinung nach nicht so gut lief, ist die Kommunikation mit dem Betreuer und Experten, ich hätte hier offener und öfters meinen aktuellen Stand mitteilen sollen. Zudem war auch mein online Repository auf Github nicht immer auf dem aktuellsten Stand, was es für Sie schwer gemacht hat, mehr über den aktuellen Stand zu erfahren. Ansonsten ist die Arbeit aber sehr gut gelaufen und ich bin sehr zufrieden mit dem Ergebniss.

Glossar

CEO : Chief Executive Officer, Chef einer Firma

CFO: Chief Financial Officer, Verantwortlich für die Finanzen im Betrieb

CRM-System: Customer-Relationship-Management-System, System zum Verwalten von Kunden

Klassifizieren: im Kontext von Data Engineering, zuordnen einer Klasse

Proof-of-Concept: Machbarkeitsnachweis

Scrum: Vorgehensmodell insbesondere in der agilen Softwareentwicklung

Sprint: Zeitintervall in dem an einer neuen Version gearbeitet wird

Lexikon: Nachschlagewerk, Sammlung von allen möglichen Varianten

Testset: Datensammlung die zum Testen einer Applikation verwendet werden

Paging: Unterteilen der Antwort in Teilhappen

API: application programming interface, Programmierschnittstelle zur Interaktion mit dem Programm

Single-Page-Applikation: Webanwendung, die Daten zur Laufzeit anfragen kann

DACH Region: deutschsprachiger Raum, Deutschland, Österreich Schweiz

Crawling: Programm zur Sammlung von Daten aus dem Internet

Parsing: Zerlegen von Daten in deren Bestandteile

Repository: Verzeichnis zur Speicherung digitaler Daten

Tabellenverzeichnis

Tabelle 1 Resultate Signatur parsen.....	12
Tabelle 2 Resultate Webscraping	12
Tabelle 3 Preisliste	14

Abbildungsverzeichnis

Abbildung 1 Einbettung der Arbeit	6
Abbildung 2 Burn-up Chart	7
Abbildung 3 Wikidata Export Scripts	8

Anhang

Der Sourcecode und alle zum Projekt gehörenden Files sind unter <https://github.com/supertoub/bachelorarbeit> abgelegt.

CRM Berater: Sebastian Fluri, Nordfabrik AG

Inhaber der Testdaten: POLYPOINT AG

Projekthosting: Metanet AG

Quelle Namenslexikon: wikidata.org

Quelle Joblexikon: jobs.ch + wikidata.org

Quelle Firmensegment: Google Maps API