Big Data Praktikum: Attribute extraction from eCommerce product descriptions

Gregor Pfänder Thilo Brummerloh

25. Mai 2021

1 Thema

Produktseiten von Onlineshops enthalten oft viele unzureichend strukturierte Produktbeschreibungen oder sogar gar keine Beschreibungen. Zum Preisvergleich des gleichen Produkts auf verschiedenen Webseiten muss allerdings bekannt sein um welches Produkt, wo angeboten wird. So können Preise nicht nur von Produkten auf gut strukturierten Webseiten mit hoher Datenqualität verglichen werden. Es müsste allerdings eine weitere Verarbeitung von abgefragten Daten geben. Diese ist aufgrund der Anzahl an Produkten und eCommerce-Webseiten aber nicht händisch zu erledigen.

2 Fragestellung

Es sollte möglich sein mithilfe von einem Computerprogramm diese Arbeit automatisch durchzuführen. Wenn Produktspezifikationen innerhalb eines Freitextes vorliegen, sollte es möglich sein daraus Wörter und Wortgruppen zu erkennen und einer Attributgruppe zuzuweisen. Im Kontext dieser Arbeit werden Produkttitel und -beschreibungen auf eCommerceseiten verwendet um Produktmarke und Produktnummer mithilfe von Machine Learning zu extrahieren. Dieser Prozess wird auch Named Entity Recognition genannt.

3 Methodenüberblick

Alle Methoden lassen sich unter dem Begriff Named Entity Recognition zusammenfassen. Nachfolgend werden ausgewählte Methoden erklärt. Bei allen handelt es sich um Machine Learning Ansätze.

3.1 Naive Bayes

Naive Bayes Ansätze waren sehr beliebt[1] Eine Möglichkeit des NER liegt darin das Klassifizierungsproblem mit einem Naive Bayes Klassifikators zu lösen. Bei Naive Bayes wird sich ‚supervised learning' zu nutzen gemacht. Es werden also gelabelte Trainingsdaten benötigt. Anhand der Trainingsdaten werden für alle Werte die Wahrscheinlichkeiten bestimmt, dass der Wert einer bestimmten Klasse zugehört (bzw. eine Hypothese erfüllt). In unserem Fall könnten die Werte die Wörter einer Sequenz und mögliche Klassen ‚Attribut', ‚Attributwert' und ‚Keins von beiden' sein. Nachdem die Wahrscheinlichkeiten bekannt sind, können anhand des Naive Bayes Klassifikators die ungelabelten Daten immer der Klasse mit der höchsten Wahrscheinlichkeit zugeordnet werden. Um eine erfolgreiche NER Methode auf Basis von Naive Bayes entwickeln zu können benötigt es noch weitere Schritte wie beispielsweise das Verlinken zwischen Attributen und Attributwerten. Bewertung: Der Naive Bayes Klassifikator ist ein weiteverbreitetes Mittel zur Textklassifizierung und wird beispielsweise bei Spam Filtern erfolgreich eingesetzt um Wörter in die Klassen ‚Spam' (Schlecht) oder ‚Ham' (gut) einzuordnen. Allerdings ist der Ansatz wie bereits im Namen steht naiv. Er geht davon aus, dass keine Beziehungen zwischen den Werten bestehen. Diese sich also nicht gegenseitig beeinflussen. Diese Annahme ist kritisch bei der Textverarbeitung. In unserem Anwendungsfall könnte sich das unter anderem negativ auf die Extraktion von Attributwerten welche aus mehreren Wörtern zusammengesetzt sind auswirken. Bei solchen Attributwerten bestehen Abhängigkeiten zwischen mehreren Wörtern, die das Modell beachten sollte. [2]

3.2 CNNs

Neueste Ansätze verwenden CNNs zur Klassifikation. Zhu u. a. 2018 haben erfolgreich ein CNN verwendet um 2017 die genaueste Methode zur Named Entity Recognition von verschiedenen Biologieliteraturkorpora vorzustellen. Die GRAM-CNN genannte Methode hat laut den Autoren für das labeling von Biologieliteratur einen Genauigkeitsvorteil indem, nicht wie in klassischen LSTMs der ganze Satz betrachtet wurde, sondern nur die um ein Wort liegenden Nachbarworte. Die Worte, dieser N-Gram genannten Wortketten, werden in einem ersten Schritt zu einer Darstellung umgewandelt, die nicht das Wort selber enthält, sondern das Wort durch seine einzelnen Buchstaben als Vektor darstellt. Dazu wird dem Wort noch ein Part-of-Speech tag zugewiesen und der Character des Worts als Vektor dargestellt. Die Vektorisierung erfolgt durch Abgleich der Worte mit vorgefertigte Bibliotheken die auf ähnlichen Korpora Vektoren berechnet haben. Das Part-of-Speech tag enthält Informationen zu Abhängigkeiten eines Wortes zu anderen. Eine Eingabe in das GRAM-CNN ist also ein n-gram, das wie beschrieben durch seine Buchstaben mit Zusatzinformationen dargestellt wird. Innerhalb des CNNs wird diese Information versteckt verarbeitet und es liefert die features der Worte an ein CRF in dem die Verbindung von mehreren Worten erkannt werden kann. Damit sind auch auseinander geschriebene zusammengehörige Worte als solche erkennbar.
CNNs wurden auch in Lee, Chung u. a. 2019 und Lee, Park und Cho 2020 verwendet

[1] Ghani u. a. 2006.
[2] (Quelle: Text Mining for Product Attribute Extraction) (Quelle: https://course.elementsofai.com/de/3/3)

um Produkteigenschaften aus Benutzerbewertungen zu extrahieren. Mit den extrahierten Eigenschaften werden von den Autoren letztlich Sentiment-Scores der Benutzerbewertungen erstellt, die nicht nur die Erwähnung von bewertbaren Wörtern zählen, sondern auch die relative Gewichtung von verschiedenen Eigenschaften des bewerteten Produkts herausfinden.

3.3 RNNs (LSTMs)

RNNs sind die Standardmethode für Textklassifikation, dabei wird meist die Spezialform der LSTMs verwendet.[3] RNNs verwenden wie die vorherigen Methoden Trainingsdaten um einen Klassifikator zu trainieren. Eingaben sind Texte aus denen bereits die Interessanten Textstellen mit ihrer Beschreibung extrahiert wurden. Damit wird ein Neuronales Netzwerk trainiert. Das trainierte Netzwerk ist dann in der Lage beliebige Texte einzulesen und die darin enthaltenen Entitäten in Form von Markennamen zu erkennen und zuzuordnen. Mit diesen Informationen sollte es möglich sein größere Korpora von Produktbeschreibungen automatisch mit Informationen anzureichern.

4 Daten

4.1 Trainingsdaten

Je nach Methodenwahl werden unterschiedliche Daten benötigt. Zur Entity Resolution wäre nur ein Menge von ungeordneten Produktbeschreibungen nötig. Wenn allerdings zusätzlich ein Neural Network trainiert werden soll ist es auch nötig einen bereits gelabelten Datensatz zum training zu haben.

4.2 Beschreibung Datensatz

Zur Verfügung steht ein Datensatz mit 56005 Produkten in Form einer `.csv`-Datei. Bei den Produkten, die in den Daten erfasst wurden, handelt es sich um unterschiedliche Arten von Haushaltsgeräten. Die Produkte reichen dabei von Waschmaschinen und Wäschetrocknern bis zu Kühlschränken und Mikrowellen. Eine genaue Zusammenfassung welche Produkttypen in den Daten vorkommen, kann durch eine erste Datenbegutachtung nicht getroffen werden. Der Produkttyp wäre also ein weiteres Attribut, bei dem eine Extraktion sinnvoll wäre. Die Werte wurden aus den Produktseiten verschiedener eCommerce Seiten extrahiert. Die Aufteilung der Daten nach Quelle ist wie folgt:

Quelle	Anzahl
shopee.my	24849
appliancesconnection.us	21434
sharafdg.ae	5443
productreview.au	2502
spencerstv.us	1777

[3] Majumder u. a. 2018.

Nachdem die .csv Datei mithilfe von Pandas in ein Dataframe überführt wurde, erhält man die Nachfolgende Datenstruktur:

id	source	name	productdescription	url	brand	modelnumber

Die Zeilen von Interesse sind ‚name' und ‚productdescription'. In diesen Feldern stehen die Sequenzen, aus denen die Attribute extrahiert werden sollen. Außerdem ist der Datensatz bereits für die Attribute ‚brand' und ‚modelnumber' gelabelt. (der Absatz könnte weggelassen werden. Um einen ersten Eindruck der Daten zu bekommen wird nachfolgen der erste Eintrag des Datensatzes aufgeführt:

id	000266f5e7ab2344315290174dfb75f7
source	appliancesconnection.us
name	Broan TEN136WW
productdecription	Broan TEN136WW Overview The Tenya 1 Series Under Cabinet Range Hood
url	https://www.appliancesconnection.com/broan-ten136ww.html?zipcode=20001
brand	Broan
modelnumber	TEN136WW

Bei der ersten Begutachtung der Daten sind einige Problemstellen und mögliche Störfaktoren . . . :
1. "-Zeichen wird in der Zeile ‚productdescription' oft als Größen Bezeichnung benutzt: Dadurch denkt Pandas, dass der Satz vorbei ist und jedes weiter Komma wird als neue Zeile interpretiert. . .
2. Chinesiche Zeichen bei mind. einem Produkt
3. Andere Sonderzeichen: verarbeitbar oder Störfaktoren?
4. Verschiedene Quellen könnten zu verschiedenen Mustern führen: Beobachten wie gut unser Modell je nach Quelle funktioniert.
5. Marke und Modelnummer zu einfach zu extrahieren?: weitere Attribute festlegen und von Hand labeln?

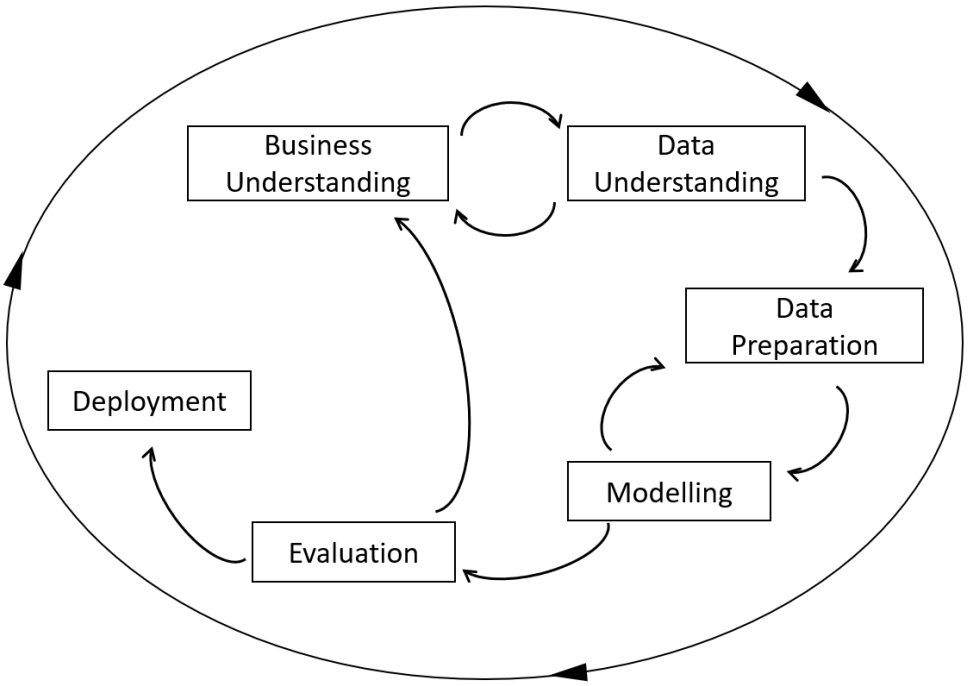
5 Implementierung

5.1 Architektur

Das gesamte Programm soll mithilfe von Python implementiert werden. Innerhalb von Jupyter Notebooks sollen Codeabschnitte einzeln ausführbar sein. Die weit verbreiteten Bibliotheken spacy und keras sollen als Grundbausteine verwendet werden, da sie nützliche Funktionen bereits implementieren.
Spacy ist dabei zur Textbearbeitung und -vorbereitung nützlich. Darin enthalten sind WErkzuege zur Tokenisierung und süezielleren Werkzeugen wie zum Stemming der Tokens. Damit könnten Wörter auf ihren Wortstamm zurückdekliniert werden. In Keras ist eine Verbindungsstelle zwischen python und der Tensorflow Bibliothek hergestellt. Tensorflow ist der für dieses Projekt notwendige Baustein um künstliche Neuronale Netzwerke einfach zu verwenden.

Der Aufbau der Neuronalen Netze ist noch nicht endgültig festzulegen. Nach erster Implementierung werden verschiedene Tiefen und Zusammensetzungen von Ebenen ausprobiert um zu einem möglichst guten Ergebnis zu kommen. Dabei muss auch ein Ausgleich zwischen Geschwindigkeit und Genauigkeit der Klassifizierung gefunden werden.

Abbildung 1: CRISP-DM in Eigendarstellung Cleve und Lämmel 2016



5.2 Vorgehensweise:Ablauf

Die Vorgehensweise orientiert sich an der des CRISP-DM von Cleve und Lämmel 2016 (vgl. Bild).

Die Schritte Business Understanding und Data Understanding sind gleichzusetzen mit Kapitel. . . (Aufgabenstellung) bzw. Kapitel. . . (Beschreibung Datensatz) Die eigentliche Entwicklung der Methode kann in die Schritte Data Preparation, Modelling, Evaluation und Deployment aufgeteilt werden. Die Schritte verschwimmen in der Praxis alle miteinander. Das eigentliche Konzept an dem sich orientiert wird besteht darin, dass das Modell schrittweise verbessert wird. Es wird also zuerst ein ‚einfaches' Modell entwickelt und dann nach einer Evaluation immer wieder verbessert, bis ein zufriedenstellendes Ergebnis vorliegt.

5.3 Training/Validation/Testing der ML Methode

Der Datensatz soll in drei Teile aufgespalten werden. Diese drei Teile werden zum Training, der Validierung und für die abschließenden Tests verwendet.

Das Trainingsset soll der größte Teil mit etwa 70% der Daten sein. Mit diesen Daten soll das Netzwerk mit seinen Gewichten darauf trainiert werden Markennamen und Produktnummern zu erkennen. Je 15% sollen für Validierung und Testen verwendet werden. Die Validierung ist zur Feinjustierung von Hyperparametern zuständig. Das Testen soll vor allem overfitting des Modells überprüfen. Wenn ein auf den Trainingsdaten gut funktionierendes Modell trainiert wurde sollte es ähnlich gut für die bisher ungesehenen Testdaten funktionieren. Ist dies nicht der Fall, könnte ein Vergleich auf Overfitting hindeuten.

Abschließend sollen verschiedene automatisch mitgeschnittene Metriken in Grafiken aufbereitet werden um die Ergebnisse anschaulich darstellen zu können.

5.4 Verbesserungsmöglichkeiten/Mögliche Erweiterungen

Mehr Attribute Attention-Schicht CRF (Conditional random field)

5.4.1 CRF

Eine Möglichkeit besteht darin Conditional Random Fields (CRF) wie in Majumder u. a. 2018 und Zheng u. a. 2018 zu verwenden. Dabei würde CRF die Aufgabe der Output Schicht übernehmen und wäre für das Labeling zuständig. Wird keine CRF-Schicht verwendet, dann berechnet das LSTM das passende Label für jedes Token eigenständig und unabhängig. Durch die CRF-Schicht passiert das Labeling der gesamten Eingabesequenz auf einmal, wodurch es möglich ist, die Abhängigkeiten zwischen Wörtern, die in Nachbarschaft zueinander stehen, zu berücksichtigen. CRFs sind immer dann hilfreich, wenn Attributwerte gesucht werden, die sich aus mehreren Wörtern zusammensetzen können. In diesem Projekt ist das das Attribut ‚Marke' bzw. ‚brand'. Markennamen bestehen nicht selten aus mehr als einem Wort (Beispiele aus dem Datensatz: ‚Fulgor Milano', ‚Fisher Paykel', ‚Ready Hot', ...). CRF hilft dabei, auch diese Markennamen als einen Attributwert, der aus mehreren Tokens besteht, zu erkennen.

5.4.2 Attention

Eine weitere Verbesserung könnte durch eine Attention Schicht wie in Majumder u. a. 2018 und Zheng u. a. 2018 zustande kommen. Diese würde nach den LSTM Schichten und vor der Output Schicht eingefügt werden. Attention wird bei NER dafür eingesetzt, um den Fokus auf die wichtigen Konzepte in den Sequenzen zu legen, anstatt alle Informationen gleich zu behandeln. Dadurch wird das CRF bzw. die Output Schicht bei der Entscheidung optimal unterstützt und kann sich auf diese Konzepte fokussieren. Bei Attention wird für jedes Token ein Wert lt berechnet, der beschreiben soll, wie viel Beachtung einem Token geschenkt werden soll, um den Kontext der Nachbarn zu beschreiben. Allerdings legen die Dokumente nah, dass durch Attention nicht unbedingt verbesserte Labeling Ergebnisse erzielt werden können. Die Schicht wird dort eher dazu verwendet, die Entscheidungsfindung des Netzwerks anhand der als wichtig erkannten Tokens zu beschreiben und nachvollziehen zu können.

Literaturverzeichnis

Cleve, Jürgen und Uwe Lämmel (2016). *Data mining*. 2. Auflage. Studium. Berlin und Boston: De Gruyter Oldenbourg. ISBN: 978-3-11-045675-2.

Ghani, Rayid u. a. (2006). "Text mining for product attribute extraction". In: *ACM SIGKDD Explorations Newsletter* 8.1, S. 41–48. ISSN: 1931-0145. DOI: 10.1145/1147234.1147241.

Lee, Younghoon, Minki Chung u. a. (2019). "Extraction of Product Evaluation Factors with a Convolutional Neural Network and Transfer Learning". In: *Neural Processing Letters* 50.1, S. 149–164. ISSN: 1370-4621. DOI: 10.1007/s11063-018-9964-8.

Lee, Younghoon, Jungmin Park und Sungzoon Cho (2020). "Extraction and prioritization of product attributes using an explainable neural network". In: *Pattern Analysis and Applications* 23.4, S. 1767–1777. ISSN: 1433-7541. DOI: 10.1007/s10044-020-00878-5.

Majumder, Bodhisattwa Prasad u. a. (2018). *Deep Recurrent Neural Networks for Product Attribute Extraction in eCommerce*. URL: http://arxiv.org/pdf/1803.11284v1.

Zheng, Guineng u. a. (2018). "OpenTag". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Hrsg. von Yike Guo und Faisal Farooq. New York, NY, USA: ACM, S. 1049–1058. ISBN: 9781450355520. DOI: 10.1145/3219819.3219839.

Zhu, Qile u. a. (2018). "GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text". In: *Bioinformatics (Oxford, England)* 34.9, S. 1547–1554. DOI: 10.1093/bioinformatics/btx815.