

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN  
INSTITUT FÜR STATISTIK

---

## Zutatenbasierte Empfehlungssysteme für Rezepte

---



## Bachelorarbeit

**Autor** Viktoria Szabo

**Betreuer** Prof. Dr. Christian Heumann

**Datum** München, den 28. Juli 2020

### **Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und dabei keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht. Die Arbeit wurde bisher weder gesamt noch in Teilen einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

---

Viktoria Szabo

---

Ort, Datum

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Food Science</b>	<b>3</b>
<b>3</b>	<b>Methodik</b>	<b>6</b>
3.1	Empfehlungssysteme . . . . .	7
3.2	Association Rules . . . . .	8
3.3	Word Embeddings . . . . .	11
3.4	Netzwerk Modelle . . . . .	14
<b>4</b>	<b>Anwendung</b>	<b>16</b>
4.1	Deskriptive Auswertung . . . . .	16
4.2	Association Rules . . . . .	20
4.3	Word Embeddings . . . . .	22
4.4	Netzwerk Modelle . . . . .	27
<b>5</b>	<b>Evaluierung</b>	<b>31</b>
<b>6</b>	<b>Fazit und Ausblick</b>	<b>33</b>

# 1 Einleitung

Nahrung ist für das Leben unerlässlich und sie ist grundlegend für unsere alltägliche menschliche Erfahrung. Die meisten Menschen haben sich durch Bücher und Durchprobieren Wissen darüber angeeignet, welche Lebensmittel zusammenpassen, welche einem selbst schmecken und welche gesund sind. In vielen anderen Bereichen verlassen wir uns darauf, dass Algorithmen Produkte identifizieren, die unserem Geschmack entsprechen. Sei es der Film, der von einer Streaming-Plattform für den Abend vorgeschlagen wird, das passende Buch in einem Online-Geschäft oder Personen mit denen wir uns auf einer Karriere-Plattform vernetzen sollen. Bei all diesen Anwendungen laufen datenbasierte Empfehlungsalgorithmen im Hintergrund, um uns passende Vorschläge zu machen.

Beim Essen verlassen wir uns jedoch meistens noch auf unser Gefühl. Zwar nutzen viele bereits Plattformen für Online-Rezepte, um sich ein neues Rezept rauszusuchen oder sich inspirieren zu lassen, jedoch läuft dies größtenteils ohne Empfehlungsalgorithmen im Hintergrund ab. Dabei könnte es vor allem in diesem Bereich wichtig sein, sich „beraten“ zu lassen, denn unsere Ernährung spielt eine extrem wichtige Rolle für unsere Gesundheit und unser allgemeines Wohlbefinden (Nordström et al., 2013). Empfehlungsalgorithmen könnten je nach Wunsch der Person darauf ausgerichtet werden, eine gesunde und ausgewogene Ernährung zu unterstützen oder aber auch zu einer Diät oder einem bestimmten sportlichen Ziel beizutragen.

Empfehlungsalgorithmen können aber auch bei einem Problem namens *information overload* von Vorteil sein. In mehreren Feldexperimenten zeigten die Psychologen Sheena Iyengar und Mark Lepper im Jahr 2000, dass zu viele Auswahlmöglichkeiten die Bereitschaft senken, eine finale Wahl zu treffen und zusätzlich auch die Zufriedenheit am Ende der Auswahl abnimmt (Iyengar and Lepper, 2000). Eine verkleinerte Auswahl, welche von Empfehlungsalgorithmen festgelegt wird, könnte dem entgegenwirken.

Neben den gesundheitlichen und persönlichen Aspekten gibt es aber auch eine Vielzahl praktischer Vorteile. Anwendungen, Produkte, Informationen und auch Rezepte sind in so großen Mengen vorhanden, dass es ohne Sortier- und Empfehlungsalgorithmen enorm zeitaufwendig ist, das Richtige zu finden. In anderen Bereichen haben deswegen Empfehlungsalgorithmen, sogenannte Recommender Systems, in den letzten Jahren große Bedeutung gewonnen und sind vor allem im Zusammenhang mit Big Data unerlässlich geworden. Auf Plattformen, bei denen es um Verkäufe oder die Anzahl von



Nutzern geht, können gute Empfehlungen über Erfolg und Misserfolg bestimmen. Dies hat den Streamingdienst Netflix 2006 sogar dazu veranlasst, einen 1 Millionen US-Dollar Preis an denjenigen zu verleihen, der seinen Empfehlungsalgorithmus verbessern könnte (<https://www.netflixprize.com/>, zuletzt aufgerufen am 25.7.2020).

Neben Online-Plattformen sind Empfehlungsalgorithmen jedoch auch in anderen Anwendungsgebieten vorstellbar. Beispielsweise zeigten Osadchiy et al. in einer 2019 erschienenen Publikation mit Daten einer Ernährungserhebung, dass ein guter Empfehlungsalgorithmus eine genauere und bessere Ernährungsverfolgung mit sich bringen kann (Osadchiy et al., 2019).

Trotz des großen Erfolgs von Recommender Systemen in all diesen Bereichen werden sie in der Ernährung bisher eher selten angewendet. Dies spiegelt sich auch in der bisherigen Forschung wider. Während Empfehlungsalgorithmen allgemein umfangreich erforscht werden, existiert weit weniger Literatur für die Anwendung dieser im Bereich der Ernährung. Zwei grundlegende Motivationen in der Food Recommendation Literatur können festgestellt werden: die Personalisierung von Empfehlungen und die gesundheitliche Optimierung. In Kapitel 2 wird es hierzu eine Literaturzusammenfassung geben.

So unterschiedlich die Anwendungsmöglichkeiten von Empfehlungssystemen sind, kann auch die Ausgestaltung dieser sein. Je nach Verfügbarkeit der Daten können inhaltsbasierte Systeme, kollaborative Systeme oder auch Hybrid-Systeme die richtige Wahl sein. Hierfür steht eine Vielzahl an Methoden zur Verfügung. Ziel dieser Arbeit ist es, verschiedene Empfehlungssystem-Modelle für die Empfehlung von Zutaten und Rezepten auf Basis von Chefkoch.de-Rezeptdaten anzuwenden und diese zu vergleichen. Für eine zutatenbasierte Rezeptsuche sollen anschließend diese Modelle anhand der Chefkoch.de-Rezepte in einer Shiny-App implementiert werden. Hierfür wird in Kapitel 2 erst auf aktuelle Forschung im Bereich der Ernährung eingegangen. Kapitel 3 beschäftigt sich mit der Methodik der Recommender Systeme und stellt die Theorie von drei Alternativen für die Erstellung von Empfehlungen vor. Kapitel 4 beschreibt die Implementierung der vorher erklärten Methoden. Hierfür wird zuerst der Datensatz genauer vorgestellt. Weiterhin werden deskriptive Ergebnisse und Probleme bei der Implementierung aufgezeigt. Anschließend soll in Kapitel 5 kurz auf Evaluierungsmöglichkeiten eingegangen werden. Kapitel 6 fasst die Vor- und Nachteile der Methoden zusammen und gibt einen Ausblick, mit welchen Methoden der Empfehlungsalgorithmus weiter verbessert werden könnte.

## 2 Food Science

Lebensmittelbezogene Studien können unterschiedlichste Dinge untersuchen wie z.B. die Verbesserung der Gesundheit oder das Verständnis der kulinarischen Kultur. Mit der enormen Entwicklung von sozialen Netzwerken und dem Austausch über das Internet haben sich in den letzten Jahren umfangreiche Lebensmitteldaten angesammelt. Menschen laden Bilder von Lebensmitteln, Rezepte, Kochvideos und Ernährungstagebücher hoch und tauschen sie aus. Während sich historisch hauptsächlich Köche mit langjähriger Erfahrung mit der Untersuchung von Nahrung auseinandersetzten, hat sich in den letzten Jahren eine ganze Ernährungswissenschaftssparte entwickelt, die verschiedenste Wissenschaften vereint. Mit der Zunahme von Rechenleistung und den Mengen an Daten spielt nun auch die datengetriebene Analyse von Ernährung eine immer größere Rolle. Im Folgenden wird relevante Literatur aus diesem Bereich vorgestellt, wobei auch zwei Grundkonzepte im Rezepte-Empfehlungsbereich genauer betrachtet werden. Einen umfangreichen Überblick zur *Food-Science* bezogenen Literatur findet man beispielsweise in Min et al. (2019) und Mouritsen et al. (2017).

Verschiedene Bereiche und Daten werden beim sogenannten Food Computing betrachtet. Min et al. (2019) definieren den Begriff folgendermaßen: Food Computing erfasst und analysiert heterogene Lebensmitteldaten aus verschiedenen Quellen zur Wahrnehmung, Erkennung, Suche, Empfehlung und Überwachung von Lebensmitteln und Essen. Beim Food Computing werden rechnergestützte Ansätze angewandt, um lebensmittelbezogene Fragen in Medizin, Biologie, Gastronomie und Agronomie zu behandeln.

Freyne und Berkovsky beschreiben 2010 ein Rezepte-Empfehlungssystem, welches personalisiert ist und auf gesundheitliche Aspekte achtet. Dabei werden explizite Angaben der Nutzer, vergangenes Browsing-Verhalten und Essenstagebücher für den Vergleich von verschiedenen Algorithmen verwendet (Freyne and Berkovsky, 2010). 2011 wurde in der Publikation von Ueda et al. ein Algorithmus vorgeschlagen, welcher unter Verwendung des persönlichen Browsing-Verhaltens und der Kochhistorie mit einer Präzision von 60 bis 83% die Lieblingszutat der Nutzer vorhersagen konnte. Hierfür nutzten sie Häufigkeit und Charakteristika der Zutaten, die in bereits vom Nutzer gekochten Rezepten vorkamen, und bildeten daraus Scores für Zutaten und Rezepte Ueda et al. (2011). In dem Rezepte-Empfehlungssystem von Elsweiler et al. (2017) sollen Rezepte, die typischerweise einem Benutzer vorgeschlagen werden, mit gesünderen ähnlichen Rezepten ersetzt werden.

Hierfür nutzten sie drei verschiedene Modelle, Random Forest, Logistische Regression und Naive Bayes, bei denen Meta-Daten von Rezepten sowie zugehörige Bilder, Nährstoffgehalt und Popularität des Rezeptes als Variablen festgelegt wurden, um passende Rezepte für Nutzer vorzuschlagen. Sie konnten dabei auch zeigen, dass Bilder einen großen Einfluss auf die Rezeptauswahl der Nutzer haben. Nezis et al. präsentieren 2018 ein personalisiertes zutatenbasiertes Empfehlungssystem, welches mittels neuronaler Netze und unter Verwendung von Zutaten und deren Nährstoffen versucht, personalisierte und gesunde Rezepte vorzuschlagen. Hierfür erstellten sie Embeddings (siehe Kapitel 3.3) für Rezepte und nutzten neben den Zutaten auch die Bewertungen der Rezepte, um zusätzliche Rezeptempfehlungen zu generieren (Nezis et al., 2018). Gao et al. (2019) verwenden Image-Processing mit einem hierarchischen Attention Neural Network, um hochgeladene Bilder von Rezepten in eine personalisierte Rezeptempfehlung mit einzubeziehen.

Zwei wichtige Konzepte, die bei der Untersuchung von Lebensmitteln und deren Verwendung und Kombination in Rezepten aufkommen, heißen Food-Pairing und Food-Bridging. Food-Pairing beschreibt das Zusammenkommen von komplementären Lebensmitteln. Welche Lebensmittel als komplementär betrachtet werden können, basiert größtenteils auf chemischen Untersuchungen der Geschmacks- und Aromenkomponenten von Lebensmitteln. Jedoch werden mittlerweile solche Untersuchungen auch empirisch, also datenbasiert durchgeführt.

Der Geschmack von Lebensmitteln hängt von vielen Faktoren ab. Geschmack bezieht sich auf die Fähigkeit, die Empfindungen zu erkennen, welche als die 5 Grundgeschmacksrichtungen bekannt sind, für welche die menschliche Zunge Rezeptoren besitzt: süß, sauer, salzig, bitter und umami („herzhaft, fleischig“). Wir nehmen den Geschmack auf der Zunge wahr, während das Aroma hauptsächlich durch die Nase wahrgenommen wird. Das Aroma, welches wir über die Nase wahrnehmen, beeinflusst jedoch einen Großteil dessen, was wir Geschmack nennen (Small and Green (2012) oder Burdock (2019)).

Die Bandbreite der Aromen in einzelnen Lebensmitteln ist groß. So genannte Schlüsselaromen stechen als Aromaverbindung in einem Lebensmittel besonders hervor und bestimmen damit sein Geruchsbild. Dieses Zusammenspiel von Geschmacksrichtungen und Lebensmittelaromen wurde historisch von Köchen untersucht. Jedoch gibt es mittlerweile auch Untersuchungen von Chemikern, die das Aroma-Profil der Lebensmittel mittels Gas-Chromatographie und Massenspektrometrie (GC-MS) ausarbeiten. Ein

Aroma muss in einer Zutat einen bestimmten Schwellenwert erreichen, um vom Menschen wahrgenommen zu werden. Durch die Verstärkung von Wechselwirkungen erzeugen einige Aromen, die unterhalb dieser Schwelle liegen, einen wahrnehmbaren Geruch. Aus diesen Ergebnissen extrahierten die Wissenschaftler die für den menschlichen Geruchssinn relevanten Aromadaten (vgl. Burdock (2019) oder Foodpairing Website: <http://www.foodpairing.com/en/home>, zuletzt abgerufen am: 22.7.2020). Abbildung 1 zeigt beispielhaft die Aromazusammensetzung von Erdbeeren:

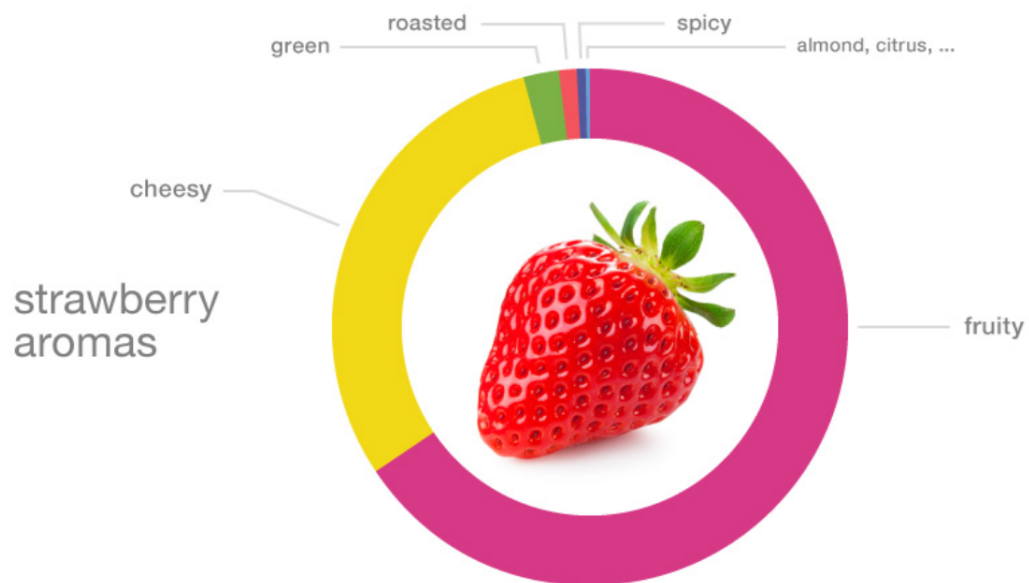


Abbildung 1: Aromakomponenten einer Erdbeere; Quelle: <https://www.foodpairing.com/en/science-behind>, zuletzt aufgerufen am: 25.7.2020

Diese Aromen können anschließend dafür verwendet werden, zusammenpassende Zutaten zu finden. Eine Online-Seite namens [foodpairing.com](http://www.foodpairing.com) verwendet dieses System bereits, um neue Rezeptkreationen zu unterstützen. Und auch die Forscher Ahn et al. untersuchten dieses Prinzip im Jahr 2011 empirisch anhand einer Großzahl von Rezepten und den Aromakomponenten der Zutaten. Ziel war es herauszufinden, ob es allgemeine Muster gibt, die die in Rezepten verwendeten Zutatenkombinationen bestimmen, oder ob sich Prinzipien finden lassen, die über individuelle Geschmäcker und Rezepte hinausgehen. Hierfür verwendeten sie Geschmacks- und Aromanetzwerke der den Rezepten zugrundeliegenden Zutaten. Sie fanden, dass die westliche Küche dazu neigt, Zutatenpaare zu verwenden, die viele Geschmackskomponenten gemeinsam haben, was die so genannte *Food-Pairing* Hypothese unterstützt. Im Gegensatz dazu fanden sie heraus, dass ostasiatische Rezepte

Food-Pairing eher vermeiden (Ahn et al., 2011). In Anlehnung hierzu schlugen Simas et al. (2017) die *Food-Bridging* Hypothese vor. Die Food-Bridging Hypothese geht davon aus, dass auch wenn zwei Inhaltsstoffe keine starke molekulare oder empirische Affinität aufweisen, sie durch eine Kette von paarweisen Affinitäten verbunden werden können. Das heißt, in einem Netzwerkmodell, wie es von Ahn et al. (2011) verwendet wurde, stellt eine Kette einen Weg dar, der die beiden Zutaten verbindet, wobei der kürzeste Weg die stärkste paarweise Affinitätskette zwischen den beiden Zutaten darstellt. Simas et al. (2017) fanden, wie auch Ahn et al. (2011), bei der Analyse von Rezepten, dass die Verwendung von Food-Pairing oder Food-Bridging kulturell bestimmt wird. Sie beobachteten in ihrer Arbeit, dass in unterschiedlichen traditionellen Küchen sowohl Food-Pairing als auch Food-Bridging betrieben wird. Sie haben hierfür vier verschiedene regionale Küchen verglichen und generierten folgende Erkenntnis: Ostasiatische Rezepte neigen in einem Extrem dazu, sowohl Food-Pairing als auch Food-Bridging zu vermeiden. Lateinamerikanische Rezepte folgen in dem anderen Extrem beiden Prinzipien. Die südostasiatische Küche vermeidet Food-Pairing und folgt Food-Bridging. Und die westliche Küche folgt Food-Pairing und vermeidet Food-Bridging. Food-Pairing und Food-Bridging sind damit Hypothesen, die mögliche Mechanismen hinter der Zusammenstellung von Zutaten in Rezepten beschreiben können. Food-Pairing verstärkt den Geschmack, indem Zutaten in einem Rezept mit ähnlichen Geschmackskomponenten gemischt werden, während Food-Bridging versucht, den Kontrast zwischen den Zutaten zu glätten, indem es auch nicht ganz passende Geschmackskomponenten über mehrere Zutaten zusammenführt. Im Folgenden wird ebenfalls empirisch die Komplementarität der Zutaten anhand einer Großzahl von Rezepten untersucht. Da deutsche Rezepte von Chefkoch.de verwendet werden, ist davon auszugehen, dass ein Großteil der Rezepte der westlichen Küche entsprechen und deswegen eher Food-Pairing anwenden.

### 3 Methodik

Empfehlungssysteme aller Art bedienen sich vieler Erkenntnissen aus klassischen statistischen Modellen. Im Grunde können die meisten statistischen und Machine Learning Modelle auch für Empfehlungsanwendungen umgebaut werden (Aggarwal et al., 2016). Im Folgenden werde ich zuerst einen Einblick in grundlegende Empfehlungsmechanismen

geben und anschließend auf die Theorie hinter drei Implementierungen eingehen: Association Rules, Word Embeddings und Netzwerk Modelle.

Angemerkt sei, dass im Folgendem das Wort „Bewertung“ universal auch für das Kauf- und Nutzungsverhalten der Nutzer verwendet wird. Hierbei spiegelt sich die Annahme wider, dass eine Kauf- oder Nutzungsentscheidung implizit für eine positive Bewertung steht.

### 3.1 Empfehlungssysteme

Meistens werden im Recommendation Bereich Algorithmen einem von zwei Typen zugeordnet. Das sind zum einen die kollaborativen (englisch collaborative) und zum anderen die inhaltsbasierten (englisch content-based) Empfehlungssysteme. Es gibt jedoch auch noch weitere Empfehlungssysteme, wie zum Beispiel das wissensbasierte System, bei dem Nutzer ihre Präferenzen angeben und diese dann direkt in die Empfehlung miteinbezogen werden können. Hybride Systeme, die verschiedene Empfehlungsarten kombinieren, sind oftmals besonders erfolgreich, da damit Schwächen der einzelnen Systeme ausgeglichen werden können. Die Einteilung wird anhand der Datengrundlage, die die Empfehlungssysteme zum Empfehlen nutzen, vorgenommen (Aggarwal et al., 2016).

**Kollaborative Empfehlungssysteme** (oder **Collaborative Filtering**) nutzen für Empfehlungen das vergangene Verhalten der Benutzer. Sie basieren auf der Annahme, dass ähnliche Benutzer vergleichbare Bewertungsverhalten aufweisen und daher artverwandte Objekte vergleichbare Bewertungen erhalten. Ähnlichkeiten werden hier anhand der bestehenden Bewertungen berechnet. Das heißt, es müssen keine weiteren Kenntnisse über Charakteristika des Objekts oder des Nutzers vorhanden sein, sondern nur die oben genannten Interaktionen zwischen verschiedenen Nutzern und Objekten. Diese Herangehensweise bietet vor allem Vorteile, wenn die Beurteilung der Nutzer nicht anhand von äußerlich bestimmbar Kriterien geschieht oder wenn schlicht keine weiteren Eigenschaftsdaten für Nutzer oder Objekte vorliegen (Aggarwal et al., 2016).

Für die Anwendung in dieser Arbeit könnte dies bedeuten, dass Zutaten (=Objekte) empfohlen werden, welche in vielen Rezepten (= Benutzer) zusammen vorkommen. Algorithmen und Methoden, die in diesem Bereich oft verwendet werden, sind zum Beispiel Association Rules, Nearest Neighbour, Decision Trees aber auch Techniken der Netzwerkanalyse.

**Inhaltsbasierte Empfehlungssysteme** (oder **Content-Based Filtering**) basieren auf Charakteristika der Nutzer oder der Objekte. Sie empfehlen Objekte, die ähnlich den Objekten sind, die ein Benutzer bereits verwendet hat oder verwenden möchte. Alternativ empfehlen sie Objekte, die ähnliche Benutzer verwendeten. Dazu ist es erforderlich die Ähnlichkeit zwischen zwei Objekten oder Benutzern anhand deren Eigenschaften bestimmen zu können. Diese Herangehensweise bietet vor allem Vorteile, wenn neue Objekte auftreten, die noch keine auswertbaren Interaktionen aufweisen. Außerdem ist sie hilfreich, wenn Nutzer die Anwendung anonym verwenden und somit auch keine Informationen zu den Nutzern verfügbar sind. Ein weiterer Vorteil von inhaltsbasierten Empfehlungen ist, dass eine resultierende Empfehlung gut zu erklären ist. Wenn ein Nutzer versteht, warum eine konkrete Empfehlung gemacht wurden, ist es tendenziell wahrscheinlicher, dass er diese annimmt (Aggarwal et al., 2016).

Für diese Herangehensweise ist es erforderlich, die Ähnlichkeit zwischen zwei Objekten bestimmen zu können, z.B. über die Anwendung einer Clusteranalyse. Hierfür werden zusätzliche Attribute für die Zutaten und/oder die Rezepte benötigt (z.B. billig, schnell zum Zubereiten, besonders nährstoffreich, bei Rezepten bestimmte regionale Küchen (z.B. asiatisch)). Auch Regressionsmodellierung wird in diesem Bereich verwendet.

Da in dem für diese Arbeit verwendeten Datensatz nur spärlich weitere Charakteristika für die Rezepte und Zutaten vorliegen, liegt der Fokus auf kollaborativen Empfehlungsalgorithmen. Für eine genauere Auflistung und einen genaueren Vergleich verschiedener Methoden im Bereich Food Recommendation sei auf Trattner and Elsweiler (2017) verwiesen.

## 3.2 Association Rules

Die Analyse von Assoziationsregeln ist eine Technik, um zu identifizieren, wie Elemente miteinander assoziiert sind. Es gibt drei gebräuchliche Methoden zur Messung der Assoziation (Hornik et al., 2005).

### 1. Support

Der Support sagt aus, wie beliebt ein Produkt ist, indem der Anteil berechnet wird, in wie vielen Transaktionen das Produkt vorkommt. Abbildung 2 zeigt Warenkörbe mit unterschiedlichen Produktkombinationen. Der Support von Äpfeln wird berechnet,

indem der Anteil von Äpfeln an allen Transaktionen angegeben wird. Im graphischen Beispiel entspricht dies 4 von 8 bzw. 50%. Der Support kann nicht nur für einzelne Produkte, sondern auch Produktkombinationen berechnet werden. Zum Beispiel beträgt der Support von Apfel, Bier, Reis 2 von 8 oder 25%.

$$\text{Support} \{\text{🍏}\} = \frac{4}{8}$$

Transaction 1	🍏 🍺 🍚 🍗
Transaction 2	🍏 🍺 🍚
Transaction 3	🍏 🍺
Transaction 4	🍏 🍏
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍏

Abbildung 2: Beispiele für Transaktionen und die Berechnung des Supports für Apfel; Quelle: <https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html>, zuletzt aufgerufen am 25.7.2020

## 2. Confidence

Die Confidence gibt an wie wahrscheinlich Artikel Y gekauft wird, gegeben Artikel X wird auch gekauft. Damit kann die Confidence als eine Schätzung der bedingten Wahrscheinlichkeit  $P(Y|X)$  interpretiert werden, d.h. der Wahrscheinlichkeit die Items der rechten Seite der Regel in Transaktionen zu finden, unter der Bedingung,



dass diese Transaktionen auch die linke Seite enthalten. Dies wird anhand des Anteils der Transaktionen mit Artikel X gemessen, in denen auch Artikel Y vorkommt geteilt durch den Anteil an Transaktionen in denen nur Artikel X vorkommt. In dem Beispiel von oben beträgt die Confidence von Apfel -> Bier 3 von 4 oder 75%. Abbildung 3 zeigt diese Berechnung.

$$\text{Confidence} \{ \text{🍏} \rightarrow \text{🍺} \} = \frac{\text{Support} \{ \text{🍏}, \text{🍺} \}}{\text{Support} \{ \text{🍏} \}}$$

Abbildung 3: Berechnung der Confidence für Bier gegeben Apfel wurde bereits gekauft

### 3. Lift

Der Lift gibt die Wahrscheinlichkeit an, dass Artikel Y gekauft wird, wenn Artikel X bereits gekauft wurde, wobei dafür kontrolliert wird, wie beliebt Artikel Y ist. In der Tabelle 2 ist der Lift von Apfel -> Bier 1, was keine Assoziation zwischen den Artikeln impliziert. Ein Lift-Wert von mehr als 1 bedeutet, dass Artikel Y wahrscheinlich gekauft wird, wenn Artikel X gekauft wurde, während ein Wert von weniger als 1 bedeutet, dass Artikel Y wahrscheinlich nicht gekauft wird, wenn Artikel X gekauft wurde. Die Berechnung wird in Abbildung 4 beschrieben.

$$\text{Lift} \{ \text{🍏} \rightarrow \text{🍺} \} = \frac{\text{Support} \{ \text{🍏}, \text{🍺} \}}{\text{Support} \{ \text{🍏} \} \times \text{Support} \{ \text{🍺} \}}$$

Abbildung 4: Berechnung des Lifts für Bier gegeben Apfel wurde bereits gekauft

Aus diesen Kennzahlen können anschließend zum Beispiel mit dem Apriori Algorithmus die Assoziations Regeln abgeleitet werden, welche dann wiederum für Empfehlungen benutzt werden können (vgl. Agarwal et al. (1994) oder Agrawal et al. (1993)). Gegeben der Definition von Agrawal et al. (1993) werden beim Apriori Algorithmus Assoziations Regeln der Form  $X \rightarrow Y$  mit Minimum Support und Minimum Confidence gebildet, bei denen Y immer nur ein Item und X auch mehrere enthalten kann.

Osadchiy et al. (2019) verwenden Association Rules, um in einer Ernährungserhebung genauere Ergebnisse zu erhalten. Die Nutzer geben erst Zutaten an, die sie zu sich genommen haben und der Algorithmus schlägt weitere Zutaten vor, die wahrscheinlich ebenfalls

in dem Gericht vorkamen und konsumiert wurden. Für die Empfehlung verwenden sie die Kennzahlen Lift und Confidence. Am Ende vergleichen sie die vom Algorithmus empfohlenen Zutaten mit von Experten geschriebenen Vorschlagslisten und können an einem Testdatensatz aus der Erhebung weitaus bessere Ergebnisse erzielen.

### 3.3 Word Embeddings

Word Embeddings haben in Natural Language Processing (NLP) große Fortschritte bewirkt. Word Embeddings verwenden Vektoren mit kontinuierlichen Zahlen, um jedes Wort in einem Vokabular darzustellen. Diese Vektoren haben  $n$  Dimensionen, normalerweise zwischen 50 und 300, die verschiedene Aspekte des Wortes repräsentieren. Durch die Vektoren werden die Wörter auf einen kontinuierlichen Vektorraum abgebildet, in dem semantisch ähnliche Wörter nahe beieinander vorkommen, während unähnliche Wörter weit voneinander entfernt sind. Das Erlernen dieser Vektoren geschieht komplett automatisiert und unsupervised, d.h. ohne die konkrete Vorgabe an Ergebnissen mit denen der Algorithmus lernen soll. Hierfür werden meistens Neuronale Netze verwendet. Beim Training des Modells werden die  $n$  Dimensionen der Kontextwörter so lange angepasst bis das Modell das gesuchte Wort oder die gesuchten Wörter vorhersagen kann. Der Grundgedanke hinter dieser Mechanik ist, dass Wörter, die in ähnlichen Kontexten vorkommen, ähnliche Bedeutungen haben sollten.

Die zwei bzw. drei bekanntesten Algorithmen in diesem Bereich heißen GloVe und word2vec. Letzterer besteht aus den zwei Algorithmen Continuous Bag-of-Words und Continuous Skip-Gram. Diese werden in den folgenden Abschnitten beschrieben.

#### **word2vec**

In der im Jahr 2013 erschienenen Publikation von Mikolov et al. wurden die beiden word2vec-Algorithmen vorgeschlagen, die zu einer großen Welle der Popularisierung von Word Embeddings im NLP führten. Die beiden word2vec-Algorithmen namens Continuous Bag-of-Words (CBOW) und Continuous Skip-Gram verwenden neuronale Netze mit einem Input-, einem Projektions- und einem Output Layer. Abbildung 5 zeigt den Aufbau dieser neuronalen Netze. Die allgemeine Idee hinter CBOW besteht darin, das Fokuswort auf der Grundlage von Kontextwörtern vorherzusagen. Die Reihenfolge der Kontextwörter hat keinen Einfluss auf die Vorhersage, daher der Name Bag-of-Words. Während Skip-

Gram versucht, die Kontextwörter eines Quellworts vorherzusagen. Dies geschieht mit einer iterativen Anpassung der Wortdimensionen während des Trainings, so dass eine Verlustfunktion minimiert wird. Die Verlustfunktion beschreibt dabei den Unterschied zwischen der Vorhersage des Modells und dem wahren Wort, welches gegeben des Input Kontexts resultieren soll. Der Output des Modells, welches für die Vorhersage benutzt wird, besteht hierbei aus einem (bei GloVe, bei Skip-Gram aus mehreren) Vektor(en), die Wahrscheinlichkeiten für jedes Wort des Vokabulars enthalten. Diese Wahrscheinlichkeiten eines Vektors addieren sich zu 1 auf und werden in der Originalpublikation von Mikolov et al. (2013) mittels einer Softmax Funktion in dem letzten Layer des neuronalen Netzes berechnet. Das Wort mit der größten Wahrscheinlichkeit wird dann als Vorhersage ausgewählt (Mikolov et al., 2013).

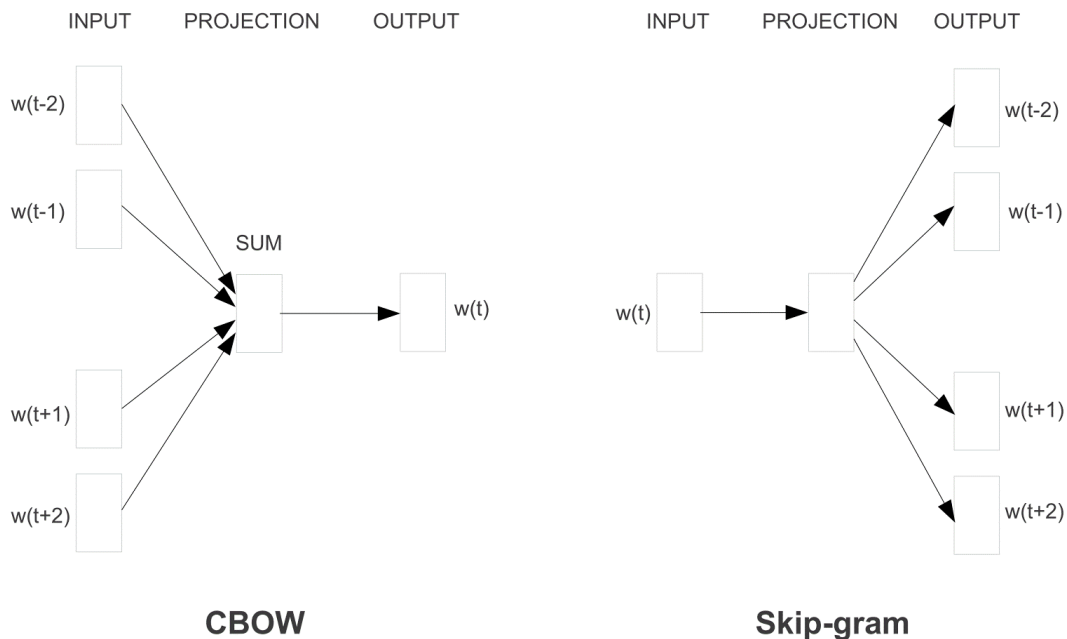


Abbildung 5: Neuronale Netze der word2vec Algorithmen; Quelle: Mikolov et al. (2013)

### GloVe

GloVe steht für Global Vector, was den globalen Charakter dieses Modells unterstreicht. Im Gegensatz zu den zuvor beschriebenen Algorithmen word2vec stützt sich GloVe nicht nur auf lokale Kontextinformationen, sondern bezieht auch globale Kookkurrenzstatistiken mit ein. Anstatt die Word Embeddings aus einem neuronalen Netz zu extrahieren, welches ein Ziel wie die Vorhersage von Nachbarwörtern (CBOW) oder die Vorhersage des Fokuswortes (Skip-Gram) verfolgt, werden die Word Embeddings direkt optimiert, so dass

das Produkt zweier Wortvektoren gleich dem Logarithmus der Häufigkeit ist, mit der die beiden Wörter nahe beieinander stehen bzw. im selben Kontext vorkommen werden:

$$w_i^T w_k + b_i + b_k = \log(X_{ik}), \quad (3.1)$$

wobei  $b_i$  und  $b_k$  Bias Terme sind,  $w_i$  und  $w_k$  die zwei Wortvektoren und  $X_{ik}$  die Anzahl wie oft die Wörter  $w_i$  und  $w_k$  zusammen vorkommen darstellt. Die Optimierung wird mit einem kleinsten Quadrate Schätzer durchgeführt.

Das Modell baut auf der Idee auf, dass aus der Kookkurrenzmatrix semantische Beziehungen zwischen Wörtern abgeleitet werden können und dass das Verhältnis der Kookkurrenzwahrscheinlichkeiten zweier Wörter mit einem dritten Vergleichswort eher auf ihre semantische Assoziation hinweist als eine direkte Kookkurrenzwahrscheinlichkeit (Pennington et al., 2014).

Für die Analyse von Zutatenkombinationen wurden diese Methoden bisher kaum verwendet, obwohl sich die Modellierung von „Zutaten Embeddings“ lohnen könnte. Den Kontext bilden in diesem Fall die Rezepte und ein mögliches Ziel könnte die Vorhersage einer fehlenden Zutat sein. Jaan Altosaar schlug online die Verwendung von Word Embeddings für die Modellierung von zusammen gehörenden Zutaten vor und nannte dies food2vec (<https://jaan.io/food2vec-augmented-cooking-machine-intelligence/>). Für die Modellierung verwendet er fastText, eine Abwandlung von Skip-Gram. Diese Website liefert eine einfache Implementierung und graphische Darstellung, aber keine genaue Untersuchung von Modelleigenschaften und Auswirkungen der Verwendung von Word Embeddings. Sauer et al. (2020) nehmen die food2vec Implementierung als Ausgangspunkt, betrachten jedoch die Auswirkungen der Verwendung von fastText im Vergleich zu CBOW. Sie finden, dass fastText bzw. der zugrunde liegende Skip-Gram Algorithmus zu Wortvektoren führt, bei denen ähnliche Vektoren Zutatenkomplemente repräsentieren, d.h. Zutaten die sich in einem Rezept gut ergänzen. Im Gegensatz dazu führen sie an, dass CBOW eher Zutaten-Substitute abbildet, d.h. Zutaten, die in Rezepten miteinander ausgetauscht werden können. Sie zeigen, dass die Verwendung von Embeddings zu guten Ergebnissen in der Vorhersage von komplementierenden und substituierenden Zutaten führt. Da diese relativ neuen Algorithmen im NLP Bereich große Verbesserungen mit sich brachten, werden sie für den Anwendungsfall in dieser Arbeit implementiert und deren Empfehlungen mit anderen Modellen verglichen.

### 3.4 Netzwerk Modelle

Die Netzwerkanalyse macht es möglich, Beziehungen, sowie das Entstehen von Beziehungen oder Verbindungen, zwischen einer Mehrzahl an Akteuren zu analysieren und zu modellieren. Netzwerke sind in vielen verschiedenen Bereichen des Lebens aufzufinden. Unter anderem gibt es künstliche neuronale Netzwerke, biologische Netzwerke, soziale Netzwerke und viele mehr. Netzwerke bestehen grundsätzlich aus sogenannten Knotenpunkten („Nodes“), die Akteure darstellen, und Kanten („Edges“), welche das Vorhandensein einer Beziehung bzw. Verbindung repräsentieren. Außerdem können Netzwerke auch mit Strömen („Flows“), also gerichteten Verbindungen zwischen zwei Knoten, dargestellt werden (Kolaczyk and Csárdi, 2014).

Aufgrund der Beziehung zwischen Rezepten und Zutaten ist die Netzwerkanalyse ein leistungsfähiges Instrument zur Erforschung und Darstellung von Zutatenassoziationen und Kochgewohnheiten von Menschen. Das Modell eines "Rezeptegraphen" unter Verwendung von Daten aus Online-Rezept-Websites wurde erstmals Ahn et al. (2011) in ihrer Publikation "Flavor Network and the Principles of Food Pairing" vorgestellt. Dort erstellten sie zunächst ein Netzwerk zwischen den Zutaten und ihren zugehörigen Aromen und leiteten anschließend daraus ein Netzwerk zwischen den Zutaten ab, wobei die ermittelten Verbindungen darauf basieren, dass Zutaten identische Aromenkomponenten aufweisen. Mithilfe dieses Netzwerks, welches Zutatenkomplemente aufzeigen soll, untersuchten sie die Food-Pairing Hypothese in unterschiedlichen regionalen Küchen (Ahn et al., 2011). Teng et al. (2012) erweiterten diese Idee von Zutatennetzwerken, indem sie neben dem Komplement-Netzwerk auch ein Substitute-Netzwerk vorschlugen. Im Gegensatz zu Ahn et al. (2011) erstellten sie das komplementäre Zutaten-Netzwerk nicht auf Basis der chemischen Aromen, sondern leiteten die Komplementarität von dem gemeinsamen Auftreten der Zutaten in Rezepten ab. Das Zutaten-Substitutionsnetzwerk hingegen erstellten sie aus von Benutzern generierten Änderungsvorschlägen der Rezepte. Eine weitere Neuerung war, dass in dem Komplement-Netzwerk, die Verbindungen (Edges) zwischen zwei Zutaten a und b mit der Pointwise Mutual Information (PMI) gewichtet wurden. Der PMI gibt die Wahrscheinlichkeit an, dass zwei Inhaltsstoffe zusammen auftreten, gegen die Wahrscheinlichkeit, dass sie getrennt auftreten. Komplementäre Bestandteile treten in der Regel viel häufiger gemeinsam auf, als man zufällig erwarten würde. Abbildung 6 zeigt das Zutatenkomplement-Netzwerk, welches sie, wie eben beschrieben, erstellt haben.

$$PMI(a, b) = \log \frac{p(a, b)}{p(a)p(b)}, \quad (3.2)$$

$$p(a, b) = \frac{\text{Anzahl Rezepte mit a und b}}{\text{Anzahl Rezepte}}, \quad (3.3)$$

$$p(a) = \frac{\text{Anzahl Rezepte mit a}}{\text{Anzahl Rezepte}}, \quad (3.4)$$

$$p(b) = \frac{\text{Anzahl Rezepte mit b}}{\text{Anzahl Rezepte}} \quad (3.5)$$

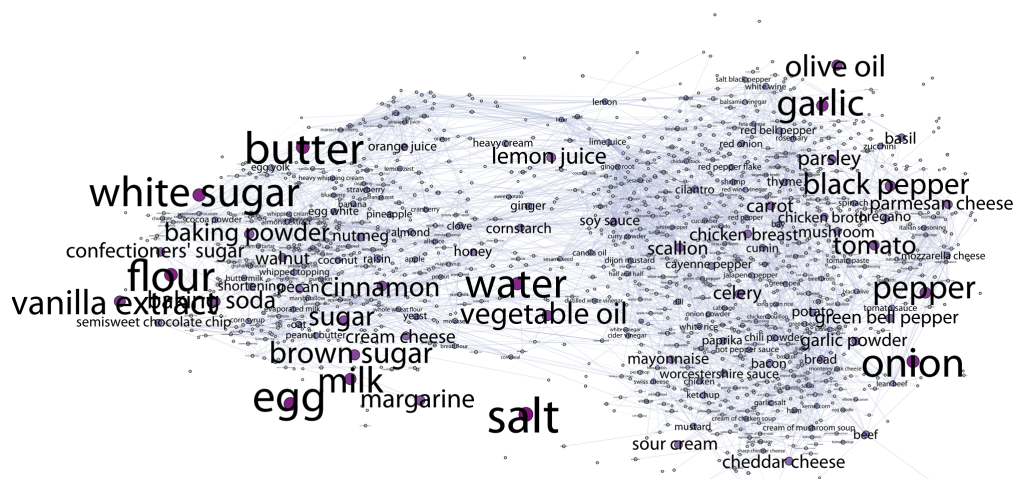


Abbildung 6: Zutatenkomplemente-Netzwerk mit PMI gewichteten Verbindungen, Quelle: Teng et al. (2012)

Nyati et al. (2019) erstellten ebenfalls Netzwerke auf Basis der Rezepte und Zutaten. Die Zutaten-Zutaten und Zutaten-Rezepte Netzwerke nutzten sie dann, um mit Hilfe des „Hyperlink-induced Topic Search“ (HITS) Algorithmus Rezeptempfehlungen zu erstellen. Im Gegensatz zu Teng et al. (2012) gewichteten sie in dem Zutaten-Zutaten Netzwerk die Verbindungen nicht mit dem PMI sondern mit dem Edge-Betweenness Centrality Maß. Das Edge-Betweenness Centrality Maß wird definiert als die Anzahl der kürzesten Wege, die mit einer Verbindung (Edge) in einem Netzwerk gebildet werden können. Jeder Verbindung im Netzwerk kann ein Wert für die Edge-Betweenness Centrality zugeordnet werden. Eine Kante mit einem hohen Edge-Betweenness Centrality Wert stellt eine brückenartige Verbindung zwischen unterschiedlichen Teilen eines Netzwerks dar.

Eine weitere Methode, Empfehlungen aus Netzwerken zu generieren, basiert auf dem node2vec Algorithmus von Grover and Leskovec (2016). In einer Arbeit von Ruis (2019) wurde dieser Algorithmus genutzt, um aus solchen, oben beschriebenen Zutatenkomplement-Netzwerken Empfehlungen herzuleiten. Der node2vec Algorithmus erstellt dabei Word

Embeddings, die wie in Kapitel 3.3 beschrieben für die Generierung von Empfehlungen verwendet werden können. Dabei werden die Word Embeddings selbst auf die gleiche Weise erlernt wie die Embeddings von word2vec - unter Verwendung eines Skip-Gramm-Modells - doch der Kontext, der für die Erlernung der Embeddings genutzt wird, entsteht aus dem Netzwerk. Dies geschieht durch Generierung von Random Walks ausgehend von jedem Zutaten-Knoten des Graphen. Dabei regelt die Gewichtung der Verbindungen mit dem PMI, wie häufig welche Zutatenkombinationen generiert werden (Grover and Leskovec, 2016).

## 4 Anwendung

### 4.1 Deskriptive Auswertung

Der Rezepte-Datensatz wurde über die API von chefkoch.de via Webscraping erstellt. Chefkoch.de ist eine deutsche Website mit 4,1 Millionen Nutzern (Stand 2019). Auf dieser Website können Rezepte mit Beschreibungen hochgeladen und dann mit anderen Website-Besuchern geteilt werden. Die Rezeptedatenbank enthält über 330.000 Rezepte, sowie über 70 Foren rund ums Kochen und Backen (Quelle: <https://de.wikipedia.org/wiki/Chefkoch.de>).

Für meine Anwendung verwende ich einen Teildatensatz von 139.337 Rezepten und 3.371 verschiedenen Zutaten vor der Bereinigung. Der Code für das Webscraping wurde mit den R Paketen httr und jsonlite erstellt. Es wurden nur Rezepte mit dem Tag „Hauptspeise“ heruntergeladen. In dem ersten Schritt wurden die IDs aus den Rezeptelisten der API besorgt. Die IDs der Rezepte wurden dann benutzt, um weitere Variablen zu den Rezepten, wie „title“, „difficulty“, „preparationTime“, „cookingTime“, „totalTime“, „biteUrl“, „owner“, „instructions“, „rating“, „numVotes“, „tags“, „categoryIds“, sowie die in den Rezepten verwendeten Zutaten und deren Produktgruppen zu erhalten.

Bei der Datenaufbereitung der Zutaten wurden die Namen der Zutaten so gekürzt, dass alles, was hinter einem Komma und einer Klammer stand, für die einfachere Verwendung in den Algorithmen wegfiel. Beispielsweise ändert sich dadurch „Paprika, rot“ zu „Paprika“ und „Gurke(n)“ zu „Gurke“. Dadurch ist zwar mit einem Informationsverlust zu rechnen, jedoch erhöht sich trotz dessen die Genauigkeit der Modelle insgesamt, da hiermit zum Beispiel „Gurke(n)“ und „Gurke“ nicht mehr als zwei verschiedene Zutaten gezählt werden.

Weiterhin wurde die sehr häufig vorkommende Zutat „Salz und Pfeffer“ auf die zwei Zutaten „Salz“ und „Pfeffer“, welche ebenfalls im Datensatz enthalten waren, aufgeteilt. Durch diese Bereinigungen verringerten sich die Zahl unterschiedlicher Zutaten bereits von 3.371 auf 3.237.

Die Verteilung der Zutaten scheint, wie üblich für die Verteilung von Worthäufigkeiten,

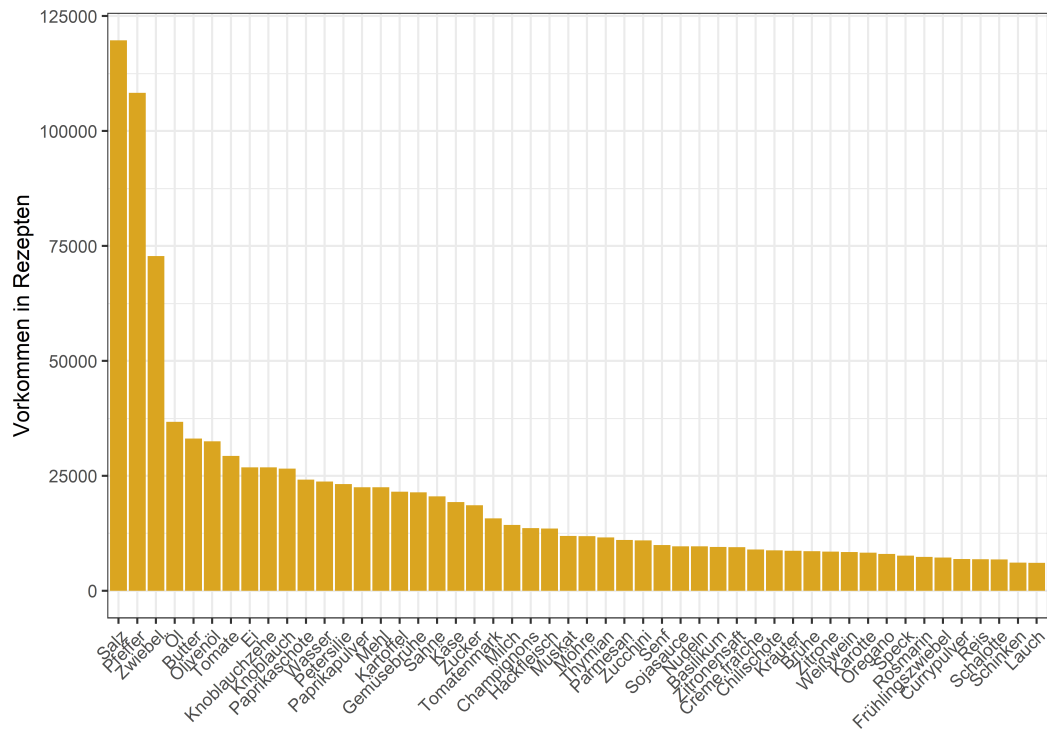


Abbildung 7: Die 50 meistverwendeten Zutaten in Rezepten; Quelle: Eigene Berechnung

einer Zipf-Verteilung zu folgen, wobei eine kleine Anzahl von Zutaten extrem häufig, die große Mehrheit jedoch weit weniger häufig vorkommt. Es gibt eine lange Reihe von Zutaten, die nur ein- oder zweimal verwendet werden, häufig weil sie entweder exotisch sind oder Schreibfehler enthalten. Für eine bessere Übersichtlichkeit werden in Abbildung 7 nur die häufigsten 50 Zutaten betrachtet. Hier ist zu sehen, dass Salz in fast allen Rezepten enthalten ist, dicht gefolgt von dem Gewürz Pfeffer. Weiterhin finden sich verschiedene Öle und Butter ganz oben in der Liste wieder. Von den Gemüsesorten wird die Zwiebel am häufigsten für Rezepte verwendet, sie kommt in knapp 75.000 Rezepten vor, während die Tomate mit weniger als halb so vielen Vorkommnissen die zweit meist verwendete Gemüseart ist. Unter den am häufigsten vorkommenden Zutaten sind auch allgemein Gewürze sowie Knoblauch zu finden. Interessanterweise gehören auch Eier in die Top 10 der häufigsten Zutaten und wie oben erwähnt auch die Butter.



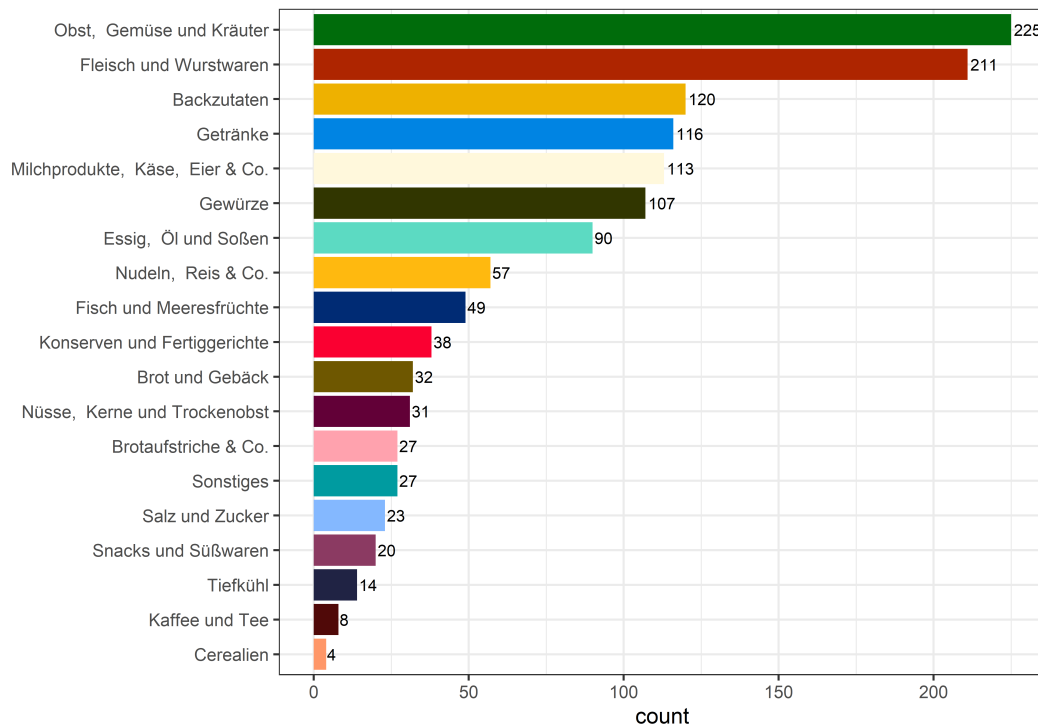


Abbildung 8: Zutatenverteilung nach Produktgruppen; Quelle: Eigene Berechnung

Die meisten Zutaten sind bereits zu Produktgruppen zugeordnet. Zutaten ohne Produktgruppen sind hauptsächlich seltene Zutaten, die jedoch den Großteil der Zutatenliste ausmachen. Da die Produktgruppen nur für deskriptive Zwecke von mir genutzt werden, stellen die fehlenden Einträge keine größeren Probleme dar. Abbildung 8 zeigt die Verteilung der Zutaten nach Produktgruppen, für welche diese Einträge nicht fehlten. Insgesamt gibt es 19 verschiedene Produktgruppen in dem Datensatz. Von den Hauptspeisenrezepten sind die meisten Zutaten, die einer Produktgruppe zugeordnet wurden, Obst, Gemüse und Kräuter zugehörig. Die wenigsten Zutaten gehören der Gruppe Cerealien an. Interessant ist die relative Häufigkeit der Gruppe Getränke, jedoch ist dies damit zu begründen, dass auch beispielsweise Kochwasser oder auch Sojamilch zu den Zutaten in der Gruppe Getränke zählt. Die vorliegende Zuteilung zu Produktgruppen ist jedoch mit Vorsicht zu genießen, da z.B. Gewürze oft aus Kräutern bestehen, diese jedoch nicht in derselben Gruppe sind. Eine genaue Überprüfung dieser Produktgruppen wäre dementsprechend für Folgeprojekte, die diese auch für Vorhersagezwecke nutzen wollen, notwendig. In Abbildung 9 sieht man die Verteilung der verwendeten Zutaten pro Rezept. Die meisten Rezepte benutzen etwa 11 Zutaten. Die Verteilung sieht größtenteils symmetrisch aus, es gibt jedoch ein paar Ausreißer Rezepte mit sehr vielen Zutaten, weshalb die X-Achse in der Graphik bis 40 reicht.

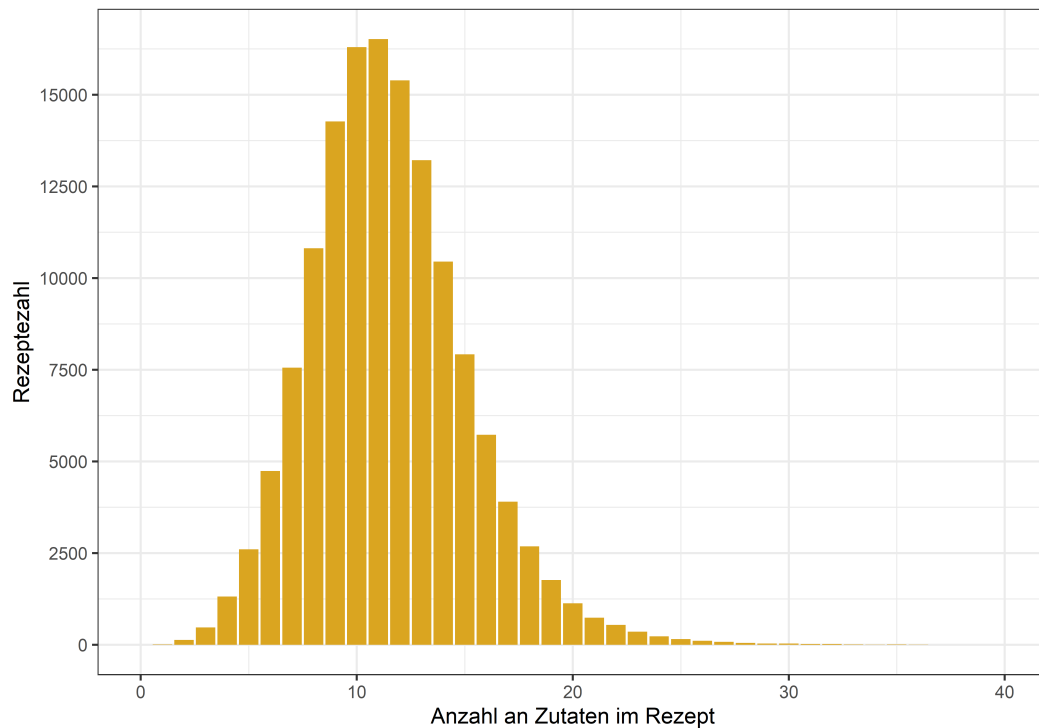


Abbildung 9: Verteilung der Anzahl an Zutaten in Rezepten; Quelle: Eigene Berechnung

Für eine bessere Darstellung wurde ein Rezept mit 64 Zutaten aus dem Datensatz für die Abbildung entfernt. Wie in der Abbildung zu sehen, gibt es auch ein paar wenige Rezepte mit einer und einige Rezepte mit zwei oder drei Zutaten. Die Rezepte mit einer Zutat sind meistens Gemüse und Kartoffelzubereitungen im Ofen oder Fleischzubereitungen.

Für die Empfehlung von Rezepten werden in der Shiny-App anhand einer Vorauswahl an Zutaten weitere Zutaten vorgeschlagen, die mittels der unterschiedlichen Empfehlungsalgorithmen gefunden werden. Dies wird in den nächsten Kapiteln detaillierter erläutert. Die ausgewählten Zutaten werden dann verwendet, um Rezepte zu identifizieren. Alle Rezepte, die die ausgewählten Zutaten enthalten, werden in der Shiny-App neben den Zutatenempfehlungen angezeigt, geordnet nach dem Bewertungsscore und der Anzahl der Bewertungen. Im Folgenden wird die genaue Implementierung der einzelnen Algorithmen und Methoden beschrieben und dabei insbesondere die Vor- und Nachteile sowie resultierenden Probleme, die mit der Implementierung verbunden sind, erläutert. Es wird jeweils detailliert beschrieben, wie für einen gegebenen Empfehlungsalgorithmus eine konkrete Empfehlung generiert wird. Desweiteren können mit den unterschiedlichen Methoden auch unterschiedliche deskriptive Analysen des Datensatzes durchgeführt werden.

## 4.2 Association Rules

Für die Implementierung des Apriori Algorithmus und Generierung der Association Rules wurde das R Package `arules` (<https://cran.r-project.org/web/packages/arules/index.html>, zuletzt aufgerufen: 24.7.2020) verwendet. Es sei auf die Publikation von Hornik et al. (2005) verwiesen, in dem sämtliche Funktionen des Package ausführlich erklärt werden. Die Daten wurden für die Anwendung in eine Document-Term-Matrix gebracht, bei der die Zeilen die Rezepte (Documents) repräsentieren, die Spalten die Zutaten (Terms) darstellen und Vorkommnisse von Zutaten in Rezepten mit TRUE/FALSE codiert sind. Nachdem diese Matrix anschließend in die Transaktionsklasse, welche `arules` verwendet, umgewandelt wird, ermöglicht `arules` einfache Ausgaben von den häufigsten Zutaten sowie dem gemeinsamen Auftreten von Zutaten. Die `summary()` Funktion gibt Aufschlüsse, wie die Anzahl an verwendeten Zutaten pro Rezept im Durchschnitt und im Median, Mindest- und Maximalzahl, sowie die Verteilung dieser Größen. Die Funktion berechnet auch die Dichte an Zutaten Vorkommnissen, welche zusammenfasst, wie viele Nullen und Einsen die Matrix enthält. Abbildung 10 zeigt diese Werte für den Datensatz in der Arbeit. Die Dichte beträgt lediglich 0.00355. Der Median liegt bei 11 Zutaten pro Rezept und das arithmetische Mittel bei 11.5 Zutaten pro Rezept. Es existieren 9 Rezepte, die nur eine Zutat verwenden, und ein Rezept mit insgesamt 64.

Die Apriori Regeln, mit denen die Empfehlungen generiert werden, können anschließend,

```
transactions as itemMatrix in sparse format with
 139337 rows (elements/itemsets/transactions) and
 3237 columns (items) and a density of 0.003550489

most frequent items:
  Salz Pfeffer Zwiebel      Öl Olivenöl (other)
 112551 104757 71054 35289 30773 1246967

element (itemset/transaction) length distribution:
sizes
 1   2   3   4   5   6   7   8   9   10  11  12  13  14
13  135 472 1314 2605 4737 7558 10811 14272 16298 16514 15392 13212 10451
15  16  17  18  19  20  21  22  23  24  25  26  27  28
7922 5724 3905 2684 1767 1132 737 542 358 231 158 108 81 49
29  30  31  32  33  34  35  36  38  39  40  45  64
37  34  21  20  14  8  10  3  2  2  2  1  1

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.00   9.00   11.00   11.49  14.00   64.00
```

Abbildung 10: Zusammenfassende Maße für Rezepte; Quelle: Eigene Berechnung

unter Angabe eines Mindest-Support und Confidence Levels, berechnet werden. Diese Mindestlevels müssen angegeben werden, da ansonsten die Berechnungen ins Unendliche

gehen. Der Algorithmus würde für jede mögliche Kombination von  $n$  Zutaten die drei oben erklärten Kennzahlen Support, Confidence und Lift berechnen. Bei einer Zutatenanzahl von 3.289 führt dies zu einer Anzahl an Kombinationen deren Berechnung mit der zur Verfügung stehenden Rechenkapazität nicht möglich ist. Aus diesem Grund müssen die Mindestwerte für Support und Confidence festgelegt werden. Leider gibt es für diese Werte bisher keine theoretischen Überlegungen, wie hoch sie am besten gesetzt werden sollen. Normalerweise werden sie mit Abwägung der Datengröße und Rechenkapazität durch Ausprobieren festgelegt. In dieser Arbeit werden für den Support 0.001 und für die Confidence 0.02 als Mindestwerte gewählt. Mit diesen Werten erstellt der Apriori Algorithmus 485.368 Regeln. Sobald die Regeln einmal erstellt sind, müssen diese für die Empfehlung in der Shiny-App nicht neu generiert werden. Die Empfehlungen können ganz einfach je nach Zutatenkombination aus den bestehenden abgespeicherten Regeln abgerufen werden. Da jedoch ein Mindest Support und Confidence Level festgelegt ist, kann dieser Algorithmus im Gegensatz zu den folgenden Modellen nicht für jede Zutatenkombination, die von dem Nutzer ausgewählt wird, eine Empfehlung für eine weitere Zutat ausgeben. In dem Fall wird für die Empfehlungsausgabe die nächstbeste passende Regel hergenommen. Abbildung 1 und 2 zeigen ein paar ausgewählte Regeln mit dem höchsten Lift und anschließend mit der höchsten Confidence. Während die Regeln mit dem höchsten Lift zwar etwas außergewöhnliche Zutaten enthalten, empfehlen die Regeln mit der höchsten Confidence fast immer nur Salz oder Pfeffer, da diese Zutaten in fast allen Rezepten vorkommen und hier nicht für die allgemeine Popularität der Zutaten kontrolliert wird. Aus diesem Grund werden für die Zutaten Empfehlungen in der Shiny-App die Regeln nach dem höchsten Lift sortiert. Auf Basis der vom Nutzer ausgewählten Zutaten, welche die LHS Seite der Regeln darstellen, werden dann die Zutaten auf den ersten fünf Plätzen mit dem höchsten Lift dem Nutzer als weitere passende Zutat empfohlen. Grundsätzlich gibt es auch noch weitere interessante Maße, die für die Auswahl von Assoziationsregeln und Empfehlungen hergenommen werden können, wie zum Beispiel die all-confidence. In dieser Arbeit wurden keine weiteren Maße betrachtet, es sei jedoch auf die Publikationen von Lenca et al. (2008) und von Zhang et al. (2009) verwiesen für den Vergleich von weiteren Maßen.

lhs	rhs	support	confidence	lift	count
{Chilischote,Peperoni}	=>{Blüten.und.Blätter}	0.001062173	0.5362319	223.03565	148
{Knoblauchzehe,Peperoni}	=>{Blüten.und.Blätter}	0.001270301	0.1921824	79.93469	177
{Chilischote,Karotte}	=>{Blüten.und.Blätter}	0.001033466	0.1907285	79.32995	144
{Palmzucker}	=>{Fischsauce}	0.001212887	0.6550388	71.69767	169
{Fischsauce}	=>{Palmzucker}	0.001212887	0.1327573	71.69767	169
{Kaffir.Limettenblätter}	=>{Zitronengras}	0.001241594	0.4402036	64.63292	173
{Zitronengras}	=>{Kaffir.Limettenblätter}	0.001241594	0.1822972	64.63292	173
{Zwiebelpulver}	=>{Knoblauchpulver}	0.001155472	0.5689046	61.64033	161
{Knoblauchpulver}	=>{Zwiebelpulver}	0.001155472	0.1251944	61.64033	161
{Kokosmilch,Chilischote}	=>{Kaffir.Limettenblätter}	0.001112411	0.1519608	53.87725	155

Tabelle 1: Die 10 Regeln mit dem größten Lift; Quelle: Eigene Berechnung

lhs	rhs	support	confidence	lift	count
Öl,Zucker,Hefe	=> Salz	0.001198533	1	1.237990	148
Pfeffer,Mehl,Gehacktes	=> Salz	0.001040642	1	1.237990	145
Zwiebel,Salz,Tomate,Cannelloni	=> Pfeffer	0.001019112	1	1.330097	142
Mehl,Öl,Zucker,Hefe	=> Salz	0.001062173	1	1.237990	148
Zwiebel,Pfeffer,Zucker,Hefe	=> Salz	0.001392308	1	1.237990	194

Tabelle 2: Die 5 Regeln mit der größten Confidence; Quelle: Eigene Berechnung

### 4.3 Word Embeddings

Für die Implementierung des GloVe Algorithmus in R wurde das *text2vec* Package (<https://cran.r-project.org/web/packages/text2vec/index.html>, zuletzt aufgerufen am 24.7.2020) verwendet und für word2vec das *word2vec* Package (<https://cran.r-project.org/web/packages/word2vec/index.html>, zuletzt aufgerufen am 24.7.2020). Der GloVe Algorithmus benötigt die Daten in einem Term-Co-Occurrence Format, bei dem Zeilen und Spalten jeweils Zutaten sind und die Felder angeben, wie oft die Zutaten zusammen in Rezepten vorkommen. Eine Reihe von Parametern wie Learning Rate, Initialisierungswerte oder Gewichtungparameter können angegeben werden. Für die meisten sind bereits sinnvolle Standardwerte, die auch in der Original Publikation von Pennington et al. (2014) vorgeschlagen wurden, vorgemerkt. Wichtig für die vorliegende Anwendung war die Anpassung der Dimensionsgröße der Wortvektoren, sowie  $x_{max}$ , welches die maximale Anzahl an Kookkurrenzen angibt, die in der Gewichtungsfunktion verwendet werden (Pennington et al., 2014). Als Startwerte werden für die Wortvektor Dimensionen 50 und für  $x_{max}$  100 verwendet. Da eine automatische Evaluierung der Embeddings schwierig ist (siehe Kapitel 5), werden die besten Werte für die Wortdimensionen und  $x_{max}$  subjektiv nach Überprüfen der ausgegebenen Empfehlungen ausgewählt.

Die Funktionen bei der Implementierung der word2vec Algorithmen benötigen Daten, bei

denen pro Zeile ein Rezept abgebildet wird und dann in einer einzigen Spalte alle Zutaten getrennt durch Leerzeichen oder Komma aufgelistet sind. Bei den word2vec Algorithmen können ebenfalls eine Reihe von Parametern festgelegt werden. Für die Wortvektor Dimensionen wird hier ebenfalls 50 benutzt. Bei CBOW und Skip-Gram ist das Kontextfenster ein wichtiger Hyperparameter. Als Startwert wurde für beide Algorithmen die durchschnittliche Anzahl an Zutaten in den Rezepten verwendet, d.h. 11. Diese Auswahl an Kontextfenstergrößen stellt tatsächlich auch eines der Probleme bei den word2vec Algorithmen. Natürlicherweise wären bei Rezepten die Kontextgrößen variabel an die Rezeptgröße angepasst. Dies ist jedoch leider bei diesen word2vec Implementierungen in R nicht von vornherein möglich. Der Code müsste hierfür zuerst angepasst werden, wie Sauer et al. (2020) es in ihrem Paper beschreiben. Weiterhin kann hier spezifiziert werden, ob das Modell mit hierarchical softmax oder negative sampling trainiert werden soll. Der Autor des Packages gibt an, dass hierarchical softmax für seltene Wörter besser geeignet ist und negative sampling bei häufigen Wörtern und mit wenigen Dimensionen besser funktioniert. Deswegen wird hier hierarchical softmax ausgewählt. Zum Schluss kann noch eine Subsampling Rate angegeben werden, bei der dann nur ein Teil der sehr häufig vorkommenden Wörter für das Training der Modelle verwendet wird. Dies kann bei großen Datensätzen die Schnelligkeit der Berechnungen erhöhen und auch die Genauigkeit von selten vorkommenden Wörter verbessern. Für die Ausgangsmodelle wird diese Samplingrate nicht benutzt, da der Datensatz im Vergleich zu den üblichen NLP-Datensätzen eher klein ist.

Bei beiden bzw. allen drei Algorithmen erhält man am Ende Vektoren mit jeweils 50 Dimensionen für jedes Wort. Die Zutatenempfehlungen werden generiert, indem das Kosinus Ähnlichkeitsmaß zwischen zwei Vektoren berechnet wird. Dieses Ähnlichkeitsmaß misst die Distanz zwischen den Vektoren und gibt dann einen Score aus, der nach der Größe sortiert für die Empfehlung weiterer passender Zutaten verwendet werden kann. Eine Publikation von Aggarwal et al. (2001) zeigt warum die Verwendung von normalen p-Normen bei hochdimensionalen Daten nicht sinnvoll ist. Aus diesem Grund wird die Kosinus-Similarität, wie eben beschrieben, verwendet. Die resultierende Ähnlichkeit reicht von -1, was genau entgegengesetzt bedeutet, bis 1, was genau gleich bedeutet. Ein Maß



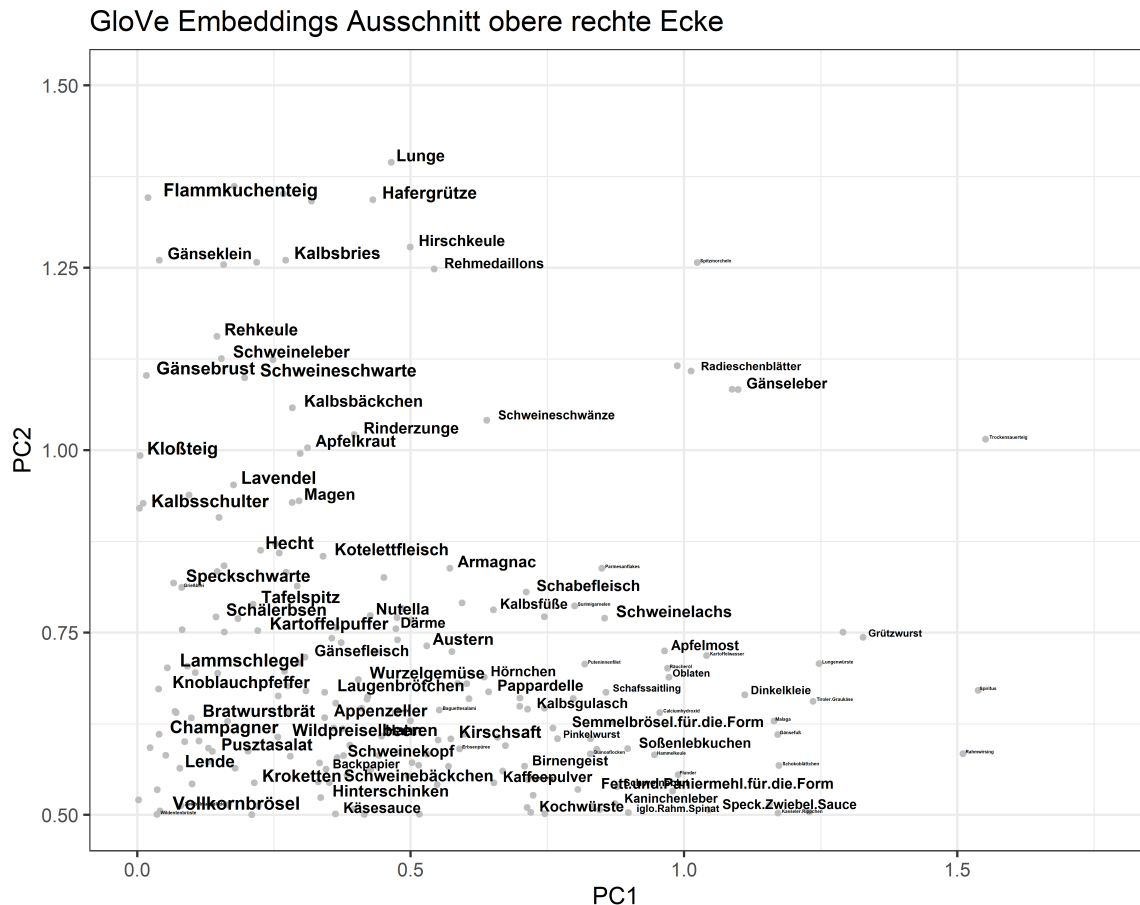


Abbildung 12: Ausschnitt aus Abbildung 11 im Bereich 0 bis 1.75 für PCA1 und 0.5 bis 1.5 für PCA2, Quelle: Eigene Berechnung

siehe z.B. Jolliffe (2002). Abbildung 11, 13 und 14 zeigen jeweils die auf zwei Dimensionen reduzierten Wortvektoren von GloVe, CBoW und Skip-Gram. Für die bessere Übersichtlichkeit werden bei allen drei Abbildungen die Größe der Zutatennamen durch ihr logarithmisches Vorkommen bestimmt. Somit erscheinen die Namen, wenn die Zutat häufiger in Rezepten verwendet wird, größer, während seltene Zutaten nur kleine Namen in den Abbildungen haben. Die X-Achse zeigt Hauptkomponente 1 und die Y-Achse zeigt die Werte in der zweiten Hauptkomponente der Zutaten. Außerdem wird in der Graphik nur ein kleiner Anteil der Zutatennamen zu ihren Punkten angezeigt, um bessere Lesbarkeit zu gewährleisten. Weiterhin findet sich in Abbildung 12 ein kleiner Ausschnitt der Graphik mit PCA1 Werten von 0 bis 1,75 und PCA2 Werten von 0,5 und 1,5, um hier ebenfalls nochmal einen Teil der Embeddings besser erkennen zu können.

Bei der Graphik für die GloVe Wortvektoren ist erkennbar, dass häufiger verwendete Zutaten eher negative Werte in der ersten Hauptkomponente haben, während seltene Zuta-





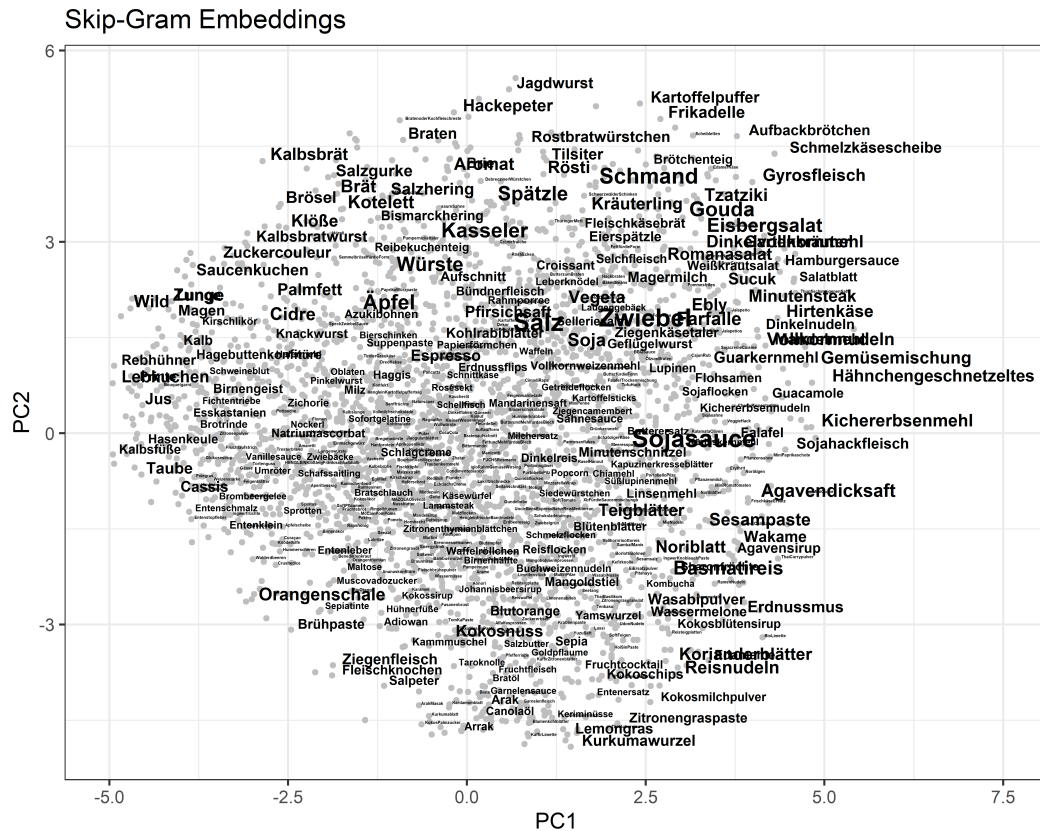


Abbildung 14: Word Embeddings des Skip-Gram Algorithmus für Zutaten, heruntergebrochen auf 2 Dimensionen mittels Hauptkomponentenanalyse; Quelle: Eigene Berechnung

Rand sammeln. Bei GloVe war dahingegen eine rechts-links Spaltung zwischen häufigen und nicht häufigen Zutaten erkennbar. Aber auch bei CBOV und Skip-Gram haben sich im oberen Bereich Fleisch-Zutaten und dazu passende Zutaten wie Bratensauce angesammelt. Eine “Asia-Ecke” gibt es diesmal jedoch nicht. Grundsätzlich scheint der GloVe Algorithmus zumindest in der deskriptiven Untersuchung einen besseren Job bei der Ordnung der Zutaten zu machen. Man muss jedoch beachten, dass bei der Hauptkomponentenanalyse die 50 Dimensionen der Word Embeddings auf lediglich zwei heruntergebrochen werden und damit viel Information verloren gehen kann.

#### 4.4 Netzwerk Modelle

In dem vorliegenden Anwendungsfall können verschiedene Arten von Netzwerken betrachtet werden. Zum einen ist es möglich, die Verbindungen zwischen Zutaten zu untersuchen, wobei die einzelnen Zutaten die Knotenpunkte sind und zum Beispiel gemeinsames Vorkommen in Rezepten die Verbindungen zwischen den Knoten sein können.

nen. Weiterhin kann auch die Beziehung von Rezept zu Zutat als Netzwerk dargestellt werden. Für die Erstellung von Netzwerken in R wird das Paket *igraph* verwendet (<https://cran.r-project.org/web/packages/igraph/index.html>, zuletzt aufgerufen am 24.7.2020). Hiermit lassen sich die Daten im Adjazenzmatrix-Format, bzw. wie oben genannt Term-Count-Format, in ein Graph-Daten Objekt oder auch in eine Edgelist, bei der jede Zeile eine Verbindung (Edge) zwischen zwei Zutaten in Von und Zu Spalten darstellt, verwandeln. Das Paket bietet auch Funktionen zur einfachen Berechnung von deskriptiven Netzwerk Maßen und auch verschiedene Plot-Methoden für Netzwerke an. Das Zutaten-Zutaten Netzwerk lässt sich mit folgenden Maßen beschreiben (Kolaczyk and Csárdi, 2014): Die Transitivität des Netzwerkes, welche die relative Häufigkeit zusammenfasst, mit der sich drei paarweise verbundene Zutaten zu einem Dreieck zusammenbilden, beträgt 0,37. Diese Kennzahl ist normalerweise für soziale Netzwerk wichtig, da sie zum Beispiel Hinweise darauf geben könnte, wie wahrscheinlich es ist, dass bei zwei Freunden ein dritter Befreundeter der zweiten Person sich auch mit der ersten Person anfreundet. In dieser Anwendung würde ein hoher Transitivitätswert bedeuten, dass wenn zwei Zutaten 1 und 2 zusammen in einem Rezept vorkommen und eine dritte Zutat 3 mit der ersten Zutat 1 in einem Rezept vorkommt, Zutat Nummer 2 und 3 wahrscheinlich auch gut zusammen in ein Rezept passen würden. Als ein weiteres Netzwerk-Maß kann der Durchmesser oder Durchmesser eines Netzes als der längste aller berechneten kürzesten Wege in einem Netz definiert werden. Er ist der kürzeste Abstand zwischen den beiden am weitesten entfernten Knoten im Netzwerk. Mit anderen Worten, sobald die kürzeste Weglänge von jeder Zutat zu allen anderen Zutaten berechnet ist, ist der Durchmesser der längste aller berechneten Weglängen. Im vorliegenden Zutaten-Zutaten Netz beträgt er 4. Die mittlere Weg-Distanz zwischen Zutaten beträgt dabei 1,94. D.h. die Zutaten sind recht häufig miteinander verbunden und es gibt keine Gruppen an Zutaten, die nur entfernt miteinander verbunden sind. Das Plotten eines Netzwerkes ist mit sehr viel Rechenkapazität verbunden und wirkt schnell überladen bei zu vielen Verbindungen und Endpunkten, so dass man mit dem bloßen Auge keine relevanten Erkenntnisse mehr gewinnen kann. In diesem Fall wäre eine Netzwerk-Abbildung mit 3.237 Zutaten und 1.473.704 Verbindungen ebenfalls zu unübersichtlich.

Dabei machen es die schiere Größe und der vielschichtige Charakter der Netzwerke sehr schwierig, die relevanten Informationen zu extrahieren, die eine reduzierte Darstellung

unter Beibehaltung der Schlüsselmerkmale ermöglichen würden. Bei einem statischen Grenzwert bei dem Verbindungen, die darunter fallen einfach weggelassen werden würden, könnten kleine aber wichtige Wechselwirkungen herausfallen. Aus diesem Grund verwendeten Ahn et al. (2011) die von Serrano et al. (2009) in der Publikation „Extracting the multiscale backbone of complex weighted networks“ vorgeschlagene Methode des *disparity filters*, um relevante Zutaten Verbindungen zu extrahieren und weniger relevante zu vernachlässigen, so dass eine Abbildung des Netzwerkes am Ende übersichtlicher wird. Die Idee dahinter ist die Beibehaltung der Verbindungen, die statistisch signifikante Abweichungen gegenüber einem Nullmodell darstellen. Diese Methode, die auf der lokalen Identifizierung der statistisch relevanten Verbindungen basiert, ist in der Lage, das Rückgrat der dominanten Verbindungen in gewichteten Netzwerken herauszufiltern, wobei strukturelle Eigenschaften und Hierarchien auf allen Skalen erhalten bleiben.

Für die Anwendung wurde der Datensatz zuerst auf die 500 meist verwendeten Zutaten reduziert und anschließend der *disparity filter* mit  $\alpha = 0.05$  verwendet, um nur die relevanten Verbindungen beizubehalten. Dies wurde in R mit dem Package *disparityfilter* und weiteren Anpassungen durchgeführt. Es verblieben weiterhin 500 Zutaten und 29.760 Verbindungen von den ursprünglich 413.496 Verbindungen. Abbildung 15 zeigt das Ergebnis dieser Filterung. Dabei sind die Verbindungen zwischen den Zutaten mit dem oben beschriebenen PMI gewichtet und die Größe der Zutatennamen werden durch ihr logarithmisches Vorkommen bestimmt. Man erkennt, dass Salz, Pfeffer und Zwiebel an den Rand gedrängt werden, während Zutaten, die zum Würzen benutzt werden können, wie Knoblauch, Paprikapulver oder Muskat, sowie auch Öle und Butter in der Mitte des Netzwerkes bleiben. In dieser reduzierten Darstellung lassen sich aber keine bestimmten Cluster an Zutaten erkennen.

Um aus diesen Netzwerken Zutaten Empfehlungen ableiten zu können verwende ich die in Kapitel 3.4 beschriebene Methode des *node2vec* Algorithmus (Grover and Leskovec, 2016). Als Basis wird dafür das Zutaten-Komplementenetzwerk verwendet, bei dem Verbindungen zwischen Zutaten mit dem in Kapitel 3.4 beschriebenen PMI-Wert gewichtet werden. Der Algorithmus erstellt mithilfe des Netzwerkes neue Zutatenkontexte für das Training von Word Embeddings. Hierfür wird ein Skip-Gram Modell mit denselben Konfigurationen, wie das in Kapitel 4.3 beschriebene Modell, mit 50 Dimensionen und einem Kontextfenster von 11 verwendet. Die Embeddings können anschließend wie im vorherigen Abschnitt

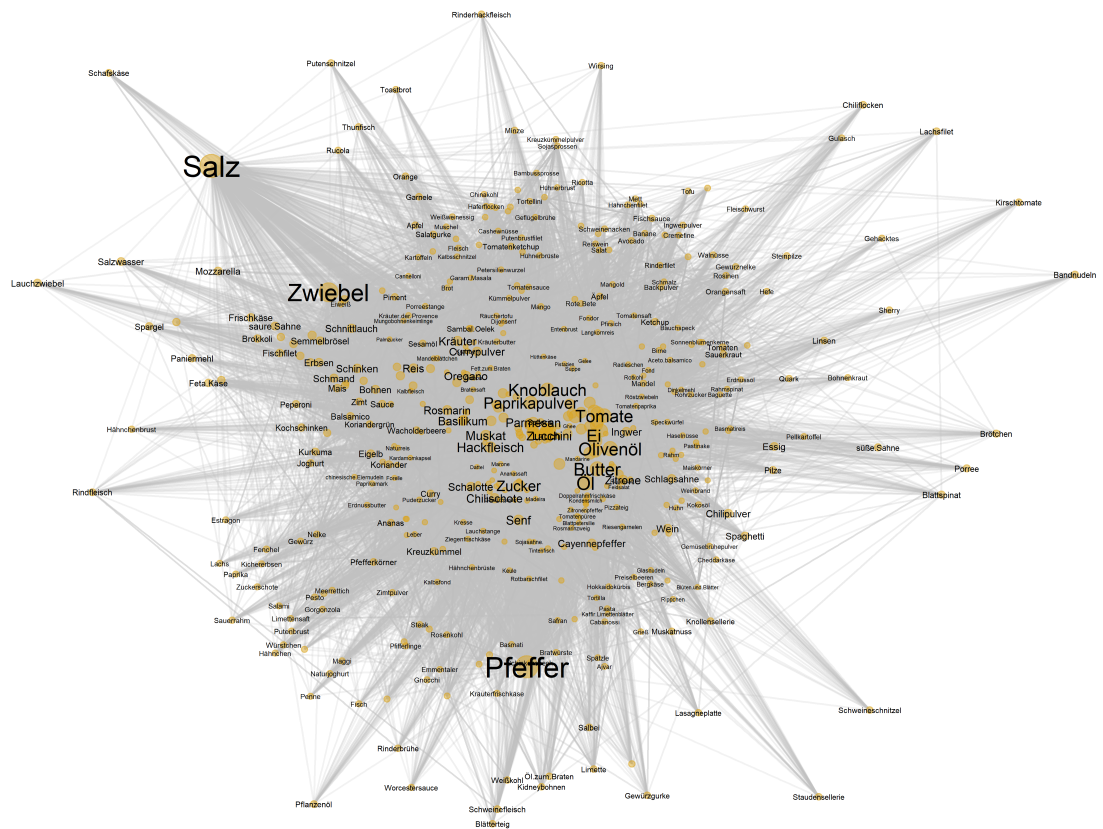


Abbildung 15: Mit dem disparity filter reduziertes Zutaten-Zutaten Netzwerk mit PMI gewichteten Verbindungen; Quelle: Eigene Berechnungen

erklärt wieder für die Empfehlungen weiterer Zutaten verwendet werden. Außerdem ist es wiederum möglich die Zutatenvektoren mittels der Hauptkomponentenanalyse in zwei Dimensionen herunterzuberechnen und anschließend graphisch darzustellen. Abbildung 16 zeigt diese Darstellung. Pfeffer und Salz sammeln sich hier wieder in der Mitte bzw. eher rechten Seite der Wolke, während seltenere Zutaten auf der linken Seite der Wolke abgebildet werden. In der unteren Ecke lassen sich wieder eher asiatische Zutaten auffinden, jedoch ist dies nicht so stark erkennbar, wie bei den GloVe Embeddings.



sollte. Aggarwal et al. (2016) listet insgesamt vier Punkte auf, die die Güte einer Empfehlung kennzeichnen. Das ist zum Ersten die *Relevanz* der Empfehlung für den Nutzer, welche selbsterklärend ist. Weiterhin spielt aber auch die *Neuartigkeit* der Empfehlung eine Rolle. In dem Kontext der Zutaten-Empfehlung könnte dies eine Zutat sein, die nicht zu den standardmäßig verwendeten Zutaten gehört und damit das Rezept außergewöhnlicher macht. Für die Rezepte-Empfehlung könnte dies wiederum bedeuten, nicht zum wiederholten Mal die Lasagne vorzuschlagen, nachdem der Nutzer diese bereits einmal gekocht hat. Als dritte Eigenschaft nennt Aggarwal *Serendipity*, welche sich eventuell mit *Originalität* übersetzen lässt. Sie lässt sich mit einem glücklichen Zufall beschreiben, bei dem ähnlich zu der Neuartigkeit der Empfehlungen nicht nur das direkt offensichtliche empfohlen werden sollte, sondern auch mal für den Nutzer unerwartete Empfehlungen zu einem besseren Ergebnis führen können. Der vierte und damit letztgenannte Punkt von Aggarwal dreht sich um die *Diversität* der vorgeschlagenen Elemente. Wenn sich alle Empfehlungen zu sehr ähneln, könnte dies dazu führen, dass dem Nutzer keines der Elemente gefällt. Dementsprechend sollten sich zum Beispiel die Top 10 der empfohlenen Zutaten voneinander unterscheiden und nicht beispielsweise nur unterschiedliche Käsesorten enthalten. All diese Kriterien wären mit der oben beschriebenen rein maschinellen Evaluierung nicht eindeutig testbar. Besser wäre dementsprechend eine menschliche Bewertung der Empfehlungen, wofür jedoch zuerst die passenden Daten erhoben werden müssten. Aus diesen Gründen wird hier keine maschinelle Evaluierung durchgeführt. Für künftige Arbeiten könnte zusätzlich untersucht werden, wie die oben genannten Kriterien eventuell doch für die automatisierte Evaluierung der Empfehlungen maschinell übersetzt werden können.

Nichtsdestotrotz können die Empfehlungen anhand ausgewählter Beispiele verglichen werden. Betrachtet man als Ausgangszutat *Avocado* werden folgende weitere Zutaten von den Algorithmen empfohlen:

Modell und Empfehlung für Avocado	1. Empfehlung	2. Empfehlung	3. Empfehlung	4. Empfehlung	5. Empfehlung
Association Rules	Paprikapulver	Koriander	Salz	Salatblätter	Knoblauch
GloVe	Limette	Tortilla	Limettensaft	Rucola	Eisbergsalat
CBOw	Romanasalat	Salat	Minze	Eisbergsalat	Römersalat
Skip-Gram	Joghurtsatz	Pita Brot	Kochsalami	Mini Gurke	Kopfsalat
Network	Lachs	Traubensirup	Thunfisch	Karottensaft	Paprikaschote

Wenn man sich diese Empfehlungen anschaut, dann lässt sich vermuten, dass die Algorithmen ihre Empfehlungen aus Rezepten wie Guacamole und Wraps nehmen. Vor allem bei



GloVe lässt sich dieser Zusammenhang erkennen. Während für Avocado die Empfehlungen bei allen Algorithmen annehmbar aussehen, ändert sich dies bei der Betrachtung von *Sellerie*. Wenn Sellerie als Ausgangszutat hergenommen wird, werden bei einigen Algorithmen größere Schwächen erkennbar:

Modell und Empfehlung für Sellerie	1. Empfehlung	2. Empfehlung	3. Empfehlung	4. Empfehlung	5. Empfehlung
Association Rules	Thymian	Limette	Tomate	Avocado	Knoblauch
GloVe	Knollensellerie	Petersilienwurzel	Möhre	Lorbeerblatt	Lauch
CBOW	Knollensellerie	Staudensellerie	Bleichsellerie	Stangensellerie	Petersilienwurzel
Skip-Gram	Petersilienwurzel	Lorbeerblatt	Knollensellerie	Liebstöckel	Bleichsellerie
Network	Beifuß	Rotwein	Gänse	Gans	Roulade

Abgesehen von den Association Rules und dem Netzwerk Modell, empfehlen die anderen Algorithmen Zutaten, die Sellerie ähneln, wie Knollensellerie oder Staudensellerie, und nicht solche Zutaten, die zu Sellerie in einem Gericht hinzupassen würden. Daraus lässt sich schließen, dass die Word Embedding Modelle hauptsächlich Substitute finden und keine Komplemente. Auch nach Anpassung der Hyperparameter änderte sich dieser Zustand nicht. Weitere Empfehlungen für andere Ausgangszutaten lassen sich mit der Shiny-App ausgeben.

## 6 Fazit und Ausblick

Zusammenfassend lässt sich sagen, dass die Word Embedding Modelle für viele Ausgangszutaten eher Substitute empfehlen als Komplemente. Damit lassen sie sich in dem aktuellen Zustand nicht uneingeschränkt für eine Rezepte-Empfehlung verwenden. Weitere Anpassungen in den Hyperparametern und der Modellkonfiguration könnten jedoch Verbesserungen mit sich bringen. Levy et al. (2015) zeigen, dass die Hyperparameter Einstellungen enorme Auswirkungen haben können und da in dieser Arbeit nicht alle Möglichkeiten implementiert werden konnten, ist es denkbar, dass eine genauere Betrachtung und Einstellung der Parameter zu einer Empfehlung von Zutatenkomplementen statt nur -substituten führen kann. Unter den Word Embedding Modellen scheint GloVe die bessere Wahl zu sein. Sowohl in der deskriptiven Analyse als auch in der stichprobenhaften Untersuchung von Empfehlungen erscheint GloVe bessere Ergebnisse zu liefern. Weiterhin war dieser Algorithmus einfach anwendbar und auch mit der großen Menge an Daten schnell berechnet.

Die Darstellung über die mit der Hauptkomponentenanalyse heruntergebrochenen Dimen-



sionen ermöglichte auch einige interessante Einblicke in die Datenbeschaffenheit. Eine Färbung der Daten nach der zugehörigen regionalen Küche, wie zum Beispiel asiatisch, mexikanisch, westlich könnte die Erkennbarkeit von Zusammenhängen nochmal erleichtern, da sich bereits eine Häufung der Zutaten nach diesen Küchen in Ansätzen gezeigt hat. Aber auch die anderen Algorithmen lieferten jeweils unterschiedliche Einblicke in die Daten. Mit den Association Rules konnten schnell und einfach Zusammenhänge abgeleitet werden. Das Netzwerk Modell war vor allem für die Darstellung der Zutatenverbindungen interessant. Außerdem bieten Netzwerk Modelle noch viele weitere Analysemaße der Daten, wie zum Beispiel einfache Clusterfindung, die hier nicht angesprochen worden sind.

In dieser Arbeit wurden nur die fünf Modelle Association Rules, GloVe, CBOW, Skip-Gram und Netzwerke miteinander verglichen, grundsätzlich gibt es jedoch noch viele andere Modelle die einer Betrachtung würdig wären, wie zum Beispiel die standardmäßig für Empfehlungen verwendete Methode der Matrix Decomposition. Außerdem sind keine Inhaltsbasierten Empfehlungen betrachtet worden. Zusätzliche Charakteristika, wie Rezeptart (asiatisch, westlich, usw.), Bewertung, Nährstoffgehalt, Kosten und Seltenheit der Zutaten oder Aufwand der Zubereitung könnten ebenfalls in die Empfehlungsgenerierung miteinbezogen werden und damit besser auf den Nutzer abgestimmte Ergebnisse liefern. Müller et al. (2012) erstellen in ihrer Publikation eine Zuordnung der auf Chefkoch.de verwendeten Zutaten zu einer Nährstoffliste vom Bundesministerium für Ernährung. Diese Zuordnung könnte verwendet werden, um die Nährstoffe der Zutaten künftig mit in die Empfehlungen einzubeziehen und damit Empfehlungen mit Berücksichtigung von Unverträglichkeiten und sportlichen oder gesundheitlichen Zielen kompatibel zu machen.

Elsweiler et al. (2017) zeigten auch, dass Bilder der Rezepte einen großen Einfluss auf die Rezeptauswahl haben. Diese sollten dementsprechend ebenfalls von der Chefkoch.de Seite via Webscraping beschafft und in das Empfehlungsmodell miteinbezogen werden. Mit neuronalen Netzen und automatischer Bilderkennung ist dies, wie Elsweiler et al. (2017) und Gao et al. (2019) gezeigt haben, möglich und sinnvoll.

Außerdem könnte eventuell eine genauere Charakterisierung der Zutaten durch Word Embeddings erreicht werden, wenn noch weitere Rezepte, nicht nur die mit dem Hauptspeise Tag versehenen, beschafft werden. Viele Rezepte, die zwar nicht als Hauptspeise klassifiziert nicht, können trotzdem einiges an Information über die Zutat und ihre Kombinations-

fähigkeit mit anderen Zutaten liefern. Außerdem sollte die Zuordnung der Produktgruppen wie in Kapitel 4 erwähnt nochmal überprüft werden, um anschließend ebenfalls mehr Informationen über die Zutaten liefern zu können. Es sind zwar einige Bereinigungen der Zutaten durchgeführt worden, nichtsdestotrotz sind noch viele Fehler in dem Datensatz enthalten, die nur schwer mittels automatischer Funktionen ausgebessert werden können und am besten händisch überprüft werden sollten. Und auch die von Aggarwal et al. (2016) genannten Punkte Relevanz, Neuartigkeit, Originalität und Diversität der Empfehlungen, welche in Kapitel 5 erklärt worden sind, sollten in die Gestaltung des Empfehlungssystems miteinbezogen und evaluiert werden. Als letzter und wichtigster Punkt ist anzumerken, dass für bessere Empfehlungen weitere Hyperparameter und Modellkonfigurationen ausprobiert werden sollten, da es hier höchst wahrscheinlich noch einiges an Spielraum gibt.

## Literatur

- Rakesh Agarwal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, pages 487–499, 1994.
- Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.
- Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.
- Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.
- Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1:196, 2011.
- George A Burdock. *Fenaroli's Handbook of Flavor Ingredients: Volume 2*, volume 2. CRC press, 2019.
- David Elsweiler, Christoph Trattner, and Morgan Harvey. Exploiting food choice biases for healthier recipe recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 575–584, 2017.
- Jill Freyne and Shlomo Berkovsky. Recommending food: Reasoning on recipes and ingredients. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 381–386. Springer, 2010.
- Xiaoyan Gao, Fuli Feng, Xiangnan He, Heyan Huang, Xinyu Guan, Chong Feng, Zhaoyan Ming, and Tat-Seng Chua. Hierarchical attention network for visually-aware food recommendation. *IEEE Transactions on Multimedia*, 22(6):1647–1659, 2019.
- Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

- Kurt Hornik, Bettina Grün, and Michael Hahsler. arules—a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15): 1–25, 2005.
- Sheena S Iyengar and Mark R Lepper. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology*, 79(6):995, 2000.
- I Jolliffe. Principal component analysis: Wiley online library. *Kaiser, HF (1960). The application of electronic computers to factor analysis*, 2002.
- Eric D Kolaczyk and Gábor Csárdi. *Statistical analysis of network data with R*, volume 65. Springer, 2014.
- Philippe Lenca, Patrick Meyer, Benoit Vaillant, and Stéphane Lallich. On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European journal of operational research*, 184(2):610–626, 2008.
- Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. A survey on food computing. *ACM Computing Surveys (CSUR)*, 52(5):1–36, 2019.
- Ole G Mouritsen, Rachel Edwards-Stuart, Yong-Yeol Ahn, and Sebastian E Ahnert. Data-driven methods for the study of food perception, preparation, consumption, and culture. *Frontiers in ICT*, 4:15, 2017.
- Manuel Müller, Morgan Harvey, David Elswiler, and Stefanie Mika. Ingredient matching to determine the nutritional properties of internet-sourced recipes. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, pages 73–80. IEEE, 2012.
- Angelos Nezis, Haris Papageorgiou, Pavlos Georgiadis, Petr Jiskra, Dimitris Pappas, and Maria Pontiki. Towards a fully personalized food recommendation tool. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, pages 1–3, 2018.

- Karin Nordström, Christian Coff, Håkan Jönsson, Lennart Nordenfelt, and Ulf Görman. Food and health: individual, cultural, or scientific matters? *Genes & nutrition*, 8(4): 357–363, 2013.
- Umang Nyati, Sneha Rawat, Devika Gupta, Niyati Aggrawal, and Anuja Arora. Characterize ingredient network for recipe suggestion. *International Journal of Information Technology*, pages 1–8, 2019.
- Timur Osadchiy, Ivan Poliakov, Patrick Olivier, Maisie Rowland, and Emma Foster. Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115:535–542, 2019.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- Frank Ruis. Spilling the beans: Food recipe popularity prediction using ingredient networks. B.S. thesis, University of Twente, 2019.
- Christopher Sauer, Alex Haigh, and Jake Rachleff. Cooking up food embeddings. 2020.
- M Ángeles Serrano, Marián Boguná, and Alessandro Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the national academy of sciences*, 106(16):6483–6488, 2009.
- Tiago Simas, Michal Ficek, Albert Diaz-Guilera, Pere Obrador, and Pablo R Rodriguez. Food-bridging: a new network construction to unveil the principles of cooking. *Frontiers in ICT*, 4:14, 2017.
- Dana M Small and Barry G Green. A proposed model of a flavor modality. In *The neural bases of multisensory processes*. CRC Press/Taylor & Francis, 2012.
- Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 298–307, 2012.
- Christoph Trattner and David Elsweiler. Food recommender systems: important contributions, challenges and future research directions. *arXiv preprint arXiv:1711.02760*, 2017.

Mayumi Ueda, Mari Takahata, and Shinsuke Nakajima. User's food preference extraction for personalized cooking recipe recommendation. In *Workshop of ISWC*, pages 98–105, 2011.

Yuejin Zhang, Lingling Zhang, Guangli Nie, and Yong Shi. A survey of interestingness measures for association rules. In *2009 International Conference on Business Intelligence and Financial Engineering*, pages 460–463. IEEE, 2009.