

KONZEPTION UND ENTWICKLUNG EINER
AUTOMATISIERTEN LÖSUNG ZUR GENERIERUNG
VON BESCHREIBUNGSTEXTEN FÜR WEIN IM
E-COMMERCE BEREICH AUF BASIS
STRUKTURIERTER PRODUKTDATEN

Masterarbeit

zur Erlangung des akademischen Grades Master of Science (M.Sc.)

vorgelegt an der Technischen Hochschule Köln (Campus Gummersbach)

im Studiengang

Medieninformatik

mit dem Schwerpunkt

Weaving the Web

ausgearbeitet von:

LARS PHILIPP TIM LOGES

Matrikelnummer: 11118655

Prüfer: Prof. Christian Noss (TH Köln)

Heiko Riffeler (Gjuce GmbH)

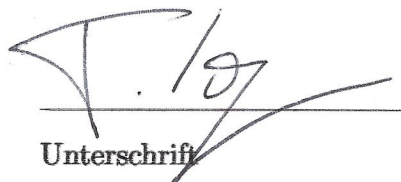
Gummersbach, 06.10.2023

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Köln, 02.10.23

Ort, Datum


Unterschrift

Abstract

This master thesis deals with the conception and development of an automated solution for generating product descriptions in the e-commerce domain, based on structured product data related to wine. This work aims to address the following questions: Which large language model is best suited for generating description texts? How can automatically generated description texts be evaluated? And how should a userinterface be designed to support automatic generation?

To answer these questions, currently available large language models were compared, and extensive research was conducted in the field of prompt engineering. Additionally, a plugin for the Shopware e-commerce system was designed and implemented, to enable the creation of description texts based on structured product data.

The analysis of the models revealed that, at the time of this work, OpenAI's large language models are best suited for the automatic generation of description texts. These are able to differentiate themselves from the competition because they produce consistent results that meet the requirements with a small amount of structured data. Due to the evolving nature of the models, the latest model (GPT-4) was used for generation.

Objective evaluation of generated description texts proved to be challenging. It has been found that no universally applicable evaluation criteria can be established for this use case. Approaches have been developed that provide a foundation for future investigations.

Kurzfassung

Die vorliegende Masterarbeit befasst sich mit der Konzeption und Entwicklung einer automatisierten Lösung zur Generierung von Beschreibungstexten im E-Commerce Bereich. Die Grundlage hierfür sind strukturierte Produktdaten über Wein. Diese Arbeit zielt darauf ab, die folgenden Fragen zu beantworten: Welches Large Language Modell eignet sich am besten für die Generierung von Beschreibungstexten? Wie lassen sich automatisch generierte Beschreibungstexte bewerten? Und wie muss ein Userinterface gestaltet sein, das eine automatische Generierung ermöglicht?

Für das Beantworten der Forschungsfragen wurden aktuell verfügbare Large Language Modells verglichen und eine intensive Recherche in der Thematik des Prompt Engineering durchgeführt. Des Weiteren ist ein Plugin für das Shopsystem Shopware konzipiert und implementiert worden, welches es ermöglicht, Beschreibungstexte anhand von strukturierten Produktdaten zu erstellen.

Die Analyse der Modelle hat ergeben, dass sich zum Zeitpunkt dieser Arbeit die Large Language Modelle von OpenAI am besten für die automatische Generierung von Beschreibungstexten eignen. Diese können sich von der Konkurrenz abgrenzen, da sie mit einer geringen Menge an strukturierten Daten konsistente und den Anforderungen entsprechende Ergebnisse liefern. Der Kurzlebigkeit der Modelle geschuldet wird das aktuellste Modell (GPT-4) für die Generierung herangezogen. Das objektive Bewerten von generierten Beschreibungstexten hat sich als herausfordernd herausgestellt. Es hat sich herausgestellt, dass für diesen Anwendungsfall keine allgemeingültigen Bewertungskriterien aufgestellt werden können. Es wurden Ansätze geschaffen, die eine Grundlage für zukünftige Untersuchungen ermöglichen.

Anmerkung

Zur besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet. Die in dieser Arbeit verwendeten Personenbezeichnungen beziehen sich auf alle Geschlechter.

Begleitrepository

Die aus dieser Arbeit resultierende Artefakte, wie das Plugin für Shopware, sind zusätzlich in einem Begleitrepository auf GitHub abgelegt:

<https://github.com/WasMachenSachen/master-thesis-tim-loges>

Inhaltsverzeichnis

Erklärung	I
Abbildungsverzeichnis	IX
Tabellenverzeichniss	X
Quellcodeverzeichniss	XI
Abkürzungsverzeichnis	XII
1. Einleitung	1
1.1. Zielsetzung	2
1.2. Kontext	2
1.3. Aufbau	3
2. Weinhandel im E-Commerce	5
2.1. Varianten des Weinhandels	5
2.2. Die Kaufphasen im E-Commerce	6
2.3. Bedeutung von Beschreibungstexten im E-Commerce	8
2.4. Redaktionsprozess bei der Integration von Produkten	9
3. Sprachmodelle und ihre Anwendung	12
3.1. Aufbau und Funktionsweise von Sprachmodellen	12
3.1.1. Lern Algorithmen	13
3.1.2. Neuronale Netze	16
3.1.3. Transformer Architektur	19
3.2. Prompt Engineering	22
3.3. Auswahl passender Sprachmodelle	25
3.3.1. Aleph Alphas Luminous Reihe	25
3.3.2. OpenAIs Generative Pretrained Transformer (GPT) Modelle	26
3.3.3. Googles Bard und PaLM 2	27
3.3.4. Metas LLaMa	27
4. Konzeption einer automatischen Lösung	30
4.1. Shopware Ecosystem	30

4.2. Stakeholder	32
4.3. Anforderungen an das Plugin	32
4.4. Visualisierung	35
4.4.1. Erweiterung durch eigenständige Seite	36
4.4.2. Erweiterung einer bestehenden Seite	37
4.4.3. Erweiterung bestehender Komponenten	37
4.5. Bearbeiten der Beschreibung	39
4.6. Architektur Entwurf	41
4.7. Datenmodell	43
5. Implementierung der automatischen Lösung	44
5.1. Vergleich der passenden Sprachmodelle	45
5.1.1. Manuelle Interaktion	47
5.1.2. Automatisierte Interaktion	49
5.1.3. Weitere Erkenntnisse aus der automatisierten Interaktion	53
5.2. Tuning des GPT-3.5-Turbo Modells	58
5.3. Beschreibung der automatischen Lösung	61
5.3.1. Implementierung der Generierung	63
5.3.2. Erweiterung der Administration	66
5.3.3. Erweiterung des Shopware WYSIWYG-Editor	68
6. Bewertung durch Experten	75
7. Fazit	78
Literaturverzeichnis	80
A. Minimale Prompt für das Erstellen einer Beschreibung von Wein	86
B. Stakeholderanalyse und Stakeholdermatrix	87
C. Beispiele aus der manuellen Interaktion	89
C.I. Beispielantwort von Bard	89
C.II. Beispielantwort LLaMa	89
D. Prompt Iterationen	90

E. Ergebnisse der automatisierten Interaktion	91
E.I. Luminous-Base-Control	92
E.I.1. Freundlich (Prompt A)	92
E.I.2. Professionell (Prompt A)	93
E.I.3. Verspielt (Prompt A)	93
E.I.4. Dramatisch (Prompt A)	93
E.I.5. Freundlich (Prompt B)	94
E.I.6. Professionell (Prompt B)	94
E.I.7. Verspielt (Prompt B)	94
E.I.8. Dramatisch (Prompt B)	94
E.II. Luminous-Supreme-Control	95
E.II.1. Verspielt (Prompt A)	95
E.II.2. Freundlich (Prompt A)	95
E.II.3. Professionell (Prompt A)	95
E.II.4. Dramatisch (Prompt A)	95
E.II.5. Verspielt (Prompt B)	96
E.II.6. Freundlich (Prompt B)	96
E.II.7. Professionell (Prompt B)	96
E.II.8. Dramatisch (Prompt B)	96
E.III.GPT-3.5-Turbo	97
E.III.1.Verspielt (Prompt A)	97
E.III.2.Freundlich (Prompt A)	97
E.III.3.Professionell (Prompt A)	98
E.III.4.Dramatisch (Prompt A)	98
E.III.5.Verspielt (Prompt B)	98
E.III.6.Freundlich (Prompt B)	99
E.III.7.Professionell (Prompt B)	99
E.III.8.Dramatisch (Prompt B)	99
E.IV.GPT-4	100
E.IV.1.Freundlich (Prompt A)	100
E.IV.2.Professionell (Prompt A)	100
E.IV.3.Verspielt (Prompt A)	100
E.IV.4.Dramatisch (Prompt A)	100

E.IV.5.Freundlich (Prompt B)	101
E.IV.6.Professionell (Prompt B)	101
E.IV.7.Verspielt (Prompt B)	101
E.IV.8.Dramatisch (Prompt B)	101
F. Quellcode Beispiele der Plugin Implentierung	102
F.I. Die Service XML Datei für die Dependency Injection	103
F.II. ComposePrompt Service	105
F.III.JSON Body für eine Anfrage und eine Antwort an die OpenAI API	108
F.IV.Der Einstiegspunkt der Administrations Erweiterung main.js	110
G. Evaluation durch einen Winzer	111
G.I. Fragenkatalog für persönliche Korrespondenz mit Winzer	111
G.II. Email Korrespondenz mit Winzer	112
G.III.Persönliche Korrespondenz mit Winzer für Evaluation	113
G.IV.Beschreibungstexte für die Evaluierung	115
G.IV.1.Stairs n' Roses, „Struwelpitter“, QbA Mosel	115
G.IV.2.Stairs n' Roses, Stairs n' Roses Riesling feinherb, QbA Mosel	116
G.IV.3.2020 Stairs n' Roses „Sweetest Devotion“	117
G.IV.4.2020 Stairs n' Roses „Sweet Child“	117
G.IV.5.2019 Stairs n' Roses „Red Lion“	118
G.IV.6.Stairs n' Roses, „Next Generation“, QbA Mosel	119
G.IV.7.2018 Stairs n' Roses „Honeymoon“	120

Abbildungsverzeichnis

1.	Die einzelnen Kaufphasen im E-Commerce	6
2.	Ablaufdiagramm des Redaktionsprozesses	11
3.	GPT-3 Tokenizer	13
4.	Abstrahierter Ablauf des Training	15
5.	Visualisierung einer Entscheidungsgrenze bei einem zweidimensionalen Merkmalsvektor	16
6.	Ein einfaches FFNN mit dem Merkmalsvektor zwei	18
7.	Die Transformer Architektur	20
8.	Visualisierung von zwei Attention-Heads	22
9.	Übersicht der Shopware 6 Komponenten	31
10.	Die FunktionsMASTER Schablone	33
11.	Die Tab-Struktur auf der Produkt-Einzelseite	37
12.	Das Standard-Textfeld aus Shopware, erweitert durch eigene Funktionalitäten	38
13.	Die Visualisierung des geplanten Plugins	39
14.	Der Shopware WYSIWYG-Editor, ergänzt durch ein PopUp für das Umformulieren.	40
15.	Der Shopware WYSIWYG-Editor, ergänzt durch einen Menü-Eintrag für das Umformulieren.	41
16.	Der Architektur Entwurf für das Plugin	42
17.	ERD Diagram für das Plugin	44
18.	Ablauf der automatisierten Interaktion mit den LLMs	50
19.	Die minimale Dateistruktur für ein Plugin.	61
20.	Die gerenderte Einstellungsseite für das Plugin	63
21.	Die Dateistruktur für das Administrations-Frontend	67
22.	Die Auswahl einer Textpassage und eine Textpassage die innerhalb eines span-Elements liegt	68
23.	Einfluss-Interesse Matrix der Stakeholder	88

Tabellenverzeichnis

1.	Auszug aus dem Datenblatt für den BDX REVOLUTION Sauvignon Blanc Bordeaux	10
2.	Beispielhafte Trainingsdaten für die Klassifizierung von Wein.	17
3.	Übersicht der vorgestellten Modelle	29

Quellcodeverzeichnis

1.	Ein Beispieldatensatz für das Tuning.	59
2.	Die aktuelle config.xml	62
3.	Minimales JSON-Body für die Anfrage an die OpenAI API.	65
4.	Die Tab-Erweiterung in sw-product-detail.html.twig	67
5.	Die execCommand Funktion	68
6.	Die initialisierung eines Range Interfaces. Entnommen aus [17].	71
7.	Die JavaScript Implementierung für das Umschließen und Entfernung der Umschließung einer Auswahl mit einem span Element	72
8.	Die service.xml	103
9.	Der ComposePrompt Service.	105
10.	Beispiel Anfrage an die OpenAI API.	108
11.	Beispiel Antwort der OpenAI API	109
12.	Die main.js des Plugins	110

Abkürzungsverzeichnis

ADR Architecture decision record

B2B Business-to-Business

B2C Business-to-Consumer

CTP Customer Touchpoint

FFNN Feed Forward neural Network

GPT Generative Pretrained Transformer

LLM Large Language Model

POD Point of Decision

POS Point of Sale

RNN Rekurrente neuronale Netze

WYSIWYG-Editor What-You-See-Is-What-You-Get-Editor

1. Einleitung

Mit der Einführung von ChatGPT¹ im Jahr 2022 hat das Unternehmen OpenAI eine Rekordzahl von einer Millionen Nutzern in fünf Tagen erreicht [11]. Der Zugang zu Large Language Models (LLMs) ist massentauglich und im Alltag angekommen. Neben der Akzeptanz bei Nutzern ist ein Anstieg der Marktgröße in vielen Bereichen der Wirtschaft zu erkennen, sei es durch Nutzung oder durch KI-gestützte Produktinnovationen [19]. Diese Arbeit befasst sich mit dem automatischen Generieren von Beschreibungstexten für Wein im E-Commerce. Die wachsende Bedeutung des Online-Handels und die Notwendigkeit, Kunden umfassende und ansprechende Produktinformationen bereitzustellen, stellen eine aktuelle Herausforderung dar. Beschreibungstexte sind für die Informationsgewinnung äußerst relevant und sind infolgedessen ein wichtiger Bestandteil der Kaufentscheidung. Das Anwendungsgebiet dieser Arbeit, der Weinhandel, beinhaltet eine nicht zu vernachlässigende Komponente: Für das Erstellen von Beschreibungstexten muss ein umfassendes Wissen über die Produkte vorhanden sein. Eine gute Weinbeschreibung geht auf Nuancen ein wie, Aromen, Geschmack und Düfte. Hierbei handelt es sich um Informationen, die nicht trivial zu erkennen sind. Die Problemstellung, die daraus resultiert, ist zweigeteilt: Erstens erfordert die manuelle Erstellung von Produktbeschreibungen ein umfassendes Wissen der Betreiber über die Domäne oder es muss auf Experten zurückgegriffen werden. Dieses Vorgehen ist entweder zeitaufwendig oder kostspielig. Zweitens bringt die Alternative, auf die Beschreibungstexte des Großhandels zurückzugreifen, eine neue Problematik: viele Unternehmen nutzen die gleichen Texte, was zu einer mangelnden Abgrenzung der einzelnen Online-Händler führt.

Die Bewältigung dieser Problemstellung erfordert die Entwicklung einer automatischen Lösung zur Generierung von individuellen Beschreibungstexten. Hierfür soll ein LLM genutzt werden, welches auf der Basis von strukturierten Produktdaten Beschreibungstexte automatisch erstellt. Die strukturierten Produktdaten werden durch den Großhandel bereitgestellt. Sie bestehen in der Regel aus einer Beschreibung, einer Analyse sowie gastronomischen Empfehlungen. Wie relevant dieser Ansatz ist, zeigen die aktuellen Entwicklungen in der Branche. Sowohl die E-Commerce-Plattform Shopify, die mit 10% Marktanteil einer der größten Anbieter in diesem Bereich ist [14], als auch die in dieser Arbeit genutzte E-Commerce-Lösung Shopware, haben eigene KI-gestützte Produkte vorgestellt

¹Bei ChatGPT handelt es sich um das Interface, über welches Nutzer mit den LLMs von OpenAI interagieren können. Hierbei können sowohl Text als auch Bilder über ein Eingabefeld übergeben werden.

[39] [35]. Shopify und Shopware setzen auf eine individuelle Anbindung an ChatGPT, um den Shop-Betreibern die Möglichkeit zu bieten, Produktbeschreibungen automatisch zu erstellen. Shopware geht hierbei noch einen Schritt weiter und bietet mit dem Shopware AI-Copilot verschiedene Funktionen, die die Shop-Betreiber unterstützen sollen. Diese umfassen unter anderem KI-Assistenten für Übersetzung und Export, aber auch automatische Inhaltsgenerierung für Blog-Beiträge, Bewertungen und Beschreibungen [35].

1.1. Zielsetzung

Ziel dieser Arbeit ist die Konzeption und Entwicklung eines Plugins für ein Shopsystem, das auf der Basis von strukturierten Produktdaten Beschreibungstexte generieren kann. Um dieses Ziel zu erreichen, wurden die folgenden Forschungsfragen aufgestellt:

1. Welches Large Language Model eignet sich am besten für die automatische Generierung von Beschreibungstexten aus strukturierten Produktdaten?
2. Auf Basis welcher Kriterien lassen sich automatisch generierte Beschreibungstexte Bewerten und gibt es Unterschiede, wenn sich die Zielgruppe ändert?
3. Wie muss das Userinterface gestaltet sein, um eine automatische Generierung und Evaluierung der Beschreibungstexte durch einen Redakteur im Shop-Backend effizient durchführen zu können?

1.2. Kontext

Diese Arbeit entsteht in Zusammenarbeit mit dem externen Partner Gjuce GmbH. Das inhabergeführte Unternehmen Gjuce ist eine Webagentur aus Köln, die sich seit über 10 Jahren auf E-Commerce-Lösungen spezialisiert hat. Neben dem Verkauf und der Entwicklung von individuellen Lösungen betreibt das Unternehmen Online-Shops im Bereich des Weinhandels [23].

Im Rahmen dieser Arbeit wird auf den Produktkatalog von Vindor² zurückgegriffen. Vindor ist ein Online-Shop für alkoholische Getränke, der durch Gjuce betreut wird. Der Produktkatalog umfasst über 1000 alkoholische Produkte, wobei der Fokus auf Rot-, Weiß- und Roséweinen liegt [20]. Innerhalb von Vindor stehen umfassende Beispiele zu den strukturierten Produktdaten sowie bereits vorhandene Beschreibungstexte zur Verfügung.

²Siehe: <https://vindor.de>

Dementsprechend ist eine Datengrundlage bereits vorhanden und muss nicht selbstständig erhoben werden.

Die Kooperation bietet außerdem den Vorteil, dass Gjuce durch ihre Expertise im Weinhandel die Umsetzung und Validierung der automatischen Lösung unterstützen kann. Darüber hinaus besteht die Möglichkeit, Kontakt zu Experten im Weinhandel, wie zum Beispiel Winzern, herzustellen. Diese sollen durch Interviews die generierten Beschreibungen bewerten.

Durch den Partner entstehen gewisse Implikationen, die den Aufbau dieser Arbeit beeinflussen. Insbesondere die Rahmenbedingungen für die Implementierung der automatischen Lösung sind hierbei relevant. Da Gjuce auf E-Commerce-Lösungen mit Shopware³ spezialisiert ist, wird sich diese Arbeit auf eine Umsetzung mit Shopware beschränken.

1.3. Aufbau

Zu Beginn dieser Arbeit werden in Kapitel 2 ein Überblick über den Weinhandels im E-Commerce gegeben. Dazu werden die einzelnen Kaufphasen behandelt, die der idealtypische Kunde durchläuft. Daraufhin wird betrachtet, welchen Einfluss die Beschreibungstexte auf die Kaufentscheidung haben, um mit dem aktuellen Redaktionsprozess bei Vindor abzuschließen.

Innerhalb von Kapitel 3 werden die Grundlagen geschaffen, die benötigt werden, um die darauffolgenden Kapitel einzuordnen. Es wird eine Auswahl an LLM Aufbauten vorgestellt und ein Blick auf die aktuell verfügbaren Anbieter geworfen. Ein besonderer Fokus wird auf die Transformer-Architektur gelegt, da diese allen betrachteten LLMs zugrunde liegt. Ebenfalls werden die Grundlagen des Prompt Engineering vorgestellt. Hierbei handelt es sich um die Hauptinteraktionsmöglichkeit mit den LLMs, da diese in der Regel über eine Texteingabe durch den Anwender gesteuert werden.

In Kapitel 4 wird die Konzeption der automatischen Lösung durchgeführt. Es werden die in der Medieninformatik erlernten Konzepte angewendet. Über eine Stakeholderanalyse werden die Anforderungen an das Plugin aufgestellt und priorisiert. Daraufhin werden die wichtigsten Funktionalitäten visualisiert, um Entscheidungen für die darauffolgende Implementierung zu erleichtern. Ebenfalls müssen für eine erfolgreiche Implementierung Entscheidungen über die Architektur und den neuen Redaktionsprozess getroffen werden.

³Siehe: <https://shopware.com/de/>

Anschließend an die Konzeption folgt in Kapitel 5 die Implementierung der automatischen Lösung. Zu Beginn wird hierfür die Auswahl der möglichen LLMs aus Kapitel 4 anhand realer Daten von Vindor geprüft, um eine Entscheidung zu treffen. Daraufhin wird in Kapitel 5.3 die durchgeführte Implementierung beschrieben. Hierbei wird sich auf einzelne wichtige Aspekte konzentriert, wie die Implementierung der Generierung und die Erweiterung der Administrationsoberfläche innerhalb von Shopware.

Beendet wird die Arbeit mit einer Evaluation der Beschreibungstexte durch einen Experten aus dem Fachgebiet in Kapitel 6, um in Kapitel 7 ein Fazit der Arbeit zu ziehen und die Forschungsfragen zu beantworten.

2. Weinhandel im E-Commerce

Innerhalb dieses Kapitels wird ein Überblick über den Weinhandel und den allgemeinen Einzelhandel gegeben. Im Anschluss wird betrachtet, welche Relevanz Beschreibungstexte im E-Commerce haben. Das Kapitel endet mit der Darstellung des aktuellen Ablaufs der Integration von Produkten in das Shopsystem von Vindor. Durch die Verlagerung des Point of Sale (POS) vom stationären Handel hin zum Online-Handel wird auch der Weinhandel vermehrt Online abgewickelt [31]. Da man davon ausgehen kann, dass sich sowohl der Weinhandel als auch der allgemeine Online-Handel mit Lebensmitteln in den kommenden Jahren weiterhin vergrößern werden, ist die Relevanz der Thematik gegeben [47]. Es gilt zu betrachten, wie Verbraucher Produkte einkaufen und auswählen, um einen Rückschluss auf das Kaufverhalten für Wein zu treffen. Hierbei wird der Fokus auf den Business-to-Consumer (B2C) Markt gerichtet. Der Business-to-Business (B2B) Markt hat ähnliche Strukturen, durch den Kontext dieser Arbeit ist er jedoch nicht relevant.

2.1. Varianten des Weinhandels

Da sich die Varianten des Weinhandels mit den allgemeinen Varianten des Einzelhandels überschneiden, wird an dieser Stelle zuerst ein genereller Überblick gegeben. Daraufhin wird dargelegt, ob und gegebenenfalls in welchem Umfang eine Differenzierung zum Weinhandel existiert. Es wird zwischen vier Formen, den sogenannten Kontaktprinzipien, im Einzelhandel unterschieden [15]:

Stationärer Handel: Der Handel wird von einem festen Ort aus betrieben. Gängige Formen sind der Supermarkt, Discounter oder der Fachmarkt.

Marktplatz Handel: Temporäre Form des stationären Handels. Aufbauten sind mobil oder leicht rückbaubar. Hierunter fallen Wochenmärkte, Straßenhandel und Ähnliches. Alles Formen, die nach dem Treffpunktprinzip funktionieren.

Ambulanter Handel: Eine nicht ortsgebundene Handelsform. Bei dieser Handelsform suchen die Verkäufer die Kunden auf. Beispiele hierfür ist das sogenannte Hausieren oder das Aufstellen von Verkaufswägen auf Straßenfesten.

Distanzhandel: Räumliche Trennung von Verkäufer und Kunde, welche über ein Kommunikationsmedium überbrückt wird. Unter den Distanzhandel fällt der klassische Online-Handel, aber auch Formen wie Teleshopping oder der Katalogversand.

Durch die Verlagerung des POS existieren in der Praxis immer häufiger Hybridvarianten der klassischen Handelsvarianten. Immer mehr Einzelhändler verbinden den stationären Handel mit dem Distanzhandel [50] [26].

2.2. Die Kaufphasen im E-Commerce

Betrachtet man den Kaufprozess im **Offline-Handel**, so kann dieser in drei Hauptphasen differenziert werden. Es wird in die Vorkaufsphase, die eigentliche Kaufphase und die darauf folgende Nachkauf- beziehungsweise Nutzungsphase unterteilt. Die einzelnen Phasen lassen sich daraufhin in weitere Unterphasen zerteilen. Es ist zu beachten, dass im Online-Handel, im Gegensatz zum Offline-Handel, eine stärkere Verschmelzung der einzelnen Phasen stattfindet [16]. Wichtig ist, dass die hier dargelegten Phasen den idealtypischen Prozess darstellen. In der Praxis ist es selten der Fall, dass Kunden konsequent alle Phasen durchlaufen.

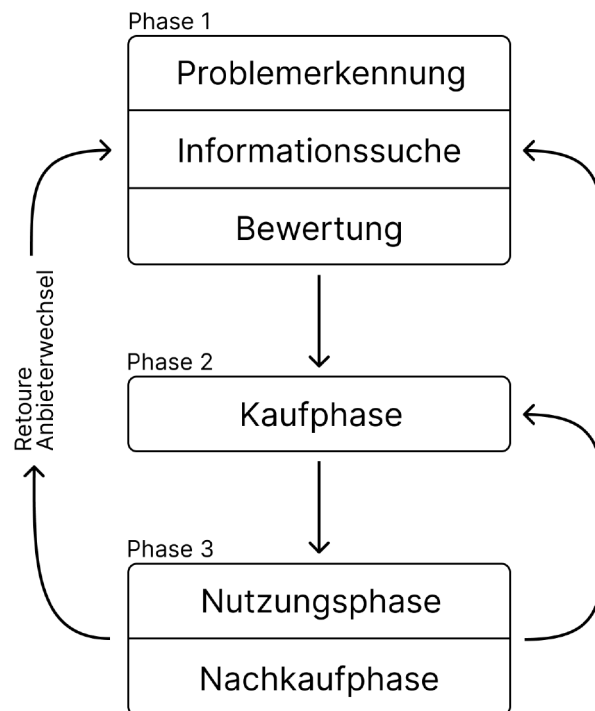


Abbildung 1: Die einzelnen Kaufphasen im E-Commerce. Eigene Darstellung nach [16].

Phase 1, die Vorkaufsphase, besteht aus der Problemerkennung, der Informationssuche sowie der Bewertung der Informationen und Alternativen. Die Problemerkennung, also das Bewusstwerden eines Bedarfs, initiiert einen Kaufprozess. Die verschiedenen Stimuli

können unterschiedlicher Natur sein und intrinsisch sowie extrinsisch auftreten. Hier kann beispielsweise Werbung ein Auslöser sein oder schlicht das Verbrauchen eines Produkts. Darauf folgt die Informationssuche. Die Informationssuche steht in direkter Relation mit der Art des Produkts. Hier sind Faktoren wie der Preis eines Produkts, die Anzahl der verfügbaren Alternativen oder die Relevanz bezogen auf die Lebenssituation ausschlaggebend. Beispielsweise wird in die Suche nach einem neuen Auto deutlich mehr Zeit investiert als bei der Auswahl neuer Klamotten. Ebenfalls spielt der Unterschied zwischen dem Ist- und Ideal-Zustand eine Rolle bei dem Umfang an Zeit, die für die Informationssuche aufgebracht wird. Die Abweichung vom Ideal-Zustand, also beispielsweise das Verbrauchen eines Produkts, ist oftmals auch die erste Aktivierung, um die Vorkaufsphase zu beginnen [16]. Die Vorkaufsphase nimmt den größten Teil des Prozesses ein. Es existieren eine Menge an variablen Faktoren, die eine Entscheidung beeinflussen. Hier wird je nach Produktgruppe oftmals eine Alternativenbewertung beziehungsweise Identifikation von Konkurrenzanbietern durch Kunden durchgeführt. Es muss davon ausgegangen werden, dass bei der Produktgruppe Wein, welche sich tendenziell im höherpreisigen Segment befindet, Kunden einen höheren Fokus auf die Informationssuche legen. Hier gilt es, über klar strukturierte Informationen Argumente zu liefern, um eine Kaufentscheidung einzuleiten.

In Phase 2, der Kaufphase, wird der Einkauf durchgeführt. Der Kunde hat eine Entscheidung getroffen und führt diese aus [15].

Phase 3, die Nachkauf beziehungsweise Nutzungsphase, versetzt im besten Fall den Kunden wieder in Phase 1. In dieser Phase bewertet der Kunde seinen Einkauf und evaluiert die Informationssuche [16].

Im Gegensatz dazu lässt sich der Online-Handel nicht so klar in einzelne Phasen differenzieren. Hier ist eine Verschmelzung der Phasen zu erkennen. Beispielsweise können sowohl Informationssuche und Alternativenrecherche über eine Suchmaschine durchgeführt werden. Der eigentliche Einkauf wird oft ebenfalls über eine Suchmaschine initiiert. Daneben führen größere Online-Händler in der Regel eine Auswahl von verschiedenen Herstellern und Marken, so muss für ein Alternativprodukt nicht einmal mehr der Online-Shop gewechselt werden [15].

Durch die Menge an Händlern, sowohl Offline als auch Online, gibt es eine immer größere Diskrepanz zwischen dem Point of Decision (POD) und dem POS. Kunden haben und

nutzen die Möglichkeit, sich an einem Ort zu informieren und an einem anderen Ort zu kaufen. Hier gilt es durch die Verringerung von Hürden POD und POS anzunähern. Es wird von dem sogenannten ROPO-Effekt gesprochen. ROPO steht als Synonym für *Research Offline and Purchase Online* sowie für den gegensätzlichen Fall, also *Research Online and Purchase Offline*. Konsumenten nutzen also Beratungsangebote an einer Stelle, sei es Offline oder Online, tätigen den eigentlichen Kauf dann aber an anderer Stelle [15]. Diese Arbeit legt keinen Fokus auf die Bewertung der einzelnen Phasen im Kaufprozess. Es wird davon ausgegangen, dass sich Kunden bereits in der Phase der Informationssuche befinden.

2.3. Bedeutung von Beschreibungstexten im E-Commerce

Analysiert man die einzelnen Phasen, so kann man erkennen, dass Beschreibungstexte in allen Phasen ein tragender Bestandteil sind. Innerhalb der ersten Phase sind Beschreibungstexte eine der wichtigsten Informationsquellen für potenzielle Käufer. Sie bieten detaillierte Einblicke in die Eigenschaften, Funktionen und Anwendungen eines Produkts. Diese Informationen stellen einen wichtigen Customer Touchpoint (CTP) dar, da sie das Verständnis für das Produkt fördern und die Erwartungen der Kunden formen [15]. Beschreibungstexte ermöglichen es den Kunden, sich vor dem Kauf ein umfassendes Bild von einem Produkt zu machen und es mit ihren individuellen Anforderungen abzugleichen. Dementsprechend tragen sie maßgeblich zur Kaufentscheidung bei und haben somit auch auf zweite Phase einen direkten Einfluss.

Auch in der Nachkaufphase spielen Beschreibungstexte eine wichtige Rolle. Durch detaillierte und umfassende Informationen in den Beschreibungstexten werden die Kunden in die Lage versetzt, ihre Kaufentscheidungen zu validieren und sicherzustellen, dass das erhaltene Produkt ihren Erwartungen entspricht. Dies reduziert die Wahrscheinlichkeit, dass Kunden nach Erhalt des Produkts desillusioniert sind und eine negative Assoziation mit der Marke oder dem Händler bilden. Die Verbesserung der Nachkaufphase trägt nicht nur dazu bei, das Vertrauen der Kunden in den Händler zu stärken, sondern hat auch einen positiven Nebeneffekt auf die Retourenquote. Eine sorgfältig gestaltete Beschreibung, die alle relevanten Informationen darlegt, kann dazu beitragen, dass Kunden seltener Produkte zurücksenden, da sie bereits vor dem Kauf eine klare Vorstellung von dem Produkt haben [27].

Im Rahmen dieser Arbeit ist ein weiterer Faktor äußerst relevant: Nicht alle Produktarten lassen sich durch visuelle Aspekte (wie Bilder) vollständig darstellen. Insbesondere bei Produkten wie Wein, bei denen sensorische Eindrücke wie Geschmack, Aroma und Gerüche eine wichtige Rolle spielen, nimmt der Beschreibungstext eine kritische Rolle bei der Vermittlung dieser Eigenschaften ein. Die Beschreibung muss genutzt werden, um ein Narrativ zu gestalten, das Kunden informiert und überzeugt. Der Beschreibungstext ist eine Verbindung zwischen den Erfahrungen, die das Produkt verspricht und der Vorstellung des Kunden. Im Gegensatz zu einer visuellen Darstellung vermittelt der Beschreibungstext eine umfassendere und facettenreichere Vorstellung des Produkts.

2.4. Redaktionsprozess bei der Integration von Produkten

An dieser Stelle wird der gegenwärtige Ist-Zustand bezüglich der Integration von Produkten in ein Shopsystem dargestellt. Der Prozess umfasst unter anderem das Einpflegen von Produktdaten und Produktbeschreibung. Im Folgenden wird dieser als Redaktionsprozess zusammengefasst. Im Kontext dieser Arbeit bezieht sich der Ablauf direkt auf das Vorgehen innerhalb von Gjuce und somit das Shopsystem Shopware. Der Ablauf wurde durch Gespräche innerhalb des Partnerunternehmens erarbeitet. Näheres zu Shopware wird in Kapitel 4.1 dargelegt. Die Prozesse werden möglichst technologieunabhängig betrachtet und lassen sich ohne Schwierigkeiten auch auf andere Systeme übertragen. Je nach Wein-klasse und Großhändler ist der Prozess leicht abgewandelt, alle haben aber gemeinsam, dass sie derzeit vollständig manuell durchgeführt werden.

In der Regel kann davon ausgegangen werden, dass ein Großhändler zu jedem Wein auch ein Datenblatt liefert. Dieses beinhaltet neben den Analysedaten eine Beschreibung über das Weingut, sowie den Wein. Die Analysedaten umfassen Informationen wie Alkoholgehalt, Restzucker, Säuregehalt, Rebsorte und Ähnliches. In der Tabelle 1 ist ein Auszug für den Wein *BDX REVOLUTION Sauvignon Blanc Bordeaux AOC* vom Importeur *Weinkontor Freund* abgebildet⁴. Diese Analysedaten sind in den Datenblättern immer enthalten, die Beschreibungen hingegen haben unterschiedliche Ausprägungen.

Im ersten Schritt werden, anhand der Datenblätter, die Produkt- und Analysedaten über einen Wein im Shopsystem eingepflegt, da diese unabhängig von der Beschreibung für den Kunden einsehbar sein sollen. Im nächsten Schritt werden die Beschreibungstexte geprüft

⁴Das vollständige Datenblatt ist bei Weinkontor Freund einsehbar: <https://www.weinkontor-freund.de/wein/frankreich/producta.vignobles/4034130>

Beschreibung	
Jahrgang:	2022
Herkunftsland:	Frankreich
Region:	Bordeaux
Rebsorte:	Sauvignon Blanc
Farbe:	hellgelb mit grünen Reflexen
Ausbau:	Der Wein wurde imahltank ausgebaut
Charakteristik:	verlockender Duft nach frischem Gras und Stachelbeeren und einem Hauch Holunderblüten in der Nase, am Gaumen saftig, leicht mit Aromen von Zitrus und Guave, gipfelt in einem frischem, lebendigem Finale
Analysedaten	
Lagerfähigkeit:	5 Jahre
Alkoholgehalt:	11,5 % vol.
Restzucker:	0,9 g/l

Tabelle 1: Auszug aus dem Datenblatt für den BDX REVOLUTION Sauvignon Blanc Bordeaux. Entnommen aus [24]

und ihrer Ausprägung gemäß eingepflegt. Der vollständige Ablauf wird in Abbildung 2 dargestellt, zu unterscheidende Fälle sind:

Fall A, das Produkt hat keinen Beschreibungstext: Dieser Fall tritt in der Regel nur bei Produkten aus dem Niedrigpreissegment ein. Hier muss durch einen Redakteur⁵ ein Beschreibungstext für das Produkt erstellt werden.

Fall B, das Produkt hat keinen deutschen Beschreibungstext: In diesem Fall wird der ursprüngliche Text übersetzt, da Vindor hauptsächlich im deutschsprachigen Raum agiert. Dafür sind ebenfalls Sprachkompetenzen notwendig.

Fall C, das Produkt hat einen mangelhaften oder kurzen Beschreibungstext: Auch in diesem Fall wird für das Produkt eine neue Beschreibung erstellt. Angelehnt an den ursprünglichen Beschreibungstext.

Fall D, das Produkt hat nur Pressebewertungen: Neben Beschreibungen können für Produkte Pressebewertungen existieren. Beispielsweise bei Weinen mit Auszeichnung. Sollte es sich bei dem Produkt um ein stark saisonales Produkt handeln⁶, kann es vorkommen, dass Vindor für die Beschreibungstexte die Pressebewertungen nutzt. Ist dies

⁵Hierunter fallen beispielsweise die Shopbetreiber, Entwickler oder Copywriter, also alle Nutzergruppen, welche den Online-Shop initial mit Informationen anreichern müssen. Ebenfalls gehören Marketing-Personal und ähnliche dazu, welche im Laufe der Nutzungsdauer Produkte sowie Texte pflegen. Diese Nutzergruppen werden hier als Redakteure zusammengefasst.

⁶Saisonale Produkte sind nur für eine begrenzte Zeit im Online-Shop verfügbar.

nicht der Fall, wird auch hier ein neuer Beschreibungstext erstellt.

Fall E, das Produkt hat einen Beschreibungstext: Hier gilt es abzuwägen, inwiefern das Produkt auch von anderen Händlern angeboten wird. Da alle Händler die gleichen Texte der Großhändler zur Verfügung gestellt bekommen, ist hier eine starke Überschneidung vorhanden.

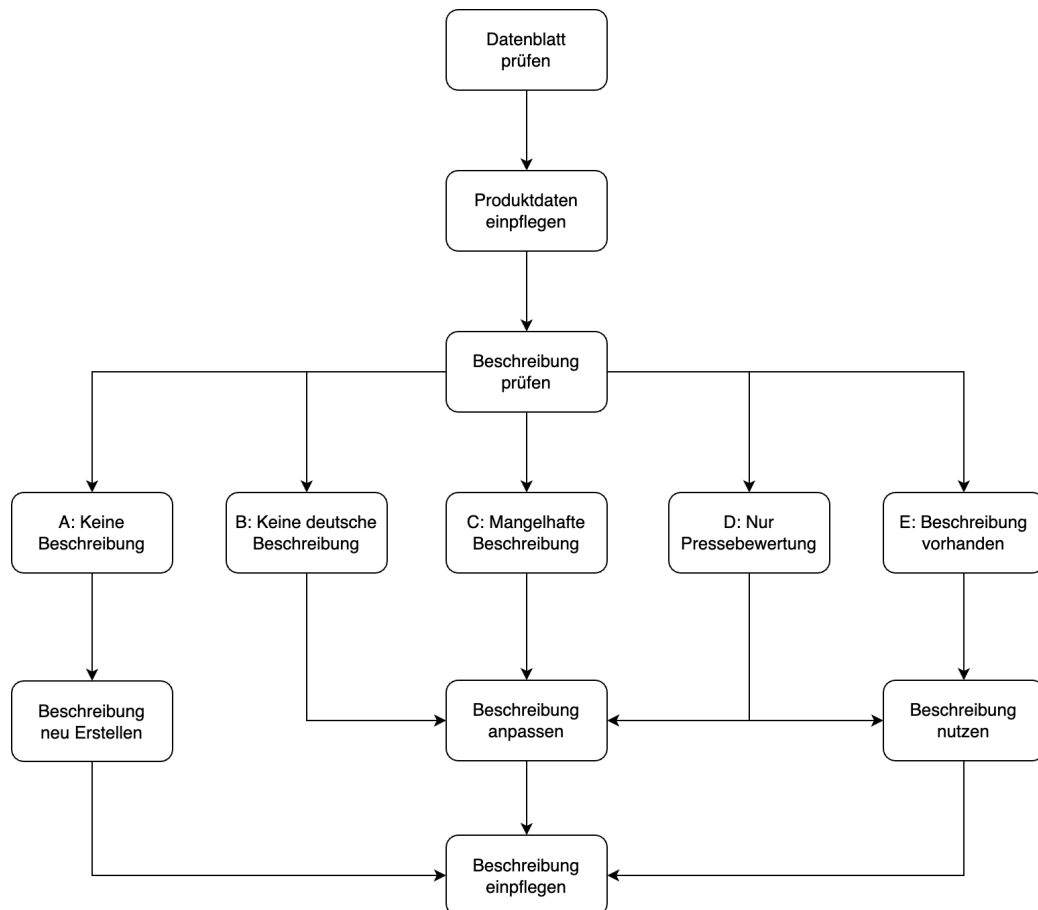


Abbildung 2: Ablaufdiagramm des Redaktionsprozesses. Eigene Darstellung.

Zusammenfassend lässt sich festhalten, dass Beschreibungstexte in 4 von 5 Fällen angepasst oder komplett neu erstellt werden. Dieser Prozess ist zeitaufwändig, da sich das Sortiment bei Vindor regelmäßig ändert und aktuell um die 1000 individuellen Produkte angeboten werden. Die Durchführung dieser Aufgabe erfordert spezifische Fachkenntnisse. Einerseits ist die Fähigkeit zur Übersetzung von verschiedenen Sprachen erforderlich, um Informationen und Inhalte korrekt zu übermitteln. Andererseits ist ein umfassendes Verständnis der Weincharakteristika notwendig, um Aromen, Düfte und ähnliche Aspekte in einer Beschreibung präzise darzustellen. Bei Gjuce ist dafür ein Partner zuständig, der auch den Einkauf beim Großhandel durchführt.

3. Sprachmodelle und ihre Anwendung

Im nun folgenden Kapitel wird dargelegt, wie LLMs aufgebaut sind und warum sich die Transformer-Architektur aktuell durchgesetzt hat. Es wird auf die Funktionsweise von LLMs eingegangen, sowie ein Vergleich der verfügbaren Sprachmodelle in Bezug auf die Textgenerierung vorgenommen. Im Kontext dieser Arbeit ist vor allem die Textgenerierung relevant, daher wird sich auch auf diese konzentriert. Da sowohl Anwendungsfälle von LLMs als auch Aufbauten ein sehr breites Spektrum füllen, hat diese Arbeit keinen Anspruch auf Vollständigkeit.

Andere Anwendungsfälle für die später betrachteten LLMs sind neben der Textgenerierung vor allem die Text-Klassifizierung, Dialog-Systeme sowie Übersetzung. Die Text-Klassifizierung findet beispielsweise in der Spam- oder Fake-News-Erkennung Anwendung. Dialog-Systeme wie Chat-Bots sind genauer betrachtet eine Kombination aus Klassifizierung und Generierung, da sie im ersten Schritt Fragen einordnen müssen, um daraufhin eine Antwort zu generieren. Bei allen Anwendungsfällen ist es von entscheidender Bedeutung, dass die Modelle ein hohes Maß an Kontext bewahren können, um Zusammenhänge zu verstehen und Relevanz zu erhalten. Dieser Mechanismus wird in der Transformer-Architektur als Attention (zu Deutsch Aufmerksamkeit) betitelt.

3.1. Aufbau und Funktionsweise von Sprachmodellen

Heruntergebrochen sind die Ergebnisse von LLMs eine Wahrscheinlichkeitsverteilung. Also, welches Wort hat die höchste Wahrscheinlichkeit, auf das vorherige Wort zu folgen. Wobei die Modelle keine Wörter nutzen, sondern eine Repräsentation in Form eines Zahlen-Vektors. Man spricht von Tokens, welche verarbeitet werden. Das Transferieren von Wörtern in ihre numerische Repräsentation wird durch Anwendungen durchgeführt, die als Tokenizer bezeichnet werden. Je nach Tokenizer werden einzelne Worte durch mehrere Tokens repräsentiert. Die Wörter `Hallo Welt` werden im Tokenizer vom GPT-3 Modell durch die Zahlen `[34194, 78, 370, 2120]` repräsentiert. In Abbildung 3 ist die Visualisierung des GPT-3 Tokenizer zu sehen. Man kann erkennen, dass sowohl `Hallo` als auch `Welt` zweigeteilt verarbeitet werden⁷.

Da LLMs ein Themenbereich aus dem Machine Learning sind, welches wiederum in die

⁷Unter <https://platform.openai.com/tokenizer> kann Beispielfür das GPT-3 Modell von OpenAI der Tokenizer betrachtet werden.

Tokens	Characters
4	10

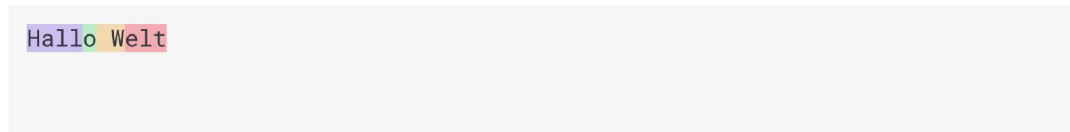


Abbildung 3: Eine Visualisierung der zusammenhängenden Buchstaben im GPT-3 Tokenizer. Jede Farbgruppe ist als ein zusammenhängendes Token zu verstehen. Entnommen aus [37].

Künstliche Intelligenz einzuordnen ist, gilt es an dieser Stelle einige Begriffe zu erläutern, die im folgenden relevant sind. Machine Learning ist das Entwickeln oder Aufstellen eines Algorithmus für das Interpretieren von Daten und Mustern. Der Begriff Machine Learning wurde 1959 durch Arthur L. Samuel geprägt, der in seiner Studie „Some Studies in Machine Learning Using the Game of Checkers“[42] einen Algorithmus beschreibt, der so programmiert ist, dass er das Spiel *Dame* besser als der Entwickler spielen kann. Er schließt damit ab, dass die Möglichkeit des Lernens aus Erfahrung beziehungsweise das Erkennen von Zusammenhängen einen Großteil des Programmieraufwandes eliminieren wird [42].

Vereinfacht gesagt ist Machine Learning das Synthetisieren der Beziehungen zwischen den zugrundeliegenden Informationen. Es wird zu Beginn ein Datensatz gesammelt, um daran ein statistisches Modell zu trainieren. Man unterscheidet beim Trainieren zwischen vier Arten. Dem *überwachten Lernen*, dem *unüberwachten Lernen*, dem *teilüberwachten* sowie dem *bestärkten Lernen*.

3.1.1. Lern Algorithmen

Beim **überwachten Lernen (supervised learning)** existiert eine Datenmenge aus gekennzeichneten Beispielen. Demnach gibt es für jeden Datenpunkt bereits das gewünschte Ergebnis. Ob diese Datenpunkte manuell oder anderweitig klassifiziert werden, ist an dieser Stelle nicht relevant. Hierbei handelt es sich dann um einen sogenannten Merkmalsvektor, welcher die einzelnen Datenpunkte klassifiziert. Ein einfaches Beispiel wäre hier der Merkmalsvektor [Spam, Kein Spam], in der Spam-Erkennung. Das Ziel ist ist, ein Modell zu entwickeln, welches neue Datenpunkte in die vorher definierten Kategorien (Merkmalsvektoren) einteilen kann [12].

Das **unüberwachte Lernen (unsupervised learning)** hat im Gegensatz dazu eine

Menge an ungekennzeichneten Beispielen. Ziel des unüberwachten Lernens ist es, ein Modell zur Erkennung von Mustern, Strukturen oder Beziehungen zwischen den Datensätzen [12] aufzustellen. Es bietet sich für Anwendungsfälle an, bei welchen im Voraus nicht bekannt ist, welche Kategorien (Merkmalsvektoren) optimal sind. Ein Beispiel ist eine Kundensegmentierung im Einzelhandel. Hierbei handelt es sich um die Gruppierung von Kunden in verschiedene Segmente auf Grundlage vorhandener Informationen, ohne Vorkenntnisse wie viele Gruppierungen es gibt oder welche Eigenschaften diese Gruppierungen haben. Liegen Informationen über Kaufverhalten, Demografie und geografischen Standort vor, könnten diese durch das unüberwachte Lernen in zusammenhängende Gruppierungen eingeteilt werden.

Die dritte Form, das **teilüberwachte Lernen (semi-supervised learning)** ist die Verbindung der vorherigen Lernformen. Es liegt eine Mischung aus gekennzeichneten, sowie ungekennzeichneten Datensätzen vor. Der Vorteil hierbei ist, die größere Menge an Datenpunkten und die damit einhergehende höhere Stichprobe für die Wahrscheinlichkeitsverteilung [12].

Im Gegensatz zu den Vorherigen Formen verfolgt das **bestärkten Lernen (reinforcement learning)** einen anderen Ansatz. Es wird über Belohnungen und Strafen ein Algorithmus (der sogenannte RL-Agent) so trainiert, dass er am Ende einen Ablauf (Policy) für das Erfüllen der Aufgabe erarbeitet hat. Folglich geht es darum, die Entscheidungsfindung dem Algorithmus zu überlassen diese durch Reaktionen zu bewerten. Hierfür eignen sich vor allem Aufgaben die sequentieller Natur sind [12].

In Abbildung 4 ist eine abstrahierte Übersicht über den Ablauf beim Trainieren eines Modells dargestellt: Je nach gewähltem Algorithmus werden die Trainingsdaten gesammelt und klassifiziert. Ab dem Moment, in dem das Training beginnt, müssen immer wieder Feedbackschleifen durchgeführt werden: Teile der Datensätze werden nicht für das Training genutzt, sondern dafür, das Modell zu testen. Sobald die benötigte Genauigkeit bei der Einteilung neuer Daten vorhanden ist, können Modelle in der Produktivumgebung für Vorhersagen genutzt werden.

Allen Lernalgorithmen für die Klassifizierung liegt zugrunde, dass sie eine Entscheidungsgrenze festlegen. Sie teilen den Datensatz in zwei oder mehr Gruppen. Je nach Modell und Algorithmus hat man unterschiedliche Ausprägungen dieser Entscheidungsgrenze. Sie reichen von einfachen linearen Funktionen bis hin zu beliebig komplexen, nicht linearen

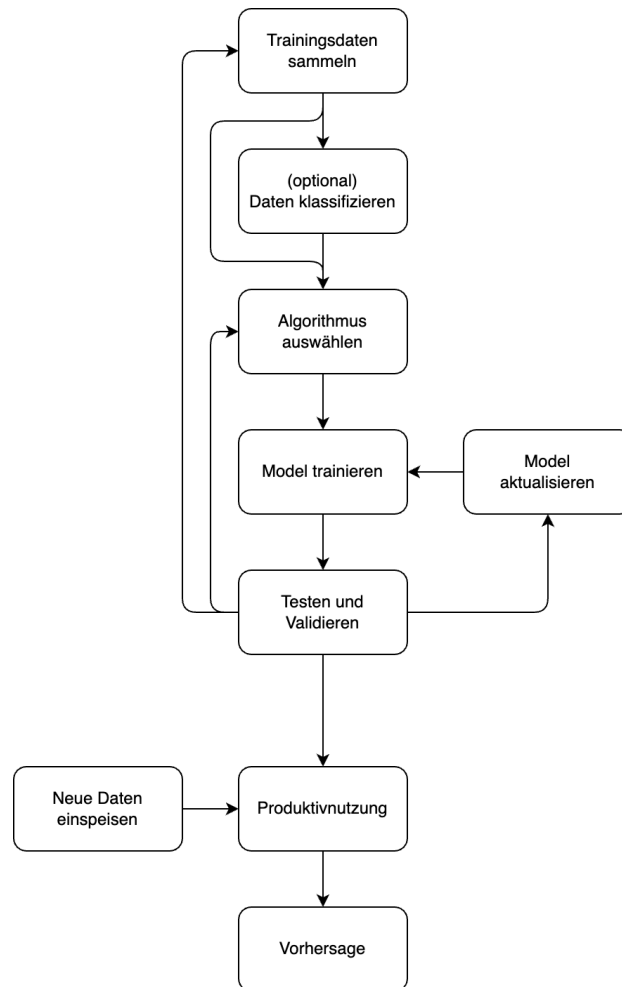


Abbildung 4: Abstrahierter Ablauf des Trainings. Eigene Darstellung nach [33].

Funktionen[8] [12].

In Abbildung 5 ist eine beispielhafte Visualisierung für ein Modell mit zweidimensionalem Merkmalsvektor nach dem Training dargestellt. Die Entscheidungsgrenze (w) trennt die Trainingsdaten durch eine lineare Funktion in zwei Gruppen⁸. Sobald eine Entscheidungsgrenze existiert, können neue Daten in das Modell eingespielt werden. Hatte man zuvor einen unabhängigen und zufälligen Trainingssatz, ist es statistisch wahrscheinlich, dass sich die neuen Daten in der Nähe der Trainingsdaten befinden. Dementsprechend kann das Modell neue Daten klassifizieren, obwohl diese nicht in den Trainingsdaten enthalten waren [12].

⁸Der zugehörige Merkmalsvektor hat dementsprechend auch die Ausprägung zwei, beispielsweise [spam, kein spam].

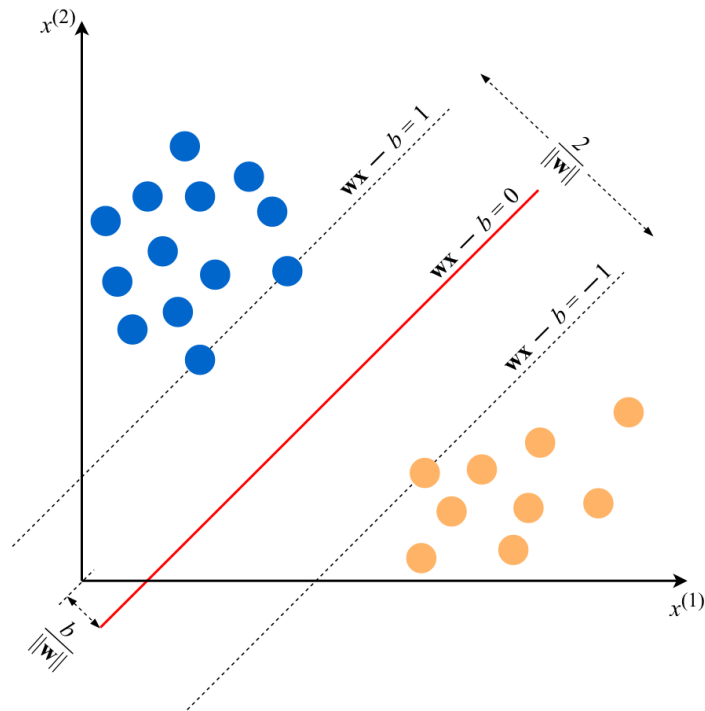


Abbildung 5: Visualisierung einer Entscheidungsgrenze bei einem zweidimensionalen Merkmalsvektor. Entnommen aus [12].

3.1.2. Neuronale Netze

Im Folgenden werden die Grundlagen zu neuronalen Netzen erläutert. Der Umfang dieser Erläuterung wird auf die für die Arbeit relevanten Teilbereiche beschränkt, da das Themengebiet der Neuronalen Netze sehr breit ist. Es wird sich auf Feed Forward neural Network (FFNN) und Rekurrente neuronale Netze (RNN) konzentriert. FFNN sind einerseits die Grundform von neuronalen Netzen und andererseits ein fester Bestandteil der später dargestellten Transformer-Architektur und aus diesen Gründen für das weitere Verständnis relevant. RNN wurden vor der Transformer-Architektur hauptsächlich für die Textgenerierung genutzt und werden infolgedessen für das Einordnen und Vergleichen benötigt.

Feed Forward Neuronale Netze sind die einfachste Form der neuronalen Netze, werden aber in der Kombination von mehreren FFNN ein mächtiges Werkzeug für das Klassifizieren von Informationen. Der *Feed Forward* Aspekt ergibt sich hier aus der Eigenart, dass die Informationsweitergabe ausschließlich in einer Richtung erfolgt. Die Verarbeitung ist damit frei von Rückkopplungen. Man geht also davon aus, dass sowohl Eingaben als auch Ausgaben unabhängig voneinander sind. Daher sind sie für Anwendungsfälle,

bei welchen die Reihenfolge⁹ der Informationen relevant ist, nicht geeignet sind. FFNN bestehen aus einer *Eingabe-Schicht* (*Input-Layer*), einer *Ausgabe-Schicht* (*Output-Layer*) sowie einer bis mehreren versteckten Schichten (*Hidden-Layer*). Jede Schicht erhält als Eingabe die Ausgabe aller vorherigen Schichten. Alle Verbindungen zwischen den einzelnen Schichten haben eine Gewichtung (weight), welche angibt, wie relevant dieser Input für das Ergebnis ist. Initial sind diese Gewichtungen zufällig gesetzt und werden während dem Trainingsprozess eingestellt. Ziel ist es, die beste Kombination aus Gewichtungen zu finden, um die Genauigkeit der Ergebnisse des neuronalen Netzwerks zu maximieren. Zusätzlich zur Gewichtung verfügt jede Verbindung über einen sogenannten Bias, welcher als Konstante auf das Ergebnis addiert wird. Der Bias verhindert, dass Ausreißer in den Datensätzen das Ergebnis verzerren. Dieser wird ebenfalls beim Training einem optimalen Wert angenähert [44]. Dies hängt stark mit dem Anwendungsfall zusammen und lässt sich nicht pauschal begründen. Einerseits könnte die Genauigkeit der Klassifizierung maximiert werden, andererseits kann ebenso eine hohe Trainingsgeschwindigkeit das Ziel sein.

Weintyp (Input 1)	Preis (Input 2)	Label
Rotwein	5€	Kaufen
Rotwein	10€	Nicht Kaufen
Weißwein	10€	Kaufen
Weißwein	14€	Nicht Kaufen
Rosewein	5€	Kaufen
Weißwein	13€	Nicht Kaufen
Rosewein	8€	Kaufen
Rotwein	7€	Nicht Kaufen
Weißwein	9€	Kaufen
Rosewein	11€	Kaufen

Tabelle 2: Beispielhafte Trainingsdaten für die Klassifizierung von Wein.

Die Funktionsweise eines FFNN wird anhand eines exemplarischen Szenarios erläutert. Angenommen, das Ziel besteht darin, die Kaufentscheidung für Wein zu unterstützen, dann gliedert sich der Prozess des Trainings eines FFNN in folgende Schritte:

Im ersten Schritt wird ein Trainingssatz definiert. Zur Vereinfachung des Beispiels werden nur zwei Merkmale angegeben: der Weintyp und der Preis. Der Weintyp kann die Ausprägung Rot-, Weiß- oder Rosewein haben. Der Preis ist eine numerische Größe. Jeder Eintrag wird mit einem Merkmalsvektor [Kaufen, Nicht Kaufen] versehen. Dies führt

⁹Beispielsweise die Textgenerierung.

zu dem beispielhaften Trainingssatz in Tabelle 2.

Im zweiten Schritt wird die Hälfte des Trainingssatzes verwendet, um das FFNN zu trainieren. Es handelt sich hierbei um einen überwachten Lernalgorithmus, da für jeden Datenpunkt das Ergebnis bereits bekannt ist. Nach jedem Trainingsdurchlauf kann nun mit den nicht genutzten Datenpunkten geprüft werden, ob Gewichtungen und Bias zufriedenstellend eingestellt sind. Sobald das Model eine akzeptable Klassifizierung erreicht hat, kann es mit neuen Informationen über einen Wein versorgt werden. Das Ergebnis wird dann in Form einer Kauf- oder Nicht-Kauf-Empfehlung zurückgegeben.

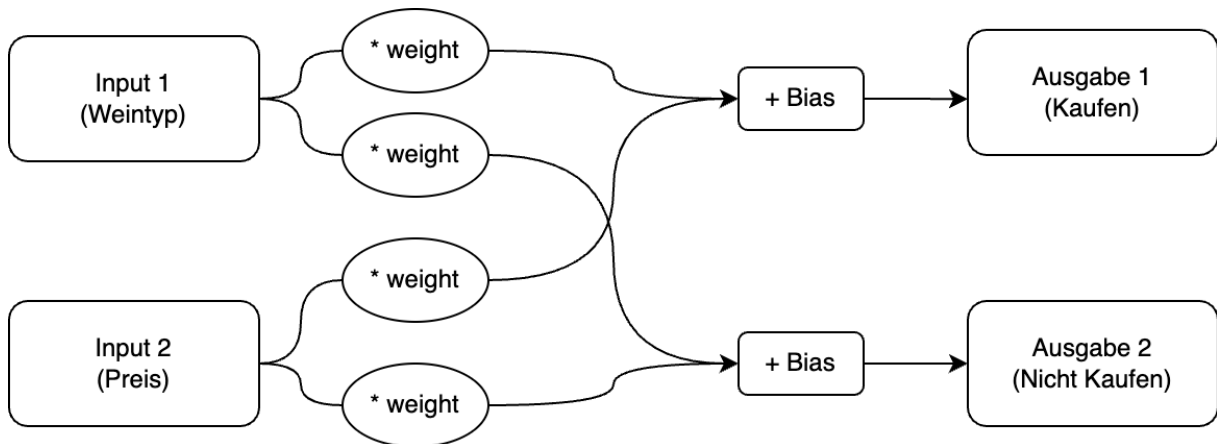


Abbildung 6: Ein einfaches FFNN mit dem Merkmalsvektor zwei. Eigene Darstellung nach [6].

Die Abbildung 6 visualisiert ein vereinfachtes FFNN in dieser Größenordnung. In der Praxis umfassen Merkmale Millionen Datenpunkte und die Ausgabe ermöglicht eine wesentlich feinere Granulierung als die hier dargestellten Merkmalsvektoren.

Rekurrente neuronale Netze Bei RNNs handelt es sich um neuronale Netze, welche im Gegensatz zu FFNNs innerhalb der einzelnen Schichten eine Schleife (Rückkopplung) enthalten. Ergebnisse aus dem vorherigen Input haben einen Einfluss auf den aktuellen Input. Es wird eine gewisse Art an Gedächtnis geschaffen, um Zusammenhänge und Abhängigkeiten innerhalb der Eingabe zu wahren. Sie sind dementsprechend nutzbar für das Erzeugen und Klassifizieren von Sequenzen, wie beispielsweise Text. Dadurch, dass Gewichtungen (weights) rückwirkend angepasst werden, entsteht jedoch das Problem, dass bei diesen Netzwerken eine Über- oder Unteranpassung auftreten kann. Man spricht hierbei von *Exploding* bzw. *vanishing gradients*. Zusätzlich bringt die Unteranpassung ein weiteres Problem mit sich, nämlich dass diese Art von Netzwerken dazu tendiert, bei längeren Inhalten den Anfang zu *vergessen*. Daneben hat man wegen der Rückkopplung keine

Möglichkeit, RNNs parallel auszuführen und zu trainieren. Teile der Probleme, wie das Verlieren von Abhängigkeiten, werden in Verbesserungen von RNNs eliminiert. Kommen aber immer mit dem Nachteil der explodierenden Trainings und Ausführungszeiten, da die Netzwerke beliebig komplexer werden [12].

3.1.3. Transformer Architektur

Als Resultat der dargelegten Probleme, wurde 2017 im Paper „Attention is all you need“ die Transformer Architektur vorgeschlagen [48]. Innerhalb dieser Architektur wird sich auf den so genannten *attention mechanism* fokussiert und für die Klassifizierung nur FFNNs genutzt. Es entsteht der Vorteil, dass keine rekursive Ausführung vorhanden ist und die Modelle parallelisiert ausgeführt werden können. Eine höhere Parallelisierung wiederum erhöht die Trainingsgeschwindigkeit signifikant [48].

Aufbau Das Modell besteht aus einem Encoder und einem Decoder, welche wiederum aus sechs gleich aufgebauten Schichten bestehen. Die Encoder Schichten beinhalten den sogenannte *Multi-Head Attention Mechanismus* und ein FFNN. Sie verarbeiten die Eingaben. Die Decoder Schichten haben neben dem Multi-Head Attention Mechanismus und dem FFNN noch eine zusätzliche *Masked Multi-Head Attention* Schicht. Die Decoder Schichten verarbeiten die Ausgaben. Wie bei anderen neuronalen Netzen wird zwischen jeder Verbindung eine Normalisierung (*Add & Norm* in Abbildung 7) durchgeführt, welche dafür sorgt, dass die Relation der Informationen erhalten bleibt. Abbildung 7 zeigt den Aufbau der Transformer Architektur, die linke Seite ist hier der Encoder, die rechte der Decoder. Zur Erhöhung der Übersichtlichkeit werden in der Abbildung lediglich ein Encoder und ein Decoder dargestellt, in der Realität sind mehrere Instanzen dieser Komponenten vorhanden. Es ist zu erkennen, dass Verbindungen (Pfeile) zwischen den Komponenten bestehen, um Teile zu überspringen. Dennoch sind keinerlei Rückkopplungen enthalten.

Die Innovation der Transformer Architektur ist einmal der Multi-Head Attention Mechanismus sowie das Positional-Encoding. Die übrigen Komponenten sind etablierte Konzepte und stellen keine Neuerungen dar.

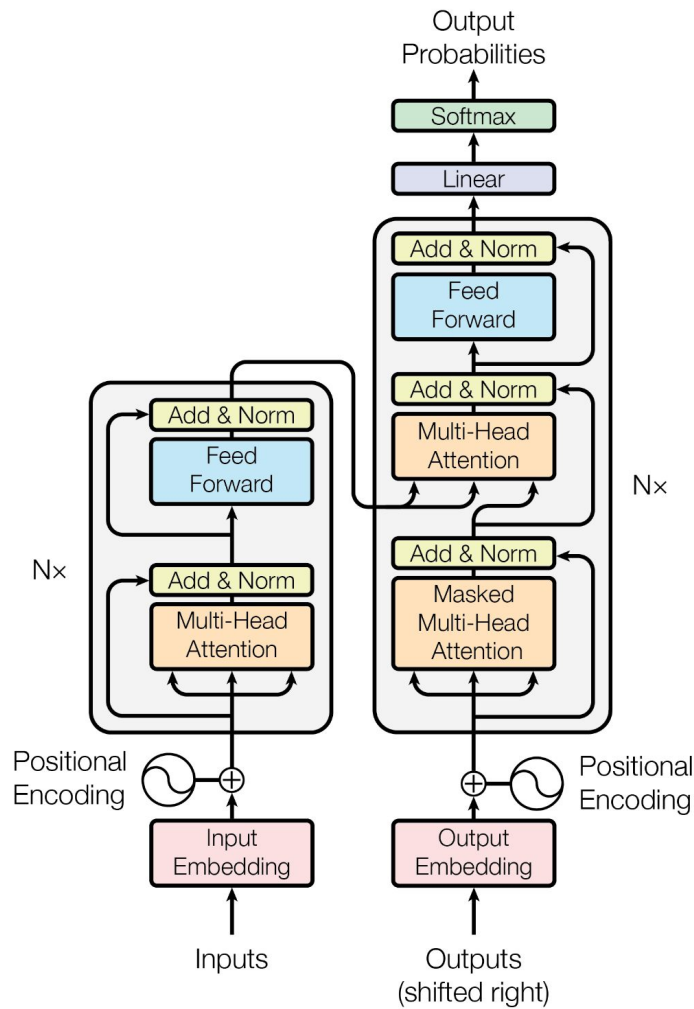


Abbildung 7: Die Transformer Architektur. Entnommen aus [48].

Positional Encoding Dadurch, dass in der Transformer Architektur keinerlei Rückkopplungen enthalten sind und die Eingaben gleichzeitig eingelesen werden¹⁰, kann das Modell keine Rückschlüsse auf die Reihenfolge der Eingabe-Tokens ziehen. Die Anwendungsfälle, wie Textgenerierung oder Übersetzung, sind aber sequentiell und benötigen eine Möglichkeit die Zusammenhänge der Tokens zu erkennen. Deshalb wird vor dem Encoder und dem Decoder Informationen über die Reihenfolge hinzugefügt. Das Positional Encoding ist also die Komponente, bei der Informationen über die Reihenfolge der Tokens hinzugefügt werden. Hierfür wird eine mathematische Funktion verwendet, die als Wellenfunktion modelliert ist. Diese Wellenfunktion wurde so aufgebaut, dass Tokens (Wörter), die sich in räumlicher Nähe befinden (zum Beispiel aufeinanderfolgende Wörter in einem Satz) einen ähnlichen Wert haben. Auf diese Weise kann das Modell die relativen Positionen

¹⁰Im Gegensatz zu FFNN und RNN, die die Eingaben Nacheinander aufnehmen (siehe Kapitel 3.1.2)

der Tokens innerhalb der Eingabe verstehen und berücksichtigen.

Die Wahl der spezifischen Methode für das Positional Encoding kann variieren, im ursprünglichen Transformer-Paper sowie in weiteren Arbeiten werden verschiedene Ansätze diskutiert. Die von den Autoren gewählte Methode zeichnet sich jedoch durch ihre Flexibilität aus. Sie kann auch auf Eingabesequenzen angewendet werden, deren Länge während des Trainings nicht explizit berücksichtigt wurde. Dies ermöglicht dem Modell, mit unterschiedlich langen Eingaben umzugehen, ohne erneut trainiert werden zu müssen. Die Zusammenhänge innerhalb der Wellenfunktion sind eine weitere Komponente, welche das Modell während des Trainings lernt, um Rückschlüsse auf Position und Distanz der Tokens zu ziehen [48] [34].

Multi-Head Attention Mechanismus Im Multi-Head Attention Mechanismus erfolgt die eigentliche Berechnung der *Self-Attention* (zu Deutsch: Aufmerksamkeit), welche die Verknüpfungen zwischen den Eingaben herstellt. Das Modell besitzt die Möglichkeit, bei jedem Token sämtliche anderen Token zu analysieren, um Schlüsse über die Relevanz des gegenwärtig betrachteten Tokens zu ziehen. Betrachtet man exemplarisch den folgenden Satz zur Übersetzung, wird deutlich, warum das relevant ist:

„The animal didn’t cross the street because **it** was too tired“ [49]

Der Zusammenhang sollte für den Leser offensichtlich sein: **it** bezieht sich auf das Tier, das zu müde war, um die Straße zu überqueren. Um genau diese Zusammenhänge zu verstehen, existieren in der Transformer-Architektur mehrere Attention Heads, wobei sich jedes einzelne auf einen individuellen linguistischen Aspekt fokussiert. Hier ist wieder der Fall, dass die anfänglichen Werte zufällig initialisiert werden, um diese beim Training anzupassen. Durch die Analyse mehrerer Millionen Sätze lernt das Modell schließlich Verbindungen und Beziehungen. In Abbildung 8 ist die Visualisierung von zwei Heads aus einer Beispielimplementation¹¹ des Papers „Tensor2Tensor for Neural Machine Translation“ dargestellt [49]. Man erkennt, dass sich eines auf *the animal* und eines auf *too* konzentriert. Zur besseren Übersichtlichkeit, werden hier nur zwei Attention-Heads gezeigt.

Das Ergebnis der Multi-Head-Attention Schicht ist eine Attention-Score für jedes Token, welches angibt, wie relevant es ist und worauf sich im Weiteren fokussiert werden muss,

¹¹Der Quelltext ist auf Github zu finden: https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/Transformer_translate.ipynb

ähnlich wie die Gewichtungen in den FFNN.

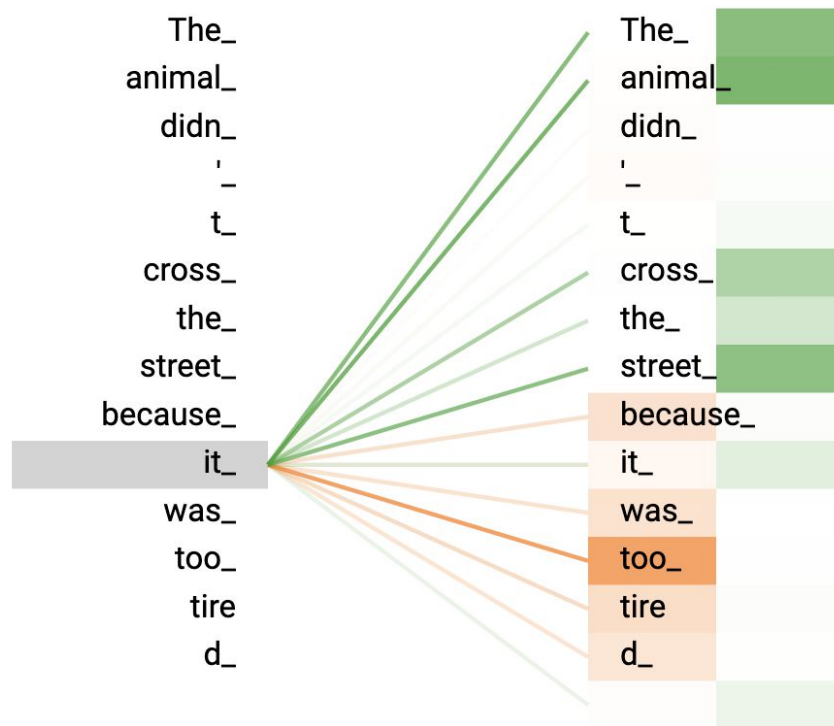


Abbildung 8: Visualisierung von zwei Attention-Heads, die sich auf die Zusammenhänge bei einer Übersetzung konzentrieren. Eine dunklere Farbe bedeutet einen höheren Attention-Score und demnach ein Fokus auf diese Wortkombinationen. Entnommen aus [49].

3.2. Prompt Engineering

Der Begriff *Prompt Engineering* bezieht sich auf das Formulieren von Anweisungen für LLMs, mit dem Ziel, eine gewünschte Ausgabe zu erhalten. Vor allem in textbasierten Modellen ist dies die einzige Möglichkeit der Interaktion. Die dem LLM zugeführte Prompt stehen im direkten Zusammenhang mit der Qualität der Ergebnisse. Eine Prompt besteht aus einem oder mehreren der folgenden Elemente [43]:

- Anweisungen
- Kontext
- Eingabedaten
- Ausgabevorgaben

Die **Anweisung** ist die Aufgabe, die von dem Modell ausgeführt werden soll. Sie kann als Aussage (*Schreibe einen Text über diesen Wein*) oder als explizite Fragestellung (*was*

versteht man unter Prompt Engineering?) formuliert werden. Der **Kontext** stellt ergänzende Informationen bereit. Der Umfang und die Genauigkeit, sowohl der Anweisung als auch des Kontexts, korrelieren mit der Präzision der erzeugten Ergebnisse. Die **Eingabedaten** referenzieren ebenfalls zusätzliche Informationen, die integriert werden können. Würde man dem Modell die Aufgabe stellen, einen Text zusammenzufassen, wäre der ursprüngliche Text die Eingabedaten. Über die **Ausgabevorgaben** kann das Modell dazu gebracht werden, eine bestimmte Syntax zu befolgen. Beispielsweise erzeugt die Anweisung *Formatiere die Antwort im Markdown-Format* eine Ausgabe, die im Markdown-Format strukturiert ist.

Weiterhin unterscheidet man zwischen *Zero-Shot* und *Few-Shot* Prompting. Beim Zero-Shot Prompting wird dem Modell eine allgemeine Anweisung oder Fragestellung gegeben, die das gewünschte Ergebnis beschreibt, aber nicht explizit auf die Aufgabe hinweist. Das Modell verwendet sein vorheriges Training, um eine angemessene Antwort zu generieren. Neben der Anweisung erhält das Modell keine Beispiele für das Lösen der Aufgabe. Für das Generieren von Beschreibungstexten kann eine Zero-Shot Prompt wie folgt aufgebaut sein:

Erstelle eine Beschreibung für einen Wein (Anweisung). Die Beschreibung wird in einem Onlineshop genutzt (Kontext). Es geht um folgenden Wein: BDX REVOLUTION Sauvignon Blanc, Sauvignon Blanc, Weißwein, Französisch (Eingabedaten).

Nutzt man diese Prompt mit dem OpenAI Model GPT-3.5 ist folgendes ein mögliches Ergebnis:

Tauchen Sie ein in die erfrischende Revolution des Geschmacks mit unserem BDX REVOLUTION Sauvignon Blanc. Dieser herausragende französische Weißwein verkörpert die Essenz eines modernen Sauvignon Blancs und entführt Ihre Sinne auf eine unvergessliche Reise.

Der BDX REVOLUTION Sauvignon Blanc beeindruckt mit seiner leuchtend blassen Farbe, die die Lebendigkeit und Frische dieses Weins widerspiegelt. Die Nase wird von einem faszinierenden Bouquet aus Aromen von saftigen Stachelbeeren, spritziger Limette und reifen weißen Pfirsichen umworben. Diese duftenden Noten werden durch zarte Anklänge von frischen Kräutern und ei-

*ner subtilen mineralischen Note ergänzt, die dem Wein eine unverwechselbare Komplexität verleihen. [...]*¹²

Das Ergebnis dieser Prompt ist mangelhaft, denn Anweisungen und Kontext sind nicht spezifisch genug. Die beschriebenen Aromen sind falsch, der Name des Weins wird in jedem Absatz wiederholt und der Text ist zu lang. Hier lässt sich eines der Hauptprobleme von LLMs erkennen: Sie tendieren dazu, bei unklaren Aufgabenstellungen und geringer Informationsdichte zu halluzinieren. Als Halluzinieren bezeichnet man das Phänomen, dass das LLMs Ausgaben generiert, die plausibel wirken, aber nicht akkurat sind [28]. Im Kontext dieser Arbeit bedeutet das, dass Beschreibungen mangelhaft bis falsch sein können. In kritischen Anwendungsgebieten sind die Konsequenzen dieses Problems um einiges schwerwiegender. Im medizinischen Sektor beispielsweise ist es zwingend erforderlich, eine präzise Überprüfung der Korrektheit und Genauigkeit von Informationen durchzuführen. Interessant wird es bei komplexeren Anweisungen, wie mathematischen Problemen oder bei sehr hoher Informationsdichte im initialen Prompt. Formulierungen wie *antworte erst, wenn du die Aufgabe selber gelöst hast* oder *gehe die Aufgabe Schritt für Schritt durch* verringern Halluzinationen signifikant [29]. OpenAI betitelt dieses Phänomen passend mit „Give GPTs time to think“ [37]. Der Autor geht davon aus, dass das Problem der Halluzinationen im Kontext der Zielsetzung von geringer Bedeutung ist. Der Großteil der Informationen, welche das Modell benötigt, wird über die strukturierten Daten bereitgestellt. Lediglich bei Aromen und Düften muss eine Transferleistung durchgeführt werden.

Few-Shot Prompting geht einen Schritt weiter, indem es dem Modell eine begrenzte Menge an Beispielen gibt, die zur Anweisung passen. Beispielsweise nach dem Format ¹³:

Anweisung oder Erklärung der Aufgabe - Beispielantwort

Erstelle eine Beschreibung für einen Wein [...] - Helles goldgelb. Im Bukett fruchtig nach Apfel, am Gaumen elegant und feinperlig. Ausdrucksstarker [...]

Erstelle eine Beschreibung für einen Wein [...] - In der der Nase ein kraftvolles, für diesen mächtigen Jahrgang ungewohnt fruchtiges Bouquet von dunklen Pflaumen [...]

¹²Die vollständige Antwort ist in Anhang A

¹³Das Trennzeichen - (Minus) ist nur eine Möglichkeit der Unterteilung. Andere Möglichkeiten wie : (Doppelpunkt) oder / (Schrägstrich) führen zum gleichen Ergebnis.

...

Erstelle eine Beschreibung für einen Wein [...] -

Das letzte Beispiel wird freigelassen, das Modell generiert daraufhin anhand der vorgegebenen Muster eine Antwort. Es ist zu beachten, dass hierbei ein Vorurteil entstehen kann, da nur eine begrenzte Menge an Beispielen zur Verfügung steht. Die Anzahl der Tokens, welche in einer Prompt möglich sind, ist immer begrenzt¹⁴. Wie die Vorurteile entstehen können, wird am Beispiel der Produktbeschreibung deutlich: Je nach Auswahl der Beispiele kann es vorkommen, dass hier nur Weine vorhanden sind, welche zitrusartige Aromen enthalten. Dementsprechend geht das Modell daraufhin davon aus, dass man erwartet, dass in jeder Beschreibung die Aromen zitrusartig sind. Um dies zu verhindern, müsste eine größere Menge an Beispielen mit mehreren Hundert Beschreibungen vorhanden sein, was über eine einzelne Prompt nicht möglich ist. Daher ist für den Anwendungsfall in dieser Arbeit das Few-Shot Prompting nicht geeignet. Eine Möglichkeit, dennoch Beispiele für einen Anwendungsfall zu übergeben, kann über das Tuning von LLMs erreicht werden.

3.3. Auswahl passender Sprachmodelle

Im folgenden Kapitel werden aktuelle Sprachmodelle vorgestellt, um eine Entscheidung für die in Kapitel 5 folgende Implementierung zu ermöglichen. Durch den aktuellen Fokus auf KI-gestützte Produktinnovationen gibt es eine Reihe an Unternehmen, welche an eigenen Sprachmodellen arbeiten. Hier muss jedoch klar unterschieden werden, zwischen Unternehmen, die eigene Sprachmodelle implementieren und denen, die nur auf vorhandenen Modellen aufbauen beziehungsweise diese nutzen. Es werden an dieser Stelle nur Unternehmen betrachtet, die eigene Lösungen implementiert haben.

3.3.1. Aleph Alphas Luminous Reihe

Als eines der wenigen europäischen Unternehmen arbeitet die deutsche Aleph Alpha GmbH seit ihrer Gründung 2019 [7] an einem eigenen LLM. Ebenfalls auf der Transformer-Architektur aufgebaut, ist die aktuelle Modellreihe mit dem Namen Luminous für Privatkunden und Firmen verfügbar. Luminous umfasst sechs verschiedene Modelle, die sich vor allem in der Anzahl der Trainingsparameter unterscheiden. In Tabelle 3 sind die verschiedenen Ausführungen aufgeschlüsselt. Wie auch das GPT-4 Modell von OpenAI haben die

¹⁴Diese hängt mit der maximalen Anzahl an Tokens zusammen, welche ein Modell entgegennehmen kann. Sie ist je nach Modell unterschiedlich hoch.

Modelle Luminous-Base und -Extended die Fähigkeit, multimodale Eingaben entgegenzunehmen und zu interpretieren. Neben der Unterscheidung bei der Modellgröße sind die Modelle aufgeteilt in den Zero-Shot und Few-Shot Ansatz beim Prompting. Die Control-Modellreihe ist optimiert für den Zero-Shot Ansatz, benötigt folglich keine Beispiele für das Ausführen von Anweisungen. Alle Modelle lassen sich sowohl über einen Playground als auch über eine API nutzen und konfigurieren [22].

3.3.2. OpenAIs GPT Modelle

OpenAI, 2015 als gemeinnützige Organisation gegründet, hat sich in den letzten Jahren zu einem Vorreiter im Bereich der künstlichen Intelligenz entwickelt. Unterstützt wurde OpenAI zu Beginn durch namhafte Geldgeber und Organisationen, die Spenden in Höhe von einer Milliarde Dollar für das Erreichen der Vision zur Verfügung stellten. OpenAI wurde mit dem Ziel gegründet, Fortschritte im Bereich der KI zu erzielen und sicherzustellen, dass diese Fortschritte der gesamten Menschheit zugutekommen [25]. Seit 2019 ist das Unternehmen dem Mutterkonzern OpenAI Inc. untergeordnet und ist im Zuge dessen zu einem gewinnorientierten Unternehmen geworden, um nach eigenen Aussagen ihre Möglichkeiten der Skalierung und Investition anheben zu können [9]. Dieser Wandel brachte Diskussionen über die Auswirkungen auf die ursprünglichen Ideale und Ziele der Organisation mit sich.

Das erste nutzbare Modell GPT-2 (Generative Pre-trained Transformer 2) wurde 2019 für die Öffentlichkeit zugänglich gemacht. Zu Beginn wurde nur eine kleinere Variante veröffentlicht¹⁵, da anfänglich Bedenken hinsichtlich Missbrauch von KI-generierten Inhalten bestanden [40]. Kurze Zeit später wurde dennoch der Quellcode als Open-Source veröffentlicht¹⁶.

Die Einführung von GPT-3 im Jahr 2020 markierte den nächsten Meilenstein. Einerseits erhöhen sich die Trainingsdaten und Parameter enorm (siehe Tabelle 3) [10], andererseits gibt es durch eine exklusive Partnerschaft mit Microsoft kein Bestreben mehr den Quellcode oder die Trainingsdaten zu veröffentlichen [45]. Seit 2023 sind die GPT-3 und GPT-4 Modelle fester Bestandteil in den Microsoft 365 Anwendungen und in der Bing Suche. Für die Öffentlichkeit ist der Zugang zum GPT-3 Modell nur über die API oder das

¹⁵Bezogen auf die Anzahl der Parameter.

¹⁶Der Quellcode für das GPT-2 Modell ist auf Github zu finden: <https://github.com/openai/gpt-2>.

Semi-Kommerzielle Produkt ChatGPT möglich.

Die aktuellste Weiterentwicklung ist seit 2023 das GPT-4 Modell und repräsentiert einen ersten Schritt in Richtung Multimodalität. Dieses Modell kann nicht nur Text verarbeiten und generieren, sondern auch Bilder interpretieren und darauf reagieren [36]. Diese Fähigkeit, sowohl visuelle als auch textuelle Informationen zu verarbeiten, eröffnet neue Anwendungsmöglichkeiten, die über reine Textgenerierung hinausgehen. Zum Zeitpunkt dieser Arbeit befindet sich das GPT-4 Modell noch in einer geschlossenen Beta-Phase und kann nur über das kostenpflichtige ChatGPT Plus¹⁷ Programm oder die API¹⁸ genutzt werden.

3.3.3. Googles Bard und PaLM 2

An dieser Stelle wird bewusst auf eine umfassende Darstellung der Geschäftsstrukturen von Google beziehungsweise dem Mutterkonzern Alphabet verzichtet, da Sprachmodelle nur einen kleinen Aspekt der unternehmensweiten Aktivitäten darstellen. Eine vollständige Darstellung würde keinen Mehrwert für diese Arbeit bieten. Die ursprüngliche Transformer-Architektur im Paper „Attention is all you need“ wurde zum Großteil von Google Mitarbeitern entwickelt. Google hat demnach bereits an einer Vielzahl von LLMs gearbeitet. Für diese Arbeit ist vor allem das 2023 vorgestellte Produkt Bard interessant. Bei Bard handelt es sich um Googles Antwort auf ChatGPT von OpenAI. Wie auch ChatGPT ist Bard das Interface, über welches mit dem internen LLM interagiert wird [46]. Im Hintergrund läuft aktuell das LLM PaLM 2 (Pathways Language Model 2). Wie auch die Modelle von OpenAI und Aleph Alpha steht PaLM 2 in verschiedenen Größen zur Verfügung, welche von 7 Mrd. Parametern (Gecko Modell) bis zu 70 Mrd. Parametern (Unicorn Modell) reichen [13].

3.3.4. Metas LLaMa

Meta (ehemals Facebook), ist wie Alphabet ein Unternehmen, welches nur einen kleinen Teil ihrer Kapazitäten auf die Entwicklung von Sprachmodellen fokussieren. Daher wird ebenfalls darauf verzichtet die Unternehmensstrukturen von Meta darzulegen. Metas Ant-

¹⁷Siehe: <https://openai.com/blog/chatgpt-plus>

¹⁸Der Zugang zur API muss über eine Warteliste beantragt werden, daneben muss innerhalb der Organisation mindestens eine erfolgreiche Zahlung durchgeführt werden. Diese Voraussetzungen waren erst zur Mitte der Arbeit gegeben.

wort auf die Modelle der anderen Anbieter ist das LLM LLaMa. Das LLaMa Modell ist seit Juli 2023 in der Version 2 verfügbar. Im Gegensatz zu den anderen hier vorgestellten Modellen ist das LLM von Meta vollständig Open-Source verfügbar. Daraus resultiert das Problem, das die Modelle Lokal ausgeführt werden müssen und über keine API verfügen [32].

Modell	Release	Parameter	Multimodal	Anmerkung	Zugang
GPT-2	2019	125 - 335 Mio.	Nein		-
text-davinci-003 (GPT-3.5)	2020	175 Mrd.	Nein	Wird Anfang 2024 abgeschaltet.	Playground und API
GPT-3.5-Turbo	2022	175 Mrd.	Nein	Aktuelles Standardmodell, wird in ChatGPT verwendet.	Playground und API
GPT-4	2023	Nicht veröffentlicht.	Ja	API Zugang nur über eine Warteliste möglich.	Playground und API
Luminous-Base	2022	13 Mrd.	Ja		Playground und API
Luminous-Extended	2022	30 Mrd.	Ja		Playground und API
Luminous-Supreme	2022	70 Mrd.	Nein		Playground und API
Luminous-Base-Control	2022	13 Mrd.	Ja	Optimiert für den Zero-Shot Ansatz.	Playground und API
Luminous-Extended-Control	2022	30 Mrd.	Ja	Optimiert für den Zero-Shot Ansatz.	Playground und API
Luminous-Supreme-Control	2022	70 Mrd.	Nein	Optimiert für den Zero-Shot Ansatz.	Playground und API
PaLM 2	2023	8 - 540 Mrd.	Ja	Drei Ausführungen, bezogen auf die Parameteranzahl, vorhanden. API Zugang nur über eine Warteliste möglich.	Über Interface Bard und API
LLaMa 2	2023	7 - 70 Mrd.	Ja	Vier Ausführungen, bezogen auf die Parameter Anzahl, vorhanden.	Muss Lokal ausgeführt werden. Vercel SDK Playground eine mögliche Alternative.

Tabelle 3: Übersicht der vorgestellten Modelle. Quellen: [40] [30] [21] [29] [13] [32].

4. Konzeption einer automatischen Lösung

Innerhalb dieses Kapitels wird auf die Konzeption der automatisierten Lösung eingegangen. Durch den Partner gibt es gewisse Rahmenbedingungen, welche vorgegeben sind. Hierunter fällt, wie bereits dargelegt, die Datengrundlage des Weinhandels. Daneben bewegt sich die automatische Lösung ebenfalls innerhalb des Kontextes des Partners, welcher sich auf E-Commerce-Lösungen mit Shopware spezialisiert hat. Dementsprechend wird die Lösung durch eine Shopware-Erweiterung realisiert. Weitere Vorgaben bezüglich des Technologie-Stacks existieren nicht und werden in den folgenden Kapiteln entschieden. Zu Beginn wird ein Überblick über das Shopware-Ecosystem gegeben, in welchem sich ab dann bewegt wird. Darauffolgend werden über eine Stakeholderanalyse die Anforderungen an das Plugin ermittelt und priorisiert. Anhand der Anforderungen wird daraufhin eine Visualisierung der Funktionalitäten erstellt, um diese zusammen mit den Redakteuren zu evaluieren. Parallel zur Visualisierung wird der Architekturentwurf erstellt.

4.1. Shopware Ecosystem

Die E-Commerce-Lösung Shopware ist aktuell in der Version 6 verfügbar. Das Ecosystem umfasst verschiedene Komponenten, darunter der Shopware Core, Plugins, Apps, Themes, API-Schnittstellen und mehr. Shopware ist vollständig Open-Source und in der Community-Edition kostenlos nutzbar. Daneben existieren verschiedene kostenpflichtige Pläne, welche unter anderem ein managed hosting sowie individuellen Support enthalten [4]. Für diese Arbeit ist die Community-Edition ausreichend, daher werden die kostenpflichtigen Pläne nicht näher erläutert. Die Plattform besteht aus drei Hauptteilen [3]:

Backend (Shopware Core) Die Grundlage von Shopware ist der Shopware-Core. Dieser enthält die Business-Logik und stellt die APIs für die Administration und die Storefront bereit. Shopware nutzt als Grundlage das PHP-Framework Symfony und ist nach dem Model-View-Controller(MVC)-Pattern aufgebaut. Für die persistente Speicherung wird eine MySQL-Datenbank genutzt. Die Datenbankzugriffe werden über ein individuelles Data Abstraction Layer (DAL) abgewickelt, was eine auf Shopware zugeschnittene Interaktion ermöglicht [3].

Administration Die Administration ist der Einstiegspunkt und das Frontend für die Shop-Betreiber. Hier werden Produkte, Kategorien, Inhalte und Verkaufskanäle verwaltet.

Ein Verkaufskanal in Shopware, ist jede Instanz, über welche Produkte verkauft werden. Das ist zum Beispiel ein Online-Shop-Frontend oder eine Social Shopping¹⁹ Integration. Aufgebaut ist die Administration als Single-Page-Application (SPA) basierend auf Vue 2, mit Twig als Template Engine. Das Administrations-Frontend enthält keinerlei Business-Logik, sondern kommuniziert mit dem Core über eine dedizierte REST-Schnittstelle (Admin API) [3].

Storefront Bei der Storefront handelt es sich um einen der möglichen Verkaufskanäle und den Einstiegspunkt für Kunden. Es ist im Gegensatz zur Administration keine SPA sondern eine eigenständige PHP-Anwendung mit Twig als Templating Engine. Wie auch die Administration kommuniziert die Storefront über eine eigene REST-Schnittstelle (Store API) mit dem Shopware Core [3].

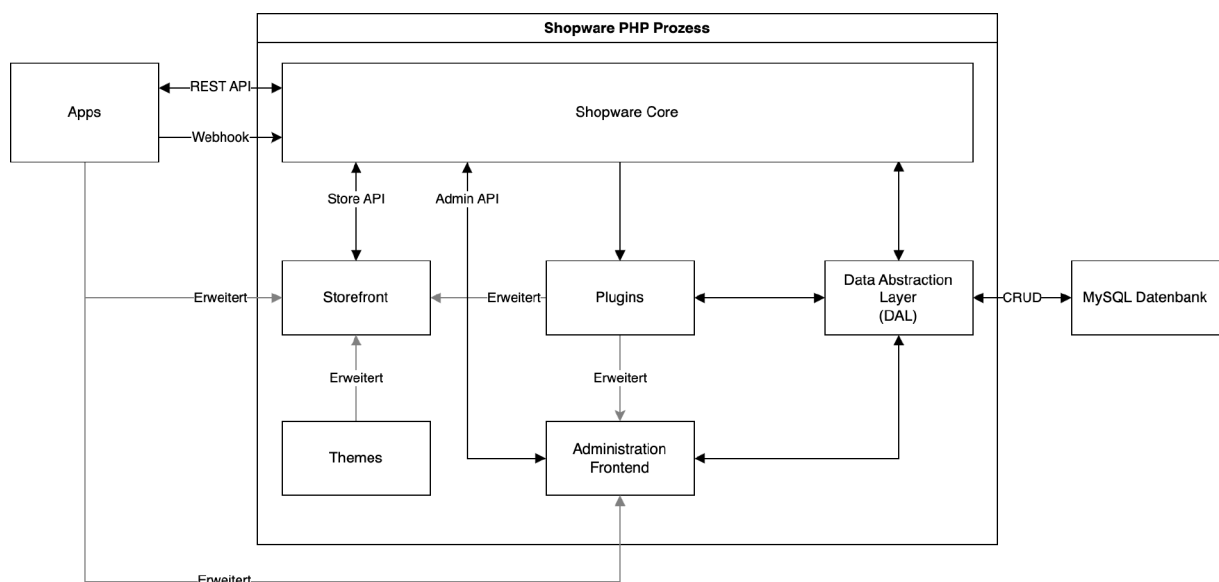


Abbildung 9: Übersicht der Shopware 6 Komponenten. Eigene Darstellung nach [3]

Alle drei Komponenten sind modular aufgebaut und lassen sich durch Plugins, Apps und Themes erweitern. Abbildung 9 zeigt die vorgestellten Komponenten und deren Kommunikationswege. Mit Hilfe von Themes können Aussehen und Templating der Storefront nach den individuellen Anforderungen angepasst werden. Themes dienen somit vorwiegend der Anpassung des visuellen Erscheinungsbilds. Plugins und Apps hingegen sind für tiefere Eingriffe in das Shopware-System gedacht. Sie ermöglichen die Erweiterung der Funktionalitäten und Prozesse innerhalb der Shopware-Plattform. Hierbei besteht ein

¹⁹Hierunter fällt das Verkaufen über Plattformen wie Amazon, Google, Meta oder Ähnliches, wo kein direkter Online-Shop vorhanden ist, sondern über die jeweilige Plattform ein Kauf getätigt wird.

wesentlicher Unterschied zwischen den beiden: Apps werden nicht innerhalb des Shopware Core ausgeführt, sondern kommunizieren ausschließlich über eine REST API. Hieraus resultiert unter anderem, dass Apps keine Änderungen an der Datenbankstruktur²⁰ vornehmen können, während Plugins diese Möglichkeit bieten [3].

4.2. Stakeholder

Zur Identifizierung der Interessengruppen für eine automatisierte Lösung wird eine Stakeholderanalyse durchgeführt. Ziel ist es, einen ersten Überblick zu erhalten, um die darauffolgende Anforderungsermittlung zu unterstützen. Infolgedessen wurden drei Hauptgruppen von Stakeholdern identifiziert, deren Bedürfnisse und Anforderungen betrachtet werden müssen.

Die erste Gruppe umfasst die Stakeholder, die unmittelbar mit der automatisierten Lösung interagieren werden. Hierzu zählen die Shopbetreiber, Redakteure und das Marketing-Personal. Sie haben ein starkes Interesse an den spezifischen Funktionalitäten, da diese direkt ihre täglichen Arbeitsprozesse beeinflussen. Zudem sind sie darauf bedacht, dass die Ergebnisse der automatisierten Lösung verkaufsfördernd sind. Sie haben einerseits den größten Einfluss und andererseits auch das größte Interesse.

Die zweite Gruppe profitiert von der automatischen Lösung. Dazu gehören die Weinproduzenten und Weinexperten. Diese Akteure haben vor allem ein Interesse daran, dass ihre Produkte auf korrekte und präzise Weise beschrieben werden. Obwohl sie weniger Einfluss auf die Lösung selbst haben als die vorherige Gruppe, darf ihre Expertise und ihre Bedenken keinesfalls außer Acht gelassen werden. Ihr umfangreiches Wissen über Ihre Produkte macht sie zu wichtigen Ansprechpartnern.

Die dritte und letzte Gruppe konsumiert im Grunde nur die Ergebnisse der automatisierten Lösung. Darunter fällt der Endkunde, der ein Produkt kaufen möchte. Sie haben keinen Einfluss auf die Lösung und voraussichtlich ebenfalls ein geringes Interesse.

Eine Aufschlüsselung der vorgestellten Stakeholder ist in Anhang B.

4.3. Anforderungen an das Plugin

Basierend auf den Stakeholder und in Zusammenarbeit mit dem Partner wurden Anforderungen an die automatisierte Lösung erstellt. Aufgebaut nach der *FunktionsMASTER*

²⁰Dies betrifft nur das Anlegen oder das Modifizieren von Tabellen. CRUD-Operationen sind in Apps möglich.

Schablone von C. Rupp [41] in Abbildung 10. Wie innerhalb der Schablone festgelegt werden die Prioritäten der Anforderungen durch die Begriffe „Muss“, „Sollte“ und „Wird“ dargestellt. Dies hat den Vorteil, dass die Bedeutung und Dringlichkeit der jeweiligen Anforderungen verdeutlicht wird [41].

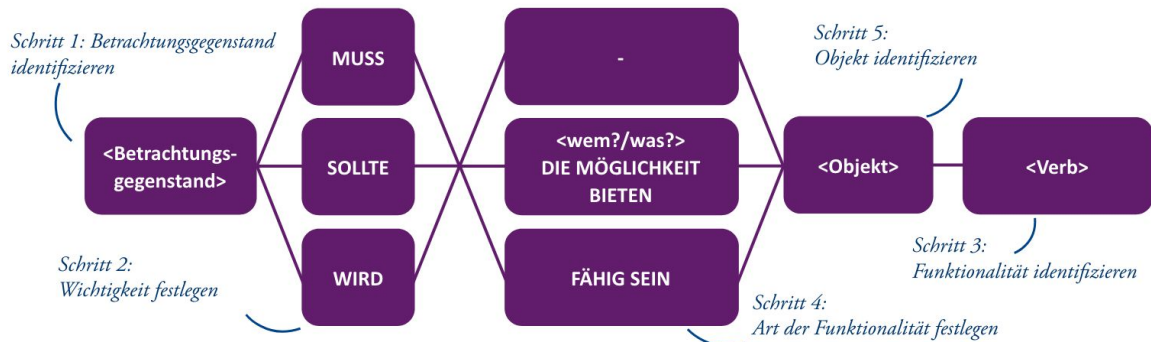


Abbildung 10: Die FunktionsMASTER Schablone. Entnommen aus [41].

Die „Muss“-Anforderungen sind essenziell und bilden die Grundvoraussetzungen, die für eine Interaktion mit dem System benötigt werden. Sie beinhalten die Funktionalitäten wie, einen Beschreibungstext zu generieren und Ergebnisse zu veröffentlichen, bestimmte Keywords auszuschließen oder zu erzwingen und eine Tonalität festzulegen. Diese Funktionen sind unerlässlich für die Kernfunktionalität des Systems. Bei der Anforderungsermittlung sind acht „Muss“-Anforderungen ermittelt worden.

Die „Sollte“-Anforderungen erweitern des Systems und bieten den Nutzenden zusätzliche Möglichkeiten, wie das Austauschen einzelner Teile der Ergebnisse, die Bewertung von Ergebnissen, die Generierung neuer Beschreibungstexte und die Betrachtung einer Historie der generierten Texte. Diese Funktionen tragen zur Verbesserung der Benutzererfahrung bei, sind aber nicht zwingend notwendig. Insgesamt sind sechs „Sollte“-Anforderungen ermittelt worden.

Die „Wird“-Anforderungen sind für zukünftige Erweiterungen des Systems und umfassen die Möglichkeit, für mehrere Produkte gleichzeitig eine Generierung zu starten, bereits erstellte Beschreibungstexte als Referenz zu nutzen und Warnungen bei fehlender Konfiguration. Diese Funktionen sind zwar aktuell nicht priorisiert, könnten jedoch in späteren Entwicklungsphasen implementiert werden. Insgesamt sind sieben „Wird“-Anforderungen ermittelt worden.

Für das Erfüllen der Anforderungen A12 wird das persistente Speichern vorausgesetzt, welches über eine Erweiterung der Datenbank realisiert wird. Dementsprechend muss ein Plugin für Shopware entwickelt werden, da Apps im Shopware-Ecosystem keine Verände-

rungen an der Datenbank vornehmen können.

Priorität 1: Muss

- A1** Das Plugin muss den Nutzenden die Möglichkeit bieten, einen Beschreibungstext zu Generieren.
- A2** Das Plugin muss den Nutzenden die Möglichkeit bieten, bestimmte Attribute auszuschließen.
- A3** Das Plugin muss den Nutzenden die Möglichkeit bieten, bestimmte Attribute zu erzwingen.
- A4** Das Plugin muss den Nutzenden die Möglichkeit bieten, Attribute für alle Produkte auszuschließen.
- A5** Das Plugin muss den Nutzenden die Möglichkeit bieten, Ergebnisse zu veröffentlichen.
- A6** Das Plugin muss den Nutzenden die Möglichkeit bieten, eine Tonalität²¹ für den Text festzulegen.
- A7** Das Plugin muss den Nutzenden die Möglichkeit bieten, Beschreibungstexte zu Bearbeiten.
- A8** Das Plugin muss den Nutzenden warnen, wenn ein Beschreibungstext bearbeitet wurde und versucht wird, einen neuen zu generieren.

Priorität 2: Sollte

- A9** Das Plugin sollte den Nutzenden die Möglichkeit bieten, einzelne Teile der Ergebnisse umzuformulieren.
- A10** Das Plugin sollte den Nutzenden die Möglichkeit bieten, Ergebnisse zu bewerten.
- A11** Das Plugin sollte den Nutzenden die Möglichkeit bieten, einen vollständig neuen Beschreibungstext zu Generieren.
- A12** Das Plugin sollte den Nutzenden die Möglichkeit bieten, eine Historie der Generierten Texte zu betrachten.

²¹Tonalität bezieht sich auf die Formulierung des Textes. Hier könnten die Ausprägungen sachlich, professionell oder verspielt möglich sein.

A13 Das Plugin sollte den Nutzenden die Möglichkeit bieten, anzugeben wo die Beschreibungstexte veröffentlicht werden.

A14 Das Plugin sollte verhindern, dass der Nutzende eine neue Generierung startet, wenn die vorherige noch nicht abgeschlossen ist.

Priorität 3: Wird

A15 Das Plugin wird den Nutzenden die Möglichkeit bieten, Informationen über den Anwendungsfall anzulegen.

A16 Das Plugin wird den Nutzenden die Möglichkeit bieten, für mehrere Produkte gleichzeitig eine Generierung zu starten.

A17 Das Plugin wird den Nutzenden die Möglichkeit bieten, bereits erstellte Beschreibungstexte als Referenz für ein Tuning zu nutzen.

A18 Das Plugin wird den Nutzenden warnen, wenn die initiale Konfiguration nicht durchgeführt wurde.

A19 Das Plugin wird den Nutzenden die Möglichkeiten bieten, den Aufbau der strukturierten Daten anzugeben.

A20 Das Plugin wird den Nutzenden die Möglichkeit bieten, Informationen, die nicht in den strukturierten Daten vorhanden sind, zu ergänzen.

A21 Das Plugin wird die Länge der Beschreibungstexte auf die vom Nutzer angegebene Länge beschränken.

4.4. Visualisierung

Vor der Implementierung des Plugins muss eine Entscheidung getroffen werden, an welcher Stelle in der Administrationsoberfläche das Plugin integriert wird. Es ist anzumerken, dass die Administrationsoberfläche von Shopware 6 auf dem firmeneigenen Meteor Design System basiert [5]. Mit dem Ziel, den Nutzern eine konsistente Benutzererfahrung zu bieten, wird eine ähnliche Designsprache wie durch Shopware verwendet. Die Nutzung vertrauter Interaktionselemente, wie einheitlich gestalteter Schaltflächen, kann dazu beitragen, dass Nutzende sich leichter in der Oberfläche zurechtfinden. Hierfür wird auf die

Figma-Komponenten-Bibliothek²² von Shopware zurückgegriffen, um sicherzustellen, dass das Erscheinungsbild der Corporate Identity von Shopware entspricht.

Der erste Schritt dieses Prozesses besteht darin, zu evaluieren, inwieweit die Funktionalitäten in die Administrationsoberfläche integriert werden können. Es ist zu berücksichtigen, dass jede dieser Integrationsmöglichkeiten Auswirkungen auf die Struktur und die Implementierung des Plugins hat, da sich Abläufe und Zusammenhänge verändern können. Die modulare Struktur der Shopware Architektur ermöglicht es Plugins, sich an verschiedenen Stellen zu integrieren.

4.4.1. Erweiterung durch eigenständige Seite

Eine Möglichkeit besteht darin, dass eine eigenständige Seite (View) die Funktionalitäten bereitstellt. Obwohl man sich weiterhin eng innerhalb der Shopware-Administration bewegt, stellt dies die nächste Annäherung an eine eigenständige Anwendung dar. Die Nutzer würden das Plugin über einen Menüeintrag erreichen und sich ab diesem Punkt innerhalb der Plugin-Seiten bewegen. Dies bringt den Vorteil mit sich, dass Interaktionen mit verschiedenen View-Wechseln realisiert werden könnten, was bei den anderen Möglichkeiten nicht der Fall ist oder schwieriger umzusetzen wäre.

Diese Option bietet sich an, wenn die Funktionalitäten über eine Startseite erklärt und eingeführt werden sollen, oder wenn die Konfiguration und Generierung über zwei eigenständige Seiten realisiert werden sollen.

Ein Problem, das sich hierbei herauskristallisiert hat, ist der fehlende Kontext. Da die Anwendung in dieser Variante isoliert aufgebaut ist, müssen Zusatzinformationen, wie das betroffene Produkt, anderweitig gelöst werden. Es muss eine Suchmaske für den Nutzer bereitstehen, über die ein Produkt gesucht und ausgewählt werden kann. Aus diesem Ansatz entstehen weitere Anforderungen, denn für eine effektive Nutzung müssen die Suchergebnisse sortiert, gefiltert und begrenzt werden können. Hierbei handelt es sich um Funktionen, welche Nutzer voraussetzen. Es gilt abzuwägen, ob die Vorteile diese komplexe Implementierung rechtfertigen. Ein weiteres Problem dieser Variante ist die starke Redundanz von Funktionalitäten. Funktionen wie Suche, Auswahl und Filterung, die bereits an anderen Stellen implementiert sind²³, würden innerhalb des Plugins lediglich erneut

²²Das Designtool Figma wird für alle Visualisierungen genutzt. Die genannte Komponenten-Bibliothek von Shopware ist hier erhältlich: <https://www.figma.com/community/file/1032569283566168066>

²³Beispielsweise in den Produktlisten in der Administrationsoberfläche.

dargestellt. Dies könnte bei den Nutzern zu Verwirrung führen, da nicht klar erkennbar ist, welche Funktionen an welcher Stelle ausgeführt werden sollen.

4.4.2. Erweiterung einer bestehenden Seite

Neben einer eigenständigen Seite besteht die Möglichkeit, bereits existierende Seiten zu erweitern, beispielsweise die Produkt-Einzelseite. Die Tab-Struktur dieser Seite ermöglicht die Integration weiterer Tabs zur Unterbringung zusätzlicher Funktionalitäten. Dies löst das zuvor genannte Problem, da die Nutzer zunächst ein Produkt aus der Produkt-Liste auswählen und anschließend weitere Prozesse, wie beispielsweise die Generierung, durchführen können. In Abbildung 11 ist die Erweiterung der Produkt-Einzelseite durch einen zusätzlichen Tab dargestellt.

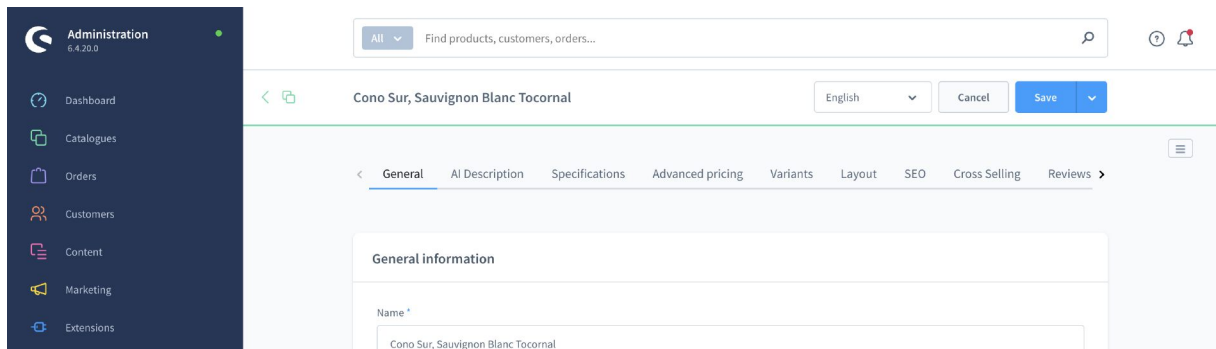
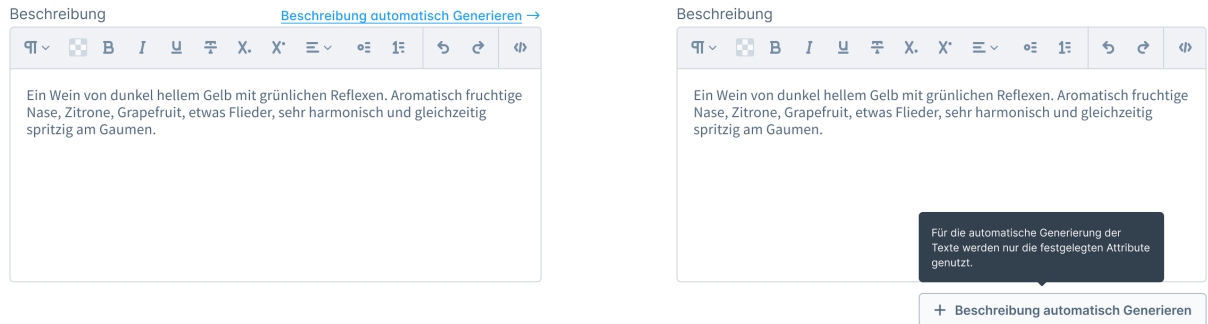


Abbildung 11: Die Tab-Struktur auf der Produkt-Einzelseite, erweitert durch einen zusätzlichen Tab *AI Description*. Eigene Darstellung.

4.4.3. Erweiterung bestehender Komponenten

Abbildung 12a und 12b illustrieren die letzte Möglichkeit, die betrachtet wird. Hierbei wird die Nutzungserfahrung mit minimalen Änderungen erweitert, beispielsweise durch die Erweiterung bestehender Komponenten. Eine geeignete Komponente für die Erweiterung ist das Beschreibungstextfeld, das bereits innerhalb der Produkt-Einzelseite vorhanden ist. Die Generierung könnte über eine Schaltfläche oder Ähnliches direkt innerhalb des Textfeldes ausgelöst werden. Allerdings kann diese Möglichkeit nur als Erweiterung einer der vorherigen angesehen werden. Für andere Anforderungen an das Plugin, wie Einstellungen oder individuelle Vorgaben bezüglich der Generierung, muss an anderer Stelle eine Ansicht vorhanden sein. Darüber hinaus kann ein Kompatibilitätsproblem entstehen, da die Erweiterung von Shopware Standardseiten und Standardkomponenten durch andere



(a) Erweiterung durch eine Schaltfläche oberhalb des Textfeld

(b) Erweiterung durch eine Schaltfläche unterhalb des Textfeld und eine Erklärung der Funktionalität

Abbildung 12: Das Standard-Textfeld aus Shopware, erweitert durch eigene Funktionalitäten. Eigene Darstellung.

Plugins nicht ausgeschlossen werden kann. Dies stellt ein unnötiges Risiko dar, da die beiden vorherigen Ansätze problemlos die Funktionalitäten bereitstellen können.

Nach Abwägung der Vor- und Nachteile der jeweiligen Möglichkeiten wurde die Entscheidung für die Erweiterung der Einzelseite getroffen. Obwohl die eigenständige Lösung ebenfalls für den Anwendungsfall geeignet wäre, hat das Feedback der Redakteure deutlich gemacht, dass ein Vorgehen bevorzugt wird, das den bisherigen Prozess²⁴ minimal bis gar nicht verändert. Dementsprechend erweitert die Modifikation der Produkt-Einzelseite den bisherigen Vorgang lediglich um eine optionale Komponente, die über einen zusätzlichen Tab erreichbar ist.

In Abbildung 13 ist der endgültige Entwurf des Plugins dargestellt. Es wurde sich entschieden, für das Anzeigen der generierten Beschreibung, auf das Textfeld von Shopware zurückzugreifen. Dies ermöglicht es Redakteuren, den Beschreibungstext direkt nach der Generierung zu bearbeiten. Neben der generierten Beschreibung wird die ursprüngliche Beschreibung ebenfalls angezeigt. Das Darstellen der ursprünglichen Beschreibung mag zwar redundant erscheinen, bietet jedoch den Vorteil, dass Redakteure an der gleichen Stelle Vergleiche anstellen können. Die Alternative würde bedeuten, dass Redakteure zu einem anderen Tab wechseln müssten, um die ursprüngliche Beschreibung zu sehen. Die Rücksprachen innerhalb des Unternehmens haben ergeben, dass für einen Großteil der Produkte bereits Texte vorhanden sein werden. Hier besteht die Möglichkeit, dass sich Redakteure von vorherigen Texten inspirieren lassen wollen, oder diese zur Prüfung der

²⁴Siehe Kapitel 2.4: „Redaktionsprozess bei der Integration von Produkten“

generierten heranziehen müssen.

Alle Visualisierungen wurden ausschließlich für die Desktop-Variante durchgeführt. Das Shopware 6 Administrations-Frontend ist mobil nutzbar. In der mobilen Variante werden alle Elemente untereinander dargestellt. Dies lässt sich ohne Probleme im geplanten Plugin realisieren und muss nicht zusätzlich visualisiert werden.

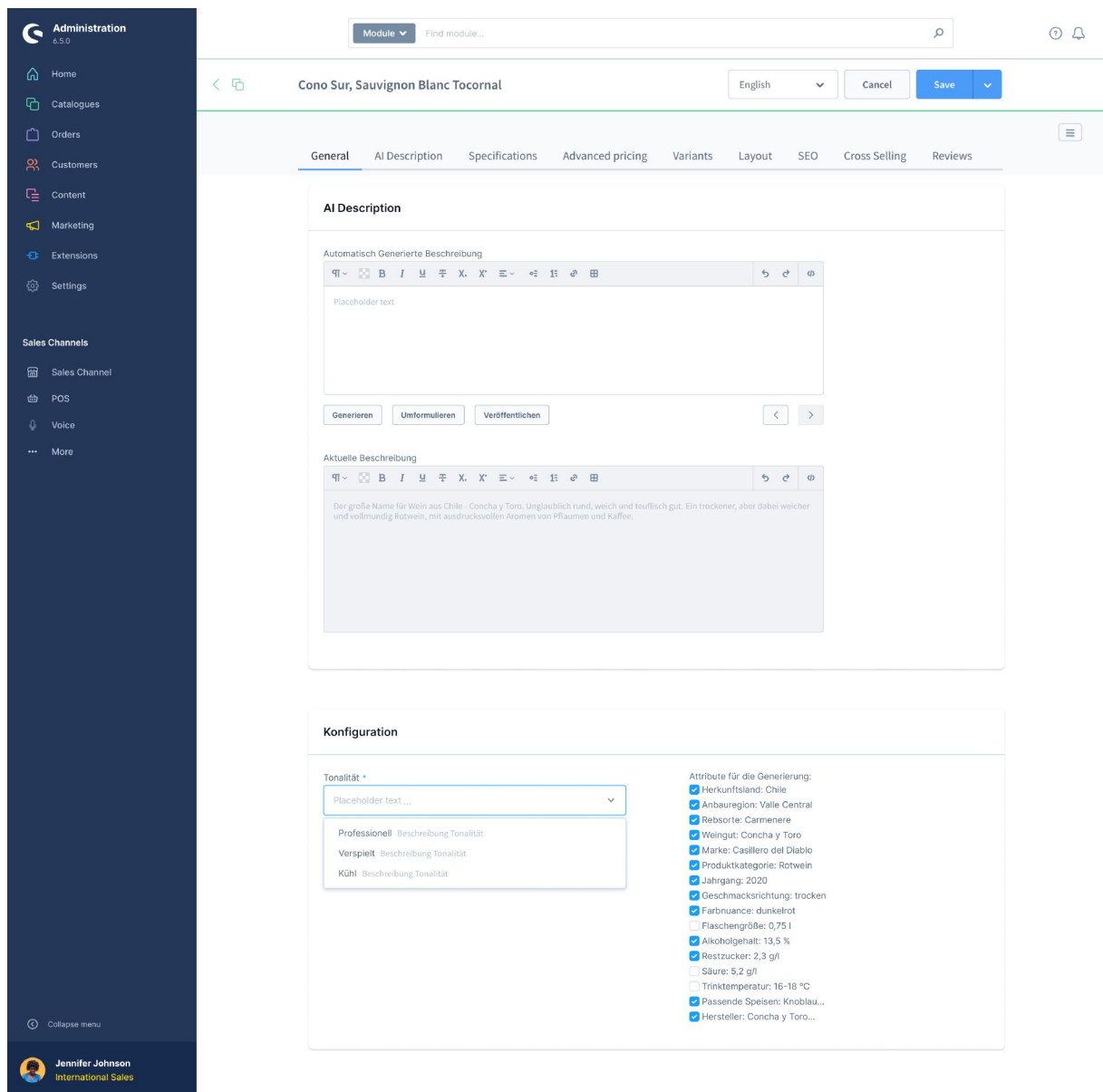


Abbildung 13: Die Visualisierung des geplanten Plugins. Eigene Darstellung.

4.5. Bearbeiten der Beschreibung

Das Bearbeiten und das Austauschen von Teilen der Beschreibungstexte stellen eine Interaktion dar, die nicht trivial ist. Diese Interaktion muss gesondert betrachtet werden.

Nutzer müssen die Möglichkeit haben, die generierte Beschreibung zu bearbeiten und zu ergänzen. Hierunter fallen verschiedene Funktionen. Einerseits müssen Standardfunktionen bereitstehen, welche bei einem Textfeld zu erwarten sind. Das sind klassische Textformatierungen wie fett, kursiv, unterstrichen oder das Ändern der Formatierung wie Überschriften, Paragraphen und Listen. Man spricht hier von einem What-You-See-Is-What-You-Get-Editor (WYSIWYG-Editor).

Andererseits muss eine Möglichkeit bereitstehen, über die die Nutzer angeben können, welche Passagen durch das LLM umformuliert werden sollen. Da für den Nutzer nicht erkennbar ist, ob er eine Standardfunktion von einem Editor nutzt oder eine Erweiterung durch das Plugin, sollen alle Interaktionen im Editor durchführbar sein. Dies gewährleistet eine nahtlose und konsistente Benutzererfahrung, bei der die Nutzer alle erforderlichen Aktionen direkt innerhalb der zu erwartenden Komponente ausführen können. Innerhalb des Editors werden Änderungen am Text wie folgt durchgeführt: Der Nutzer markiert einen Teil des Textes mit einem Eingabegerät seiner Wahl, daraufhin kann über die Menüleiste im Editor eine Aktion gewählt werden. Nach dem Auswählen einer Aktion sieht der Nutzer direkt die angewendete Änderung. Änderungen betreffen nur die markierten Passagen. Das Umformulieren soll auf ähnliche Art initiiert werden können. Abbildung 14 zeigt die erste Möglichkeit: Hier werden dem Nutzer, nach dem Markieren einer Textstelle, über ein PopUp die möglichen Funktionen angezeigt.

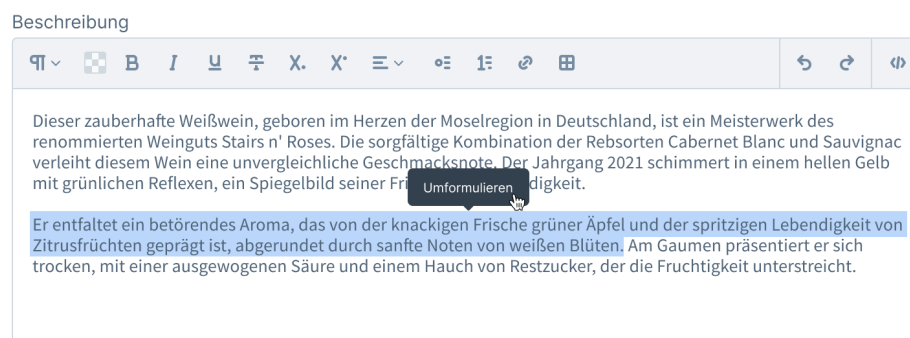


Abbildung 14: Der Shopware WYSIWYG-Editor, ergänzt durch ein PopUp für das Umformulieren. Im Fließtext ist eine Passage markiert, die für die Umformulierung ausgewählt ist. Eigene Darstellung.

Das PopUp muss automatisch angezeigt werden, sobald eine Passage markiert ist. Das bringt einerseits den Vorteil, dass die Funktion nicht eingeführt werden muss, denn der Nutzer erkennt, an welcher Stelle er eine Umformulierung anstoßen kann. Andererseits

besteht die Gefahr, dass der Nutzer gestört wird, wenn er keine Umformulierung beabsichtigt, sondern eine Formatierung vornehmen möchte. Neben der Funktionsanzeige über ein PopUp wird ebenfalls eine Ergänzung der Menüleiste im Editor in Betracht gezogen. Um eine konsistente Benutzererfahrung zu gewährleisten und keine anderen Funktionen zu blockieren, wird diese Option bevorzugt. Es muss jedoch beachtet werden, dass die Erweiterung der Menüleiste den Nutzern explizit gezeigt werden muss, da sie minimal ist. Zusätzlich zum Auswählen von Umformulierungen muss für den Nutzer jederzeit erkennbar sein, welche Passagen er für eine Umformulierung markiert hat. Eine visuelle Komponente, wie beispielsweise eine Änderung der Textfarbe oder das Durchstreichen der betreffenden Passage, wäre hierfür geeignet. Allerdings sind die meisten visuellen Möglichkeiten bereits durch Standardfunktionen des Editors belegt. Würde man eine dieser Möglichkeiten für die Markierung der Passage verwenden, wäre dies für den Nutzer nicht nachvollziehbar. Eine der wenigen nicht vorhandenen Funktionen im Editor von Shopware ist das Setzen einer Hintergrundfarbe. Dieses eignet sich daher für das Darstellen der Passagen, welche für eine Umformulierung gewählt wurden. In Abbildung 15 ist eine markierte Textpassage dargestellt, daneben ist die neue Funktion in der Menüleiste erkennbar.

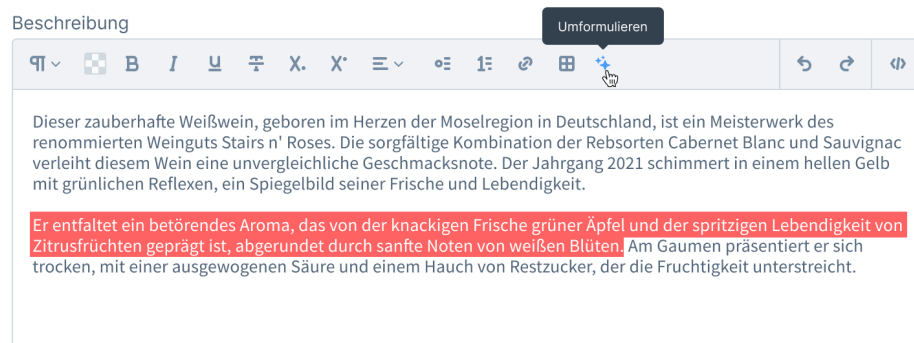


Abbildung 15: Der Shopware WYSIWYG-Editor, ergänzt durch einen Menü-Eintrag für das Umformulieren. Im Fließtext ist eine Passage markiert, die für die Umformulierung ausgewählt ist. Eigene Darstellung.

4.6. Architektur Entwurf

Resultierend aus der Shopware-Architektur ist das Plugin in zwei Komponenten aufgeteilt, das Plugin-Backend und das Plugin-Frontend. Es ist jedoch zu beachten, dass das

Frontend technisch gesehen aus zwei eigenständigen Komponenten besteht: Die im vorherigen visualisierte Erweiterung der Produkt-Einzelseite und die Einstellungsseite. Die Einstellungen von Plugins werden in Shopware über einen spezifischen View verwaltet. Dieser wird automatisch erstellt und kann visuell nicht beeinflusst werden.

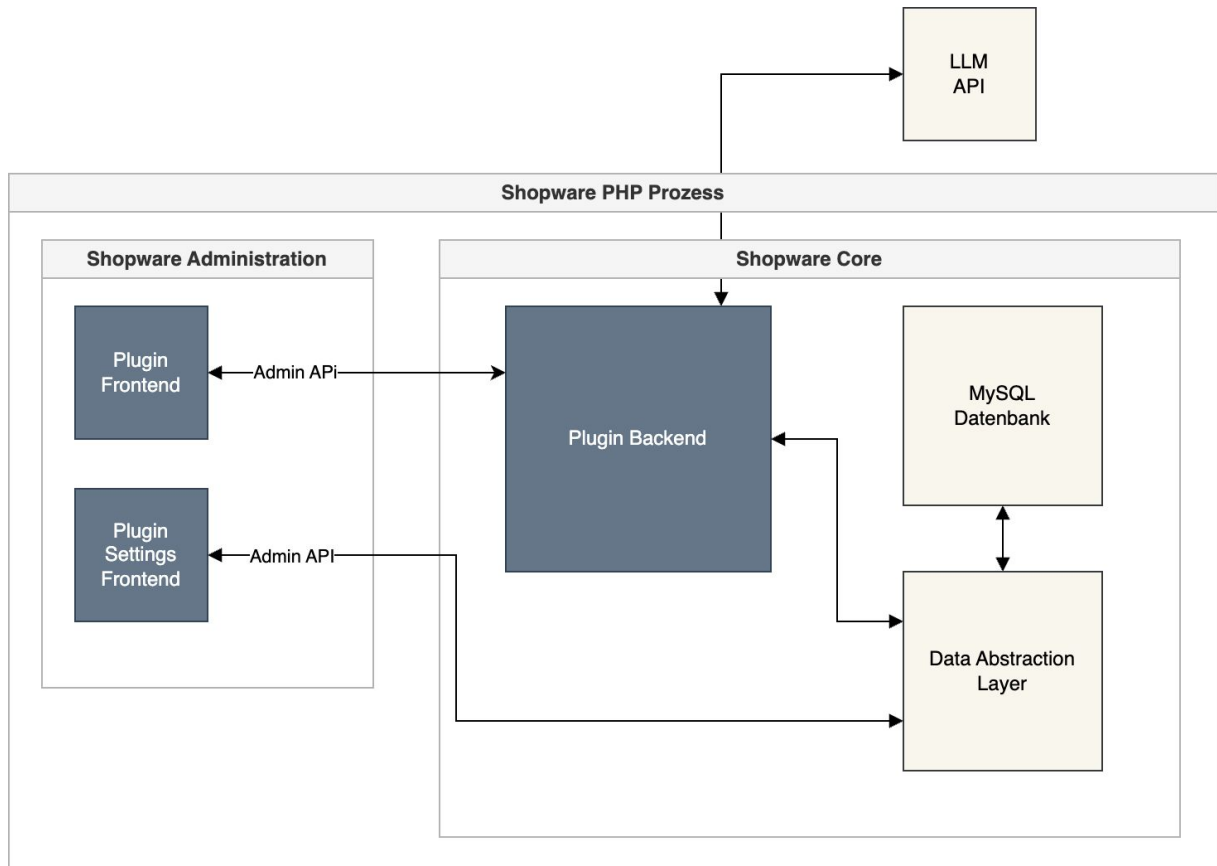


Abbildung 16: Der Architektur Entwurf für das Plugin. Eigene Darstellung.

Abbildung 16 zeigt die relevanten Komponenten, welche in folgender Auflistung erläutert werden:

- Das **Plugin Frontend** beinhaltet die Views, welche für das Plugin geschaffen werden. Hier findet die Nutzerinteraktion mit dem Plugin statt. Es enthält keinerlei Business-Logik, sondern kommuniziert ausschließlich über die Admin-API mit dem Plugin-Backend.
- Über das **Plugin Settings Frontend** können Nutzer globale Einstellungen an dem Plugin vornehmen. Diese Seite wird durch Shopware automatisch anhand einer Konfigurationsdatei gerendert.
- Das **Plugin Backend** enthält die vollständige Business-Logik der Anwendung. Hierunter fallen das Erweitern der Admin API für das Frontend, die Kommunikation mit

dem LLM und das persistente Speichern der Daten.

- Das **Data Abstraction Layer** ist eine Shopware eigene Komponente, die sowohl vom Plugin Settings Frontend als auch vom Plugin Backend konsumiert wird. Sie ist bei jeder CRUD-Operation auf der MySQL-Datenbank zwischengeschaltet.

4.7. Datenmodell

Für die Erfüllung von Anforderung A12 wird zwingend eine Datenhaltung benötigt. Das Datenmodell muss mindestens die erstellten Beschreibungen umfassen, um eine Historie zu ermöglichen. Neben der Beschreibung werden zusätzlich die Einstellungen für die Generierung sowie eine Bewertung (*evaluation*) gespeichert. Die Einstellungen sind aufgeschlüsselt in Informationen über die genutzte Prompt, die genutzte LLM Konfiguration, die gewünschte Tonalität und die genutzten Attribute. Das Speichern der zusätzlichen Informationen ermöglicht es, diese dem Nutzer beim Betrachten der Historie anzuzeigen. Dies kann auch für Entwickler nützlich sein, um Fehler zu identifizieren, insbesondere wenn während der Nutzung negatives Feedback von den Redakteuren zurückgemeldet wird. Jede Beschreibung ist genau einem Produkt zugeordnet, weitere Beziehungen existieren nicht. Die persistente Speicherung der globalen Plugin Einstellungen werden über die dafür vorgesehene Shopware Tabelle `system_config` umgesetzt. Die globalen Einstellungen umfassen alles, was für jede Generierung initial gesetzt ist. Zum Beispiel kann es Informationen (Attribute) geben, wie die Flaschengröße, die vom Nutzer in keinem der Beschreibungstexte gewünscht ist. Der Vollständigkeit halber ist diese Tabelle ebenfalls in Abbildung 17 dargestellt. Es ist zu beachten, dass zwischen der Tabelle `ai_description_content` und der `system_config` keine Beziehung existieren.

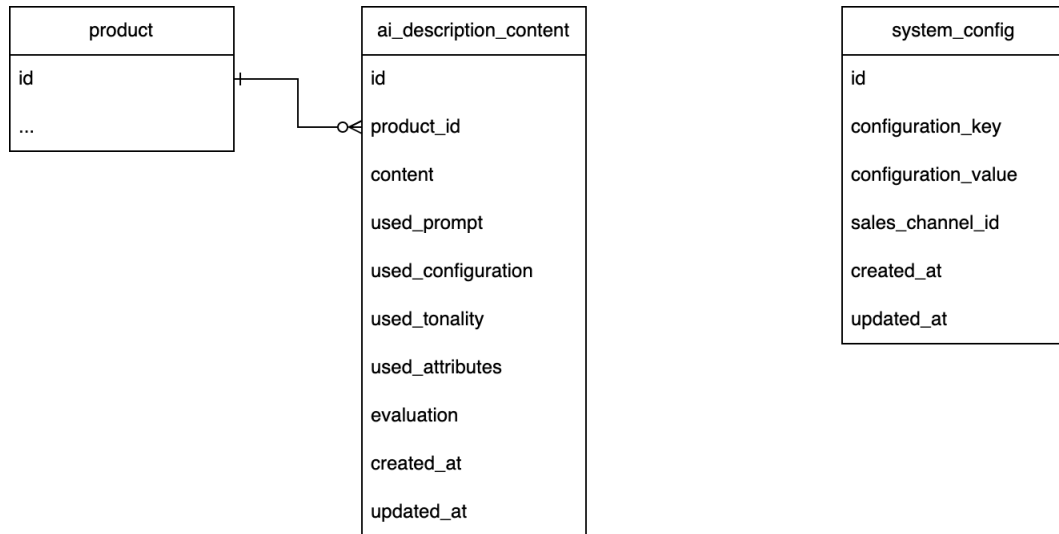


Abbildung 17: ERD Diagram für das Plugin. Die Attribute in der **product** Tabelle wurden gekürzt, da es sich hier um eine Shopware Standardtabelle handelt. Eigene Darstellung.

5. Implementierung der automatischen Lösung

Basierend auf der Auswahl aus Kapitel 3.3 werden in diesem Kapitel die Sprachmodelle verglichen, mit dem Ziel eine Entscheidung für die Implementierung zu treffen. Die Auswahl wurde in zwei Phasen durchgeführt. Im ersten Schritt wurden Interaktionen über die verfügbaren Playgrounds²⁵ gestartet. Das bringt den Vorteil mit sich, dass eine Iteration ohne initialen Entwicklungsaufwand durchgeführt werden kann. Ziel ist es, zu identifizieren, welche Modelle sich für eine weitere Untersuchung eignen. Ein Nebeneffekt dieser Tests ist, dass das initiale Prompting getestet werden kann. Im zweiten Schritt wird für die Interaktion mit der engeren Auswahl ein automatisches Tool entwickelt. Hierdurch können die Modelle parallel getestet werden und deren Ausgaben einheitlich persistent gespeichert werden. Das Ergebnis von Kapitel 5.1.2 sind die erarbeiteten Anweisungen für das Generieren und Umformulieren von Beschreibungstexten durch ein LLM. Anschließend an die Auswahl der LLMs wird das Tuning dargelegt und analysiert, mit dem Ziel zu identifizieren, ob eine Verbesserung der Ergebnisse möglich ist. Im Unterkapitel 5.3 wird dann die eigentliche Implementierung beschrieben.

²⁵Unter Playgrounds versteht man die Schnittstellen, über welche mit den Modellen interagiert werden kann. Diese sind in der Regel als Webanwendung umgesetzt.

5.1. Vergleich der passenden Sprachmodelle

Resultierend aus den Anforderungen in Kapitel 4.3 an die automatisierte Lösung sind grundlegende Fragen für das Erstellen der Prompts aufgestellt worden. Diese werden auch für das Evaluieren der Ergebnisse genutzt:

1. Wie kann die Länge der Ergebnisse beeinflusst werden? Im Kontext von Vindor existieren gewisse Vorgaben für eine optimale Länge der Produktbeschreibungen. Diese liegt bei einer Länge von 100 bis 200 Wörtern.
2. Wie kann die Formulierung der Beschreibungstexte beeinflusst werden? Vor dem Hintergrund verschiedener Zielgruppen und Produktklassen soll durch den Redakteur eine Auswahl getroffen werden können. Beispielsweise sollen manche Weine in einem sachlichen, professionellen Ton beschrieben werden, andere hingegen verspielt.
3. Wie kann verhindert werden, dass bestimmte Attribute übergeben werden, aber nicht explizit in der Produktbeschreibungen erwähnt werden? Es ist es uninteressant zu lesen, dass ein Wein 11,5% Alkoholgehalt hat. Für eine korrekte Beschreibung ist diese Informationen jedoch wichtig, um eine Aussage treffen zu können, ob ein Wein leicht bekömmlich ist oder doch mit einer sogenannten Fülle einhergeht.
4. Unterscheiden sich die Ergebnisse, wenn Daten strukturiert (*Ursprungsland: Frankreich*) oder in ausformulierten Sätzen (*Der Wein ist aus Frankreich.*) übergeben werden?
5. Wie können Teile der Ergebnisse umformuliert werden? Es kann vorkommen, dass Redakteure sich entscheiden, dass nur Teile der Beschreibung verändert werden sollen. Können Umformulierungen auch über das LLM durchgeführt werden oder muss hierfür eine andere Lösung gefunden werden?

Bei der Interaktion mit den LLMs stehen sowohl im Playground als auch bei der API Konfigurationsmöglichkeiten zur Verfügung. Diese beeinflussen die Ergebnisse und sind notwendig für den weiteren Verlauf. Neben selbsterklärenden Einstellungen, wie der maximalen Länge (`max_tokens`) der Ergebnisse oder der Anzahl der Antworten, die generiert werden sollen, gibt es eine Reihe an Einstellungen, die genauer betrachtet werden müssen. Hierunter fallen die Einstellungen `temperature`, `top_p`, `presence_penalty` sowie `frequenz_penalty`²⁶:

²⁶Es wird sich an der Stelle auf die OpenAI Dokumentation bezogen. Die Einstellmöglichkeiten sind aber

- Der `temperature` Parameter gibt an, wie das nächste Token gewählt werden soll. Hierfür werden **alle möglichen Tokens** mit in die Auswahl genommen. Ein niedriger Wert bedeutet, dass das Token mit der höchsten Wahrscheinlichkeit genutzt wird. Dies bietet sich für Ergebnisse an, die faktenbasiert sind. Ein hoher Wert dagegen sorgt dafür, dass mehr Tokens in Betracht gezogen werden, auch wenn sie unwahrscheinlicher sind. Ein hoher Wert sorgt dementsprechend für mehr Varianz im Output. Dies bietet sich daher für kreativere Ergebnisse an, wie zum Beispiel das Schreiben eines Gedichtes [37].
- Der `top_p` Parameter kontrolliert, wie der `temperature` Parameter, ebenfalls die Varianz der Ergebnisse. Hier mit dem Unterschied, wie die Auswahl gesetzt wird. Im Gegensatz zum `temperature` Parameter werden **nicht alle möglichen Tokens** betrachtet, sondern nur diejenigen, welche im angegebenen Wertebereich liegen. Ein `top_p` Wert von 0,3 gibt also an, dass nur die Tokens in Betracht gezogen werden, die zusammen die ersten 30% ausmachen. Ein niedriger Wert bedeutet dementsprechend eine höhere Diversität in den Ergebnissen, denn es existiert ein kleiner Pool an Wörtern, die möglich sind. Im Gegensatz dazu, führt ein hoher `top_p` Wert zu repetitiven Ergebnissen. Das resultiert daraus, dass alle möglichen Wörter zur Verfügung stehen und das Modell dementsprechend diejenigen wählt, die am sichersten sind [37].
- Der `presence_penalty` Parameter prüft das Vorkommen von Tokens innerhalb der Ergebnisse. Ein höherer Wert resultiert in einer höheren Variation an Worten, da das Vorkommen von gleichen Tokens **bestraft** wird [37].
- Der `frequency_penalty` Parameter prüft im Gegensatz zum `presence_penalty` Parameter, wie oft ein Token vorkommt. Ein höherer Wert verringert die Chance, dass dasselbe Token erneut gewählt wird. Es verhindert das Wiederholen von Worten und Phrasen [37].

Da sowohl der `temperature` als auch der `top_p` Parameter auf die Varianz der Ergebnisse abzielen, ist es sinnvoll, nur einen der beiden Parameter zu nutzen. Je nach Modellreihe wird einer der beiden Parameter zuerst angewendet. Für konsistente Ergebnisse wird daher vermieden, beide Parameter gleichzeitig anzupassen.

5.1.1. Manuelle Interaktion

Neben den Cloud-basierten LLMs, von denen jedes einen eigenen Playground bietet, wird auch das Open-Source-LLM LLaMa geprüft. Hierfür wird der Vercel AI Playground²⁷ verwendet. Ein Interface, das von der Firma Vercel entwickelt wurde und die Interaktion sowohl mit Cloud-basierten als auch mit Open-Source-Modellen ermöglicht. Dieses Interface ist so gestaltet, dass die Modelle direkt in einem geteilten Bildschirm miteinander verglichen werden können [38]. Es wird darauf verzichtet, für jedes Modell Beispielantworten zu zeigen. Dies würde zu starken Redundanzen führen und keinen Mehrwert bieten. Stattdessen wird dargelegt, ob die Modelle Antworten generieren, die den Anforderungen entsprechen.

Innerhalb der verschiedenen Playgrounds gibt es signifikante Unterschiede. Es muss unterschieden werden, zwischen denen, die nur eine vereinfachte Interaktion über ein Input-Feld zulassen und denjenigen, die den Nutzern die Möglichkeit bieten, Justierungen an den Einstellungen vorzunehmen.

Bard Das Bard Model von Google fällt in die erste Gruppe der LLMs mit vereinfachter Interaktionsmöglichkeit. Hier gibt es keine Möglichkeiten, Einstellungen an dem Model vorzunehmen. Dies kann verschiedenen Gründe haben, zum einen ist das Modell erst seit Mitte 2023 verfügbar und damit noch in der Beta-Phase²⁸. Zum anderen handelt es sich hierbei um ein sogenanntes *Language Model for Dialogue Applications* und ist damit laut Google bereits optimiert auf Dialoge ohne zusätzliche Konfiguration. In Anhang C.I ist eine Beispielantwort für den Wein *Concha y Toro, Casillero del Diablo Carmenere, Valle Central*²⁹ dargelegt. Im Vergleich mit der ursprünglichen Beschreibung sind die Aromen, wie Pflaume und Kaffee, treffen beschrieben. Die Ergebnisse dieses Models würden die Anforderungen erfüllen. Dennoch ist es für den Anwendungsfall in dieser Arbeit ohne Einstellungsmöglichkeiten und ohne API nicht nutzbar³⁰. Das Bard Modell wird für eine weitere Untersuchung ausgeschlossen.

²⁷Siehe: <https://sdk.vercel.ai/>

²⁸Google nutzt statt Beta-Phase die Formulierung „Experiment“

²⁹Siehe: <https://www.vindor.de/concha-y-toro-casillero-del-diablo-carmenere-valle-central>

³⁰Das Modell PaLM 2, welches im Hintergrund von Bard läuft, verfügt zwar über einen API-Zugang, jedoch wurde dieser dem Autor nicht zugänglich gemacht.

LLaMa Mit dem OS-Modell LLaMa kann nur über eine lokale Instanz oder die Vercel Playground interagiert werden. Es wurde sich nur für eine Interaktion über den Vercel Playground entschieden. Hierbei entsteht der Vorteil, dass die Modelle für eine Interaktion nicht heruntergeladen und konfiguriert werden müssen. Die Ergebnisse der initialen Tests sind gemischt. Hauptproblem mit dem Model ist das Erzwingen von deutschen Antworten. Obwohl das Model sowohl Eingaben und Ausgaben in Deutsch verstehen und generieren kann, ist in den meisten Fällen ein englischer Text erstellt worden ³¹. Daneben wurden die Anforderungen an die Beschreibungstexte nur teilweise erfüllt. Anweisungen über die Formulierung, wie sachlich oder verspielt, wurden nicht beachtet oder konnten durch den Autor nicht nachvollzogen werden. Ebenfalls konnte die Länge der Ergebnisse nicht durch Anweisungen gesteuert werden. Ein weiterer nicht zu vernachlässigender Aspekt ist die zusätzliche Komponente, welche die Nutzung von OS-Modellen einführt. Der Architekturaufbau muss um einen zusätzlichen Dienst erweitert werden, der das Modell zur Verfügung stellt und eine API anbietet. Dies geht mit zusätzlichem Wartungsaufwand und Kosten einher. In Anbetracht der alternativen Modelle wurde sich gegen eine weitere Nutzung von LLaMa entschieden.

Luminous Bei der Luminous Reihe wurde sich nur auf die Modelle konzentriert, die für das Zero-Shot prompting optimiert sind. Das hat den Grund, dass sich aus dem Anwendungsfall keine Few-Shot Prompts erstellen lassen³². Sowohl die Base- als auch die Supreme-Varianten liefern vielversprechende Ergebnisse, die den Anforderungen gerecht werden. Allerdings neigt die Base-Variante im Vergleich zur Supreme-Variante dazu, Sätze ähnlich zu beginnen und diese lediglich aneinanderzureihen, was den Lesefluss weniger ansprechend gestaltet. Für die nähere Auswahl wird nur das Supreme-Model genutzt.

GPT-3 & GPT-4 Resultierend aus den Interaktionen mit der Luminous Reihe wird sich bei den OpenAI Modellen erst auf das text-davinci-003 (GPT-3) Modell konzentriert, da davon ausgegangen wird, dass Modelle mit einer höheren Trainingsparameter-Anzahl auch besserer Ergebnisse liefern. Innerhalb dieser Arbeit ist die Verwendungsempfehlung für das text-davinci-003 Modell ausgelaufen, dieses Modell wird durch OpenAI Anfang

³¹Siehe Anhang: C.II

³²Siehe Kapitel 3.2

2024 abgeschaltet³³ [37]. Da ebenfalls erst zur Mitte der Arbeit ein Zugang zur Beta-Version des GPT-4 Modells vorlag, wurde sich entschieden auf die beiden aktuellsten Modelle GPT-3.5-Turbo und GPT-4 umzusteigen. Beide aktuellen Modelle liefern vielversprechende Ergebnisse, die den Anforderungen entsprechen. Hierdurch ändert sich ebenfalls der Prompt-Aufbau. In den vorherigen Modellen wurde über die sogenannte *Completion-Funktion* mit den Modellen interagiert. Diese erwartet eine Prompt und produziert ein Ergebnis. Mit der neuen Generation wird über die *Chat-Funktion* interagiert. Die Chat-Funktion erwartet zwei Anweisungen und generiert dann eine Antwort. Die Umstellung resultiert daraus, dass mit den neuen Generation Anwendungen wie ChatGPT unterstützt werden müssen, die alternierende Unterhaltungen zwischen einem System und einem User ermöglichen.

5.1.2. Automatisierte Interaktion

Für die nun folgende automatisierte Interaktion wurde eine Eingrenzung vorgenommen. Es wird sich auf zwei Anbieter konzentriert, Aleph Alpha und OpenAI. Beide bieten den Vorteil, dass die Modelle komplett konfigurierbar sind und über eine API verfügen. Ebenfalls sind im Vergleich zu den anderen Modellen konsistente Ergebnisse erzielt worden, die in weiteren Schritten optimiert werden können.

Nach der manuellen Interaktion wird für das weitere Testen mit den ausgewählten Modellen ein automatisiertes Tool entwickelt. Das bringt den Vorteil mit, dass eine schnellere Iteration möglich ist. Ebenfalls können hiermit mehrere Modelle parallel getestet werden und deren Ausgaben einheitlich persistent gespeichert werden. Daneben ist so eine Validierung der Umsetzbarkeit möglich, bevor das Plugin fertiggestellt ist. Es wird eine Implementierung in JavaScript realisiert, der folgende Ablauf ist aber ebenso gut mit anderen Technologien umsetzbar³⁴.

Im ersten Schritt der automatisierten Interaktion werden Produkte aus dem Vindor Katalog ausgewählt. Es wurde eine zufällige Auswahl von 20 Produkten erstellt³⁵. Anschließend wird die URL-Liste gecrawlt, um die relevanten Daten für die Prompts zu extrahieren.

³³Eine Übersicht der End-of-life Informationen zu den OpenAI Modellen ist in der Dokumentation zu finden: <https://platform.openai.com/docs/deprecations>

³⁴Die hieraus resultierende Implementierung ist im Repository dieser Arbeit auf GitHub einsehbar: <https://github.com/WasMachenSachen/master-thesis-tim-loges>

³⁵Die URL-Liste ist im Begleitrepository einsehbar

Der für diese Arbeit erstellte Web-Crawler ist nur auf Vindor zugeschnitten, daher wurden hier feste DOM-Elemente angegeben, welche die benötigten Informationen enthalten. Dies verringert den Implementierungsaufwand. Die gesammelten Produktdaten werden daraufhin als CSV-Datei exportiert, um sie vorübergehend zu speichern und für weitere Verarbeitungsschritte verfügbar zu machen.

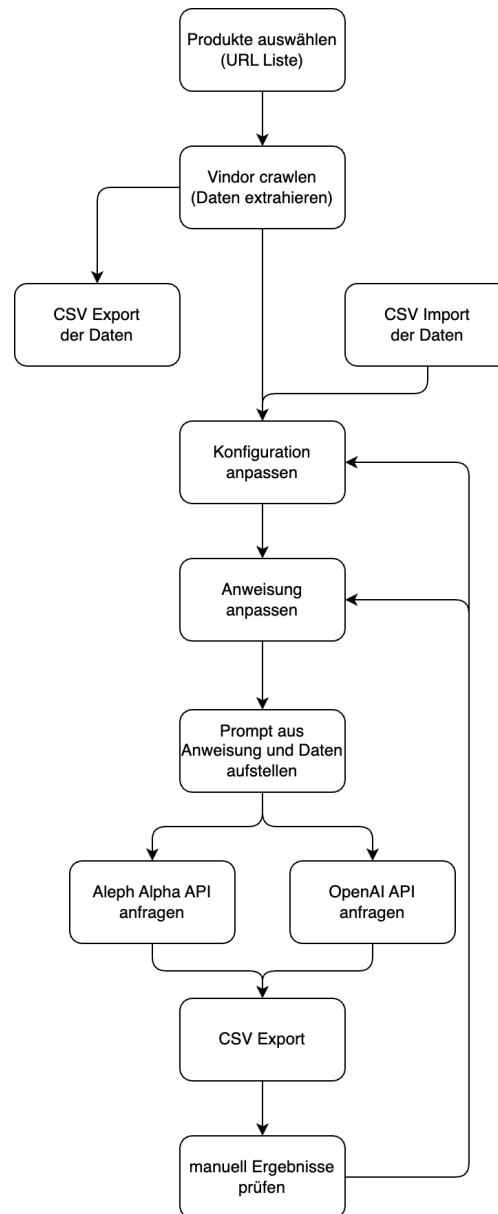


Abbildung 18: Ablauf der automatisierten Interaktion mit den LLMs. Eigene Darstellung.

Im zweiten Schritt wird die CSV-Datei eingelesen, um daraufhin einen Prompt-String zu erstellen. Durch diesen Zwischenschritt lassen sich beide Scripte unabhängig voneinander nutzen, um beispielsweise die CSV-Dateien manuell zu erstellen oder Informationen zu ergänzen. Daraufhin wird je nach genutztem Model und Konfiguration eine oder mehrere APIs mit den Informationen angefragt und die Ergebnisse der Anweisung in eine CSV-

Datei geschrieben.

Dieser Aufbau ermöglicht es, in kurzen Iterationen verschiedene Prompts und Konfigurationen mit realen Daten zu testen. Ein Nebeneffekt davon ist, dass Texte entstehen, die zur Validierung durch Redakteure und Experten herangezogen werden können. Aufgrund der engeren Auswahl an Modellen, im Vergleich zur manuellen Interaktion, werden die Ergebnisse in dieser Phase genauer untersucht. Dabei werden sowohl die Anforderungen an Beschreibungstexte wie Textlänge und Formulierung überprüft, als auch zusätzliche Kriterien festgelegt, um die Ergebnisse voneinander abgrenzen zu können.

Ein entscheidendes Element ist die Formatierung der Beschreibung. Das Ziel ist es, dass der Redakteur das Ergebnis ohne weitere Anpassungen übernehmen kann. Daher sind alle Elemente, die über einen Fließtext hinausgehen (Überschriften, Aufzählungen etc.), unerwünscht, da sie eine Anpassung durch den Nutzer erfordern würden. Die Lesbarkeit wird ebenfalls geprüft. Obwohl dieser Prozess nicht vollständig objektiv ist, wird versucht, Elemente zu identifizieren, die den Lesefluss stören können. Beispielsweise würden Texte, die jeden Satz mit dem Namen des Weines beginnen, einen schlechten Lesefluss aufweisen. Die Inhalte der Texte werden nur bedingt geprüft. Es ist nicht möglich, jede Iteration durch Experten prüfen zu lassen. Bei einigen Weinen, die in dieser Iteration verwendet werden, liegen Beispieltexte vor, in denen über die Aromen geschrieben wird. Hier ist es möglich, die Inhalte zu validieren, indem geprüft wird, ob die gleichen Aromen und Düfte sowohl im Original als auch in der automatisch generierten Beschreibung vorkommen. Dieser Prozess wird manuell und stichprobenhaft durchgeführt.

Lesefluss der Luminous Modelle: Die Hauptprobleme der generierten Texte von Luminous betreffen den Lesefluss. Es ist auffällig, dass die Texte häufig mit Formulierungen beginnen oder enden, die die gestellte Aufgabe zusammenfassen und bestätigen. Beispielsweise beginnen die Texte mit Phrasen wie *Hier ist die Beschreibung für den Wein [...]* oder enden mit Fragen wie *Möchten Sie weitere Informationen über den Wein?*

Ein weiteres Problem betrifft die Struktur der generierten Beschreibungen. Diese bestehen oft aus einer Aneinanderreihung von Sätzen, die eher an kurze Werbeslogans erinnern als an zusammenhängende Texte. Dies führt zu Formulierungen, die beispielsweise wie folgt aufgebaut sind:

Der Rockabilly Weinkult, der ist wer!

Halbtrocken, frisch und knackig, für alle, die es gerne unkompliziert mögen.

Bei diesem Wein ist alles ein Hit:

*Die Farbe, die Aromen, der Geschmack, das alles passt zu allem und zu jedem,
wirklich zu jedem.[...]*

Diese Art der Textgenerierung kann den Lesefluss stören und die Verständlichkeit der Beschreibungen beeinträchtigen.

Interessant ist in diesem Zusammenhang ein weiteres Phänomen, welches nur bei der Luminous-Reihe aufgetreten ist. Ergebnisse werden oft durch Hashtags ergänzt. Ursprünglich wurde angenommen, dass dies ein fehlgeschlagener Versuch der Markdown-Formatierung sein könnte³⁶. Bei Ergebnissen, die eine Reihe von Hashtags wie *#Weinprobe #Weinreise #Weinempfehlung #Weinkenner* enthalten, kann jedoch davon ausgegangen werden, dass diese tatsächlich als Hashtags und nicht als Überschriften gedacht sind.

Lesefluss der GPT Modelle: Im Gegensatz zu den Luminous Modellen, haben die GPT-Modelle von OpenAI weniger Schwierigkeiten, zusammenhängende Texte zu generieren. Bei diesen Modellen liegt das Hauptproblem in der Textlänge. Die Anweisungen zur maximalen Länge werden oft ignoriert, was dazu führt, dass die generierten Texte tendenziell zu lang sind.

Beeinflussung der Attribute in der Beschreibung: Die Anweisung, dass bestimmte Attribute nicht im Text auftauchen sollen, wird von den Modellen auf unterschiedliche Weise gehandhabt. Die Luminous-Reihe neigt dazu, diese Anweisungen zu ignorieren und beispielsweise dennoch über den Alkoholgehalt zu schreiben. Die GPT-Modelle hingegen umgehen die Anweisungen, indem sie vor dem generierten Text einen Einschub erzeugen, der den Text *Alkoholgehalt: 12,5 % vol* enthält. Auf diese Weise sind die Attribute technisch gesehen nicht im Fließtext enthalten, erscheinen aber dennoch in der Ausgabe.

Halluzinationen: Das Problem der Halluzinationen ist wenig bis selten aufgetreten. Soweit es möglich war zu prüfen, sind die dargestellten Aromen und Düfte sowohl im original als auch im generierten Text vorhanden.

³⁶Im Markdown-Format kennzeichnet die Raute # eine H1-Überschrift.

Weiterhin sind bei beiden Modellreihen große Schwankungen innerhalb der Qualität der Ergebnisse vorhanden, die innerhalb einer Iteration auftreten, obwohl sich die Konfiguration und die Prompt nicht ändern. In einer Iteration entstehen sowohl, ungeeignete als auch geeignete Ergebnisse. Der Zwischenfall ist selten aufgetreten.

Zusammenfassend lässt sich dennoch sagen, dass beide Modellreihen nach einer Reihe von Iterationen an Prompt und Konfiguration vielversprechende Ergebnisse liefern. Die Luminous Reihe benötigt jedoch im Vergleich zu den GPT-Modellen eine höhere Menge an Anweisungen in der Prompt um ähnliche Ergebnisse zu erreichen. Zum Zeitpunkt dieser Arbeit eignen sich die GPT-Modelle von OpenAI am besten für die Generierung von Beschreibungstexten anhand strukturierter Produktdaten. Sowohl Version 3.5 als auch Version 4 liefern die gewünschten Ergebnisse. Es liegt derzeit noch keine Information darüber vor, wann die GPT-3.5-Modelle endgültig außer Betrieb genommen werden. Es kann jedoch angenommen werden, dass das GPT-4 Modell nach Abschluss der Beta-Phase das GPT-3.5 Modell ablösen wird. Daher wird aus pragmatischen Gründen die Nutzung der aktuellen GPT-4 Modellreihe bevorzugt.

5.1.3. Weitere Erkenntnisse aus der automatisierten Interaktion

Basierend auf den Erkenntnissen aus den vorangegangenen Iterationen können im Folgenden die Fragen beantwortet werden, welche zu Beginn an die Prompt Generierung gestellt wurden.

Wie kann die Länge der Ergebnisse beeinflusst werden? Zwei Methoden haben sich als wirksam erwiesen, um die Länge der generierten Texte zu steuern: die Token-Begrenzung und die Verwendung von Prompts. Die Token-Begrenzung legt eine maximale Anzahl an Tokens fest, die das Modell nicht überschreiten darf. Es handelt sich um eine harte Grenze, die die Modelle in der Ausgabe nicht überschreiten dürfen. Hauptrisiko der Nutzung dieser Kenngröße ist das abrupte Abreißen von Texten bei Erreichen der maximalen Tokenanzahl. Dieser Fehler ist als kritisch anzusehen, da Ergebnisse mit unvollständigen Sätzen als ungeeignet anzusehen sind. Die zweite Methode besteht darin, Prompts zu verwenden, die dem Modell Kontext-Anweisungen zur Textlänge geben, wie zum Beispiel *schreibe maximal 200 Wörter* oder *der Text soll nicht länger als 10 Sätze sein*. Innerhalb der Tests hat sich ergeben, dass Anweisungen, die sich auf die Anzahl der Sätze beziehen, weniger effektiv sind als solche, die die Anzahl der Worte vorgeben.

Darüber hinaus hat sich gezeigt, dass positive Formulierungen besser funktionieren als negative. Prompts die beschreiben, was getan werden soll, sind effektiver als solche, die beschreiben, was vermieden werden soll.

Wie kann die Formulierung der Beschreibungstexte beeinflusst werden? Kontextergänzungen innerhalb der Prompt führen zum gewünschten Ergebnis. Spezifische Formulierungen wie beispielsweise *schreibe den Text verspielt* resultieren mit dem GPT-4 Modell in folgender Ausgabe: *Der „Sweetest Devotion“ ist ein Perlwein, der das Herz höher schlagen lässt. Mit seinem süßen Charakter und den feinen Perlen ist er ein wahrer Genuss. Aromen von reifen Äpfeln und Honig verzaubern die Sinne..* Die Anweisung *formuliere die Beschreibung sachlich und professionell* bewirken signifikante Veränderungen in der Ausgabe: *Der „Sweetest Devotion“ von Stairs n’ Roses ist ein edelsüßer Perlwein, der mit seiner feinen Perlage und Aromen von reifen Äpfeln und Pfirsichen begeistert. Ein Genuss für die Sinne, ideal als Aperitif oder Dessertbegleiter.*

Wie kann verhindert werden, dass bestimmte Attribute übergeben werden, aber nicht explizit in der Produktbeschreibungen erwähnt werden? Auch hierfür können die Kontextergänzungen in den Prompts genutzt werden. Zum Beispiel ermöglichen Anweisungen wie *Attribute X und Attribute Y sollen nicht in der Beschreibung vorkommen* teilweise, dass diese nicht enthalten sind. Trotz dieses Ansatzes konnte eine vollständige Prävention nicht zuverlässig erreicht werden. Eine Verbesserung wurde durch die Aufteilung der Attributliste in zwei Gruppen erzielt: Eine für diejenigen, die eingeschlossen werden sollen, und eine für diejenigen, die nicht vorkommen sollen. Dennoch kann nicht vollständig verhindert werden, dass unerwünschte Attribute auftreten. Angesichts der Tatsache, dass zusätzliche Informationen zu qualitativ hochwertigeren Texten führen, wird dieser Umstand akzeptiert. Im Zweifelsfall ist es erforderlich, dass der Redakteur eine neue Beschreibung generiert.

Unterscheiden sich die Ergebnisse, wenn Daten strukturiert (*Ursprungsland: Frankreich*) oder in ausformulierten Sätzen (*Der Wein ist aus Frankreich.*) übergeben werden? Diese Frage konnte nicht eindeutig beantwortet werden. Die generellen Qualitätsschwankungen in einzelnen Iterationen haben es erschwert, nachzuvollziehen, ob eine bessere Lesbarkeit in Korrelation mit dem Ausformulieren der Attribute steht. Sowohl Anweisungen mit ausformulierten Sätzen als auch in strukturierter Form haben zu gleich-

wertigen Ergebnissen geführt. Im Weiteren wird mit der strukturierten Form weitergearbeitet. Dies hat den Vorteil, dass Daten nicht verändert werden müssen, da sie bereits strukturiert vorliegen.

Wie können Teile der Ergebnisse umformuliert werden? Es ist möglich über eine ergänzende oder neue Anfrage an das LLM Ergebnisse umzuformulieren. Hier müssen zwei Punkte beachtet werden: Einerseits muss der ursprüngliche Text in den Eingabedaten übergeben werden. Andererseits muss eine Möglichkeit vorhanden sein, den umzuformulieren Text zu markieren, damit das LLM erkennen kann, welche Textpassagen betroffen sind. Sind diese Punkte beachtet, kann über eine neue Prompt wie *formuliere den Text innerhalb des markierten Bereichs um, der Rest soll gleich bleiben* eine Umformulierung initiiert werden.

Resultierend aus den vorangegangenen Tests sowie den beantworteten Fragen ist folgende Prompt entstanden, die für das Generieren von Beschreibungstexten zuverlässige und den Anforderungen entsprechende Resultate ermöglicht:

Du bist ein Experte für Wein und sollst eine Beschreibung für einen Wein erstellen. Du formulierst die Texte *Tonalität*³⁷. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll 200 Wörter lang sein. Es soll eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung auf.

Hier sind die Informationen über den Wein:

Liste der strukturierten Daten

Die folgenden Informationen sind auch über den selben Wein, sollen aber nicht explizit in der Beschreibung genannt werden.

Nutze sie um den Wein besser zu verstehen:

Liste der strukturierten Daten, die der Redakteur ausgewählt hat.

Für das Umformulieren von einer oder mehreren Textpassagen wird die nun folgende Prompt verwendet. Die erste Anweisung (*Lese den folgenden Text*) mit einem darauffolgenden Text ist zwingend erforderlich. Sie ermöglicht es, einen neuen Text zu referenzieren,

³⁷Die Tonalität wird durch den Redakteur gesetzt.

ohne eine neue Interaktion zu starten. Es ist nicht sinnvoll, sich auf den vorher generierten Text zu beziehen. Ein Redakteur kann einen generierten Text mit Ergänzungen erweitern, die vorher nicht vorhanden waren. Ebenfalls könnte ein Redakteur Formatierungen am Text vorgenommen haben, die erhalten werden sollen. Daher sollte bei jeder Umformulierung der betroffene Text übergeben werden. Des Weiteren ist zu erkennen, dass die Passagen, welche zur Umformulierung gewählt worden sind, in einem `span` Element mit einem `data-` Attribut liegen. Hierbei handelt es sich, um eine der möglichen Varianten Passagen zu markieren. Es kann sich in der Implementierung herausstellen, dass hierfür ein anderer Ansatz sinnvoller ist.

Lese den folgenden Text:

„Ursprüngliche Beschreibung eines Weines. Die über mehrere Sätze gehen kann. Sie kann HTML-Elemente enthalten Sätze die neu Formuliert werden sollen sind mit einem span Element umschließen.“

Formuliere den Satz innerhalb der span Elemente mit dem data-change Attribute neu, in einem sachlichen Ton. Der Rest des Text soll gleich bleiben. Behalte die HTML Formatierung bei. Entferne in der Ausgabe das data-change Attribute.

Bei beiden vorgestellten Prompts, welche im Plugin Verwendung finden, werden zusätzlich die folgenden Einstellungen innerhalb des GPT-4 Modells gesetzt:

- Der `temperature` Parameter wird in einem Bereich von 0,4 bis 0,6 gesetzt. Ein zu hoher Wert erzeugt Ausgaben, die sich nicht mehr an die übergebenen Daten halten.
- Der `top_p` Parameter wird, wie empfohlen, nicht genutzt, da bereits der `temperature` Parameter verwendet wird. Ebenfalls konnte kein Unterschied ausgemacht werden, wenn statt dem `temperature` Parameter der `top_p` Parameter gesetzt wurde.
- Die Parameter `presence_penalty` und `frequency_penalty` werden beide auf 0,2 gesetzt. Dies verhindert vor allem das häufige Wiederholen der Namen und Regionen in den Beschreibungstexten.
- Der `max_token` Parameter ist auf mindestens 500 gesetzt. Die Begrenzung der Länge lässt sich, wie bereits dargelegt, über Anweisungen kontrollieren. Ein hoher Wert verhindert, dass Sätze plötzlich abgeschnitten werden.

Vorgänger der dargelegten Prompt sind in Anhang D abgelegt. Hierbei wurden nur Änderungen beschrieben, die nicht nur einzelne Worte betreffen. Alle geprüften Prompts darzustellen würde keinen Mehrwert bieten. Des Weiteren ist eine umfassende Auflistung einzelner Antworten der Modelle zu erkennen. In Anhang E sind Antworten der Modellreihe Luminous und GPT. Ebenfalls sind die Beispieltex te für die Evaluierung in Anhang G.IV, diese sind durch das GPT-4 Modell mit der dargelegten Prompt und der dargelegten Konfiguration entstanden. Dennoch musste eine exemplarische Auswahl getroffen werden, da bei mindestens sechs Prompt Iterationen für 20 Produkte mit vier Modellen eine Anzahl von fast 500 generierten Beschreibungstexten entstanden ist.

5.2. Tuning des GPT-3.5-Turbo Modells

An dieser Stelle wird das Tuning des GPT-3.5-Turbo-Modells dargelegt. Unter *Tuning* versteht man die Feinjustierung eines Modells auf einen bestimmten Anwendungsfall. Von den in dieser Arbeit betrachteten Modellen, sind die Modelle von OpenAI die Einzigen, die eine solche Feinjustierung ermöglichen. Obwohl die GPT-Modelle von OpenAI bereits gute Ergebnisse erzielen, ist es interessant zu untersuchen, ob sich die Ergebnisse durch ein Tuning weiter verbessern lassen. Ein weiterer Nebeneffekt könnte eine Reduzierung der Ausführungskosten sein. Nach dem Tuning werden in der Regel weniger Informationen benötigt, um das gleiche Ergebnis zu bekommen. Dabei werden auch weniger Tokens übertragen, was die allgemeinen Kosten senkt [37].

Das Tuning gibt im Wesentlichen dem Modell Beispiele dafür, welche Ergebnisse bei einer Anfrage gewünscht sind. Hierbei handelt es sich um das gleiche Vorgehen, wie beim Few-Shot Prompting, mit dem Unterschied, dass über ein Tuning mehr Beispiele gegeben werden können. Dies bedeutet jedoch auch, dass für ein erfolgreiches Tuning eine hohe Anzahl hochwertiger Beispiele erforderlich ist.

OpenAI empfiehlt, das Tuning in mehreren Iterationen durchzuführen, wobei in jeder Iteration die Anzahl der Beispiele erhöht werden kann. Auf diese Weise wird schrittweise ermittelt, wann die Anzahl der Beispiele ausreichend ist, um die gewünschte Verbesserung zu erzielen. Für das Tuning in dieser Arbeit werden drei Iterationen durchgeführt, mit jeweils 200, 500 und 850 Beispielen.

Aufgrund der großen Anzahl von Produkten im Vindor-Katalog ist es nicht möglich, alle Beschreibungen manuell zu überprüfen. Eine stichprobenhafte Prüfung von 20 Produkten ergab gemischte Ergebnisse hinsichtlich der Qualitätsanforderungen. Es gibt eine Reihe von Beschreibungen, die sehr gut sind, da sie von Redakteuren verfasst wurden. Andererseits gibt es auch Beschreibungen, die sehr kurz sind oder nur aus Pressebewertungen bestehen. Es muss berücksichtigt werden, dass dies das Tuning beeinflussen könnte.

Das Tuning läuft in drei Schritten ab: Zu Beginn wird ein Datensatz erstellt, dieser wird daraufhin zu OpenAI hochgeladen, um dann im letzten Schritt diesen Datensatz für das Tuning zu nutzen.

Der Datensatz muss wie in Quellcode 1 aufgebaut sein. Es ist zu erkennen, dass Parallelen zur allgemeinen Anfrage an das GPT-Modell vorhanden sind. Des Weiteren muss der Datensatz im JSONL³⁸ Format vorliegen. Hierfür wurde das Skript aus Kapitel 5.1.2

³⁸JSON Lines: <https://jsonlines.org/>

angepasst, um aus den Informationen von Vindor den benötigten Aufbau aus Quellcode 1 zu erhalten. Das hieraus resultierende Skript und die Funktionen für das Hochladen und Initiieren des Tunings sind im Begleitrepository einsehbar³⁹.

```
1 {  
2   "messages": [  
3     {  
4       "role": "system",  
5       "content": "PROMPT"  
6     },  
7     {  
8       "role": "user",  
9       "content": "Zweite PROMPT mit strukturierten Daten"  
10    },  
11    {  
12      "role": "assistant",  
13      "content": "Ursprüngliche Beschreibung von Vindor"  
14    }  
15  ]  
16 }
```

Quellcode 1: Ein Beispieldatensatz für das Tuning.

Alle drei Trainingsdurchläufe mit erhöhtem Beispiel Anzahl führen zu unbefriedigenden Ergebnissen in den Ausgaben. Die nun beschriebenen Probleme sind bei allen Tuning Durchgängen aufgetreten. Eine Erhöhung der Beispiellanzahl hat keine Verbesserungen ergeben. Im Folgenden werden zwei Beispiele dargelegt, beide mit der gleichen Konfiguration⁴⁰. Das erste Beispiel ist durch das Standard GPT-3.5-Turbo Modell erstellt worden, das zweite Beispiel durch das getunte Modell mit 850 Beispielen

Antwort des GPT-3.5-Turbo Modell:

Der 2020 Stairs n' Roses Sweetest Devotion ist ein edelsüßer Perlwein aus dem renommierten Weingut Stairs n' Roses, gelegen in der malerischen Moselregion in Deutschland. Dieser Qualitätsperlwein ist eine Hommage an die

³⁹Siehe: <https://github.com/WasMachenSachen/master-thesis-tim-loges>

⁴⁰Es wurde die Prompt und die Konfiguration aus Kapitel 5.1.2 genutzt, sowie die strukturierten Daten des Weins 2020 Stairs n' Roses „Sweetest Devotion“ (Siehe Anhang G.IV.3).

traditionelle Rebsorte Riesling, die für ihre Vielseitigkeit und ihren charakteristischen Geschmack bekannt ist. Der Jahrgang 2020 präsentiert sich in einer ansprechenden gelben Farbnuance, die die Sinne auf ein unvergessliches Geschmackserlebnis vorbereitet. In der Nase entfaltet dieser Wein ein verführerisches Bouquet, das von der typischen Fruchtigkeit des Rieslings dominiert wird. Am Gaumen zeigt sich der Sweetest Devotion edelsüß, was ihn zu einem besonderen Genuss für Liebhaber süßer Weine macht. Seine feine Perlage sorgt für ein prickelndes Geschmackserlebnis und unterstreicht die Eleganz dieses Weines.

Antwort des getunten Modells:

Der Sweetest Devotion ist ein Riesling, der Spaß macht. Ein süffiger, fruchtiger und spritziger Riesling mit feinen Aromen. Am Gaumen zeigt er sich saftig und ausgewogen mit einer angenehmen Restsüße. Ein unkomplizierter und zugänglicher Riesling, der zu vielen Gelegenheiten passt.

Es besteht vor allem das Problem, dass die Anweisungen, welche Aromen, Düfte oder Länge betreffen, nicht mehr beachtet werden. Die Ergebnisse sind sehr kurz und enthalten selten bis nie Aromen und oder Düfte über den Wein. Nach der Analyse der Beispieldaten ist eine mögliche Ursache identifiziert worden: Die durchschnittliche Wortanzahl der Vindor Beschreibungstexten beläuft sich auf 65 Wörter, was nicht einmal die Hälfte der angestrebten Länge von 200 Wörtern entspricht, wie sie in der Prompt angegeben wurde. Infolgedessen stellt sich die Frage, ob das Modell gelernt hat, dass eine Beschreibung von etwa 65 Wörtern ausreichend sei, wenn eine Formulierung mit 200 Wörtern gefordert wird.

Es lässt sich festhalten, dass zum jetzigen Zeitpunkt die Ergebnisse durch eine Prompt besser sind als die durch ein getuntes Modell. Dies muss mit der Qualität der Beispiele zusammenhängen. Daher wird an dieser Stelle entschieden, keine weiteren Ressourcen in das Tuning der Modelle zu investieren. Die Ergebnisse in einem Modell ohne Tuning, die durch die Verwendung von Prompts erzielt werden können, sind bereits zufriedenstellend.

5.3. Beschreibung der automatischen Lösung

Die Grundlage für Shopware 6 ist, wie bereits dargelegt, das PHP-Framework Symfony⁴¹. Symfony ist nach dem MVC-Pattern aufgebaut, erzwingt es aber nicht. Shopware 6 ist ebenfalls nach dem MVC-Pattern aufgebaut, daher folgt auch die Plugin Implementierung dem MVC-Pattern.

Um den modularen Aufbau zu ermöglichen und die Plugin Entwicklung zu unterstützen wird in Shopware 6 ein großer Teil über die Dateistruktur abgewickelt. Hier gibt es eine Reihe an Konventionen, die festgelegt sind. Dies ermöglicht es, dass der Shopware Core einerseits automatisch Konfigurationen erkennen kann und andererseits JavaScript- und Styling-Dateien während der Entwicklung automatisch zu bündeln.

Die minimalen Voraussetzungen für ein Plugin sind eine `composer.json`, ein Einstiegspunkt in Form einer PHP-Klasse und die `config.xml` Datei. In Abbildung 19 ist die daraus resultierende, minimale Dateistruktur dargestellt.

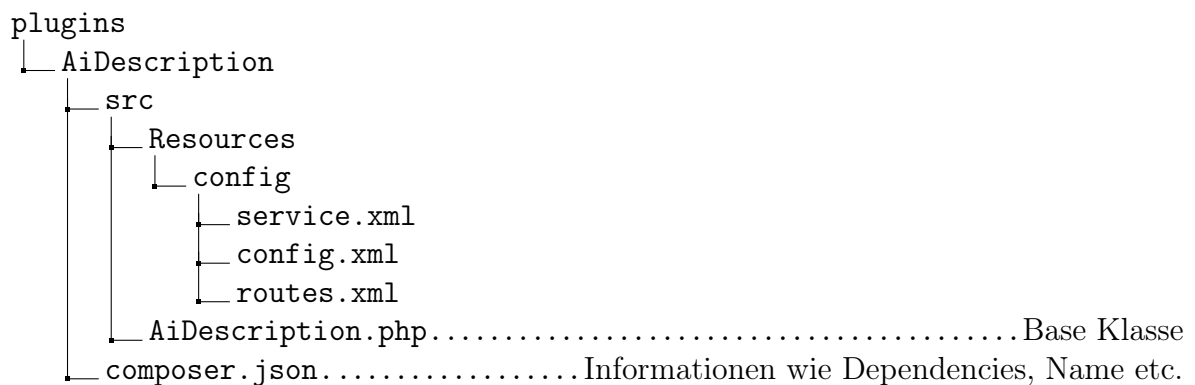


Abbildung 19: Die minimale Dateistruktur für ein Plugin.

Für die Benutzer des Plugins ist der Einstieg über die Shopware-Extensions-Seite erreichbar. Auf dieser Seite werden alle installierten Plugins gelistet, Plugins installiert oder deinstalliert sowie die Einstellungen an den Plugins vorgenommen.

Die möglichen Plugin Einstellungen werden in der `config.xml` Datei definiert und automatisch durch Shopware gerendert. Die xml-Elemente korrespondieren mit zugehörigen Vue-Komponenten aus der Meteor-Komponentenbibliothek. Eine `config.xml` muss mindestens ein Card-Element sowie ein Input-Element enthalten. Es stehen zudem verschiedene Input-Elemente zur Verfügung, die beispielsweise Text, Integer oder Passwörter entgegennehmen [3]. Für das Plugin werden zwei Einstellungsmöglichkeiten definiert: Ein

⁴¹<https://symfony.com/>

Text-Feld für den OpenAI API Schlüssel (Quellcode 2 Zeile 6-9) sowie eine Mehrfach-Auswahl (Quellcode 2 Zeile 10-14). Die Speicher-Logik für die Einstellungen ist bereits durch Shopware implementiert.

Das Mehrfach-Auswahl-Dropdown wird für das Erfüllen von Anforderung A4 benötigt. Hier können Nutzer Attribute definieren, die standardmäßig nicht in die Generierung einfließen sollen. Mit dem xml-Element `<entity>property_group</entity>` wird das zugehörige Objekt definiert. Der Wert `property_group` steht für die Produkt-Eigenschaften. In diesem Fall stehen also alle Eigenschaften in der Mehrfach-Auswahl zur Verfügung, die für die Produkte definiert sind.

Die aus der `config.xml` resultierende Einstellungsseite ist in Abbildung 20 dargestellt.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="https://raw.githubusercontent.com/
4     shopware/platform/master/src/Core/System/SystemConfig/Schema/config.
5     xsd">
6   <card>
7     <title>Ai Description</title>
8     <input-field type="text">
9       <name>apikey</name>
10      <label>Api Key</label>
11    </input-field>
12    <component name="sw-entity-multi-id-select">
13      <name>excludedProperties</name>
14      <entity>property_group</entity>
15      <label>Folgende Eigenschaften sollen standardmäßig ausgeschlossen
16      werden:</label>
17    </component>
18  </card>
19 </config>
```

Quellcode 2: Die aktuelle config.xml

Weitere Konfigurationen werden in `routes.xml` und `service.xml` vorgenommen. Die Datei `routes.xml` dient dazu, individuelle Routen bekannt zu machen, die die Admin-API erweitern. Hier werden über einen Einzeiler mit Wildcard alle Controller in der Dateistruktur bekannt gemacht:

```
<import resource="../../Controller/*Controller.php" type="annotation" />.
```

In der `service.xml` Datei wird die Dependency Injection (DI) konfiguriert. Jeder Service

← Back

AiDescription Save

Ai Description

Sales Channel

All Sales Channels

Api Key

Folgende Eigenschaften sollen standardmäßig ausgeschlossen werden:

Flaschengröße Säure Trinktemperatur Restzucker

Geschmacksrichtung

Flaschengröße ✓

Weingut

Rebsorte

Target group

Alkoholgehalt

Region

Abbildung 20: Die gerenderte Einstellungsseite für das Plugin, basierend auf der `config.xml`. Eigene Darstellung.

wird über die Id bekannt gemacht, diese ist nach dem Muster `Namespace\className` aufgebaut. Die service-Elemente können ein oder mehrere Argumente enthalten, welche die injizierten Klassen darstellen. Für den später betrachteten `GeneraterController` werden die Konfigurationen in Quelltext 9 Zeile 26-33 gesetzt. Es ist zu erkennen, dass drei verschiedene Services innerhalb des `GeneratorController` genutzt werden. Diese werden im folgenden Kapitel im Detail betrachtet.

5.3.1. Implementierung der Generierung

An dieser Stelle wird die Hauptfunktion des Plugins dargestellt: Das Generieren einer Beschreibung anhand der strukturierten Daten. Der `GeneratorController` erweitert die Admin-API und orchestriert bei einer Anfrage die verschiedenen Services, welche implementiert wurden. Bei einer POST-Anfrage an `/api/aidescription/generateDescription` wird der nun beschriebene Ablauf initiiert:

1. Der ComposePrompt Service stellt die Anweisungen für das LLM auf.
2. Der CallApi Service stellt eine Anfrage an die OpenAI API.
3. Der History Service schreibt die Antwort in die Datenbank.
4. Der History Service wird angefragt, ob bereits eine Historie für das Produkt vorhanden ist.
5. Der GeneratorController gibt schließlich die Antwort und optional die Historie zurück.

GeneratorController Die vorgestellten Services laufen im **GeneratorController** zusammen. Dieser erweitert die Admin-API um zwei Routen:

1. `/api/aidescription/generateDescription` initiiert eine neue Generierung.
2. `/api/aidescription/regenerateDescription` initiiert eine Umformulierung basierend auf einer vorherigen Generierung.

Für beide Routen ist der dargelegte Ablauf gleich, nur die genutzten Prompts und dementsprechend auch die Ergebnisse variieren. Routen im Admin-API Scope (`/api/`) werden durch Shopware automatisch hinter einer Autorisierungsebene platziert und erfordern bei jeder Anfrage eine Authentifizierung durch ein Bearer-Token. Da diese Routen jedoch ausschließlich über das Administrations-Frontend angefragt werden, bedarf es keiner zusätzlichen Logik. Alle Benutzer im Administrations-Frontend sind bereits durch ihre Anmeldedaten authentifiziert. Das vorhandene Token wird daher bei jeder Anfrage an den **GeneratorController** angefügt.

ComposePrompt Service Innerhalb dieses Services wird die in Kapitel 5.1.2 aufgestellte Prompt mit den Informationen über die Tonalität und Attributen (Produkteigenschaften) angereichert. Die daraus resultierende Anweisung wird in zwei individuelle Anweisungen aufgeteilt. Die Aufteilung in zwei Teile hängt mit dem Aufbau der API-Anfrage an OpenAI zusammen und wird im CallApi Service genauer dargelegt.

CallApi Service Dieser Service stellt die Anfrage an die OpenAI API. Er erwartet zwei Prompts. Das resultiert aus der Umstellung von den *Completion Modellen* auf die *Chat Modelle*. Die API für die Completen Reihe hat nur eine Prompt im Body erwartet, woraufhin eine Antwort generiert wurde. Die neue Modellreihe (GPT-3.5 und GPT-4) ermöglicht

alternierende Unterhaltungen, das heißt ein Wechsel zwischen System und Benutzer. Es wird sozusagen eine Auflistung einer Unterhaltung übergeben. Wobei die erste Nachricht die (System-)Anweisung darstellt und die zweite Nachricht die Kontextinformationen. Die in Kapitel 5.1.1 aufgestellte Prompt muss dementsprechend geteilt werden. Dies ist ohne Probleme umsetzbar, die Aufteilung der Prompt durch den `ComposePrompt` Service kann in Anhang F.II betrachtet werden. *Hier sind die Informationen über den Wein* trennt die ursprüngliche Prompt in eine Anweisung und eine Kontextergänzung.

Das Request Body an die OpenAI API für die Chat-Funktion muss nach dem folgenden Schema in Quellcode 3 aufgebaut sein:

```
1 {  
2   "model": "gpt-4",  
3   "messages": [  
4     {  
5       "role": "system",  
6       "content": ANWEISUNG  
7     },  
8     {  
9       "role": "user",  
10      "content": KONTEXTERGÄNZUNG  
11    }  
12  ]  
13 }
```

Quellcode 3: Minimales JSON-Body für die Anfrage an die OpenAI API.

Die Autorisierung wird über ein Bearer-Token im Header der Anfrage durchgeführt, dass benötigte API-Token wird hierfür bei jeder Anfrage aus der Datenbank ausgelesen. Eine konkrete Anfrage mit zugehöriger Antwort ist in Anhang F.III.

History Service Das Lesen und Schreiben von Informationen in die Datenbank wird über das Shopware eigene Data Abstraction Layer realisiert. Sowohl für interne als auch individuelle Datenbanktabellen werden automatisch zugehörige Getter und Setter erstellt. Die Funktionalitäten werden in sogenannten Repositories zusammengefasst, eine einzelne Instanz wird als Entität bezeichnet. Sobald die benötigten Entitäten über die DI (siehe Quellcode 9 Zeile 12) bekannt gemacht wurden, stehen die CRUD-Operationen zur Verfügung. Der `HistoryService` stellt zwei Funktionen zur Verfügung: `writeHistory` und

`readHistory`. Beide Funktionen verhalten sich dem Namen entsprechend, `writeHistory` nimmt alle benötigten Parameter entgegen und legt einen neuen Eintrag in der zugehörigen Tabelle an. Die `readHistory` Funktion gibt alle Einträge zu einer bestimmten Produkt-ID zurück.

5.3.2. Erweiterung der Administration

Die Erweiterung des Administrations-Frontend ist ebenfalls eng mit der Dateistruktur verknüpft, damit Shopware automatisch die zugehörigen Bundles erstellen kann. Der Einstiegspunkt ist die `main.js` Datei. Innerhalb der `main.js` werden grundlegende Konfigurationen gesetzt und weitere Dateien importiert⁴². Wie in der Visualisierung bereits dargelegt, gibt es verschiedene Möglichkeiten die Administration zu erweitern. Ziel ist es, die Produkt-Detailseite um einen neuen Tab zu erweitern, dieser beinhaltet dann alle Interaktionsmöglichkeiten. Eine Tab Struktur auf der Produkt-Detailseite ist in Abbildung 11 dargestellt. Jeder Tab ist im Grunde ein eigener View, der sich Navigation, Header sowie die Tab-Leiste teilen. Für das Erweitern der Produkt-Detailseite muss einerseits ein neuer View entstehen. Dieser wird beim Aufrufen der Route angezeigt. Andererseits muss eine bestehende Shopware Komponente erweitert werden, um einen zusätzlichen Tab anzuzeigen.

Individuelle Views werden im `view/view-name` Ordner abgelegt (siehe Abbildung 21). Die `index.js` ist wiederum der Einstiegspunkt des neuen View. Diese beinhaltet die JavaScript Logik für den View und importiert die Templates und das Styling. Das Twig Template für den neuen View ist in `tab-template-html.twig` definiert.

Für eine Erweiterung von bestehenden Shopware Views muss im ersten Schritt die zu erweiternde Komponente identifiziert werden. Dies muss über den Quellcode im Shopware Repository erledigt werden, da nicht alle Komponenten dokumentiert sind. Die für Produkt-Detailseite zuständige Komponente ist `sw-product`⁴³. Innerhalb von `sw-product` wird der Twig-Block `sw_product_detail_content_tabs` für das Darstellen der Tab-Struktur genutzt. Diesen gilt es, um einen neuen Tab zu erweitern. Hierfür wird der Twig-Block für einen Tab eingebunden und durch einen eigenen Tab-Item erweitert. Der Aufruf von `parent` (Quellcode 4, Zeile 4) verhindert, dass der ursprüngliche Inhalt über-

⁴²Da es sich hier um Vue 2 Standardkonventionen handelt, wird darauf verzichtet, diese darzustellen. In Anhang F.IV ist die `main.js` abgelegt.

⁴³Im Shopware GitHub Repository kann der Quellcode eingesehen werden: `sw-product`

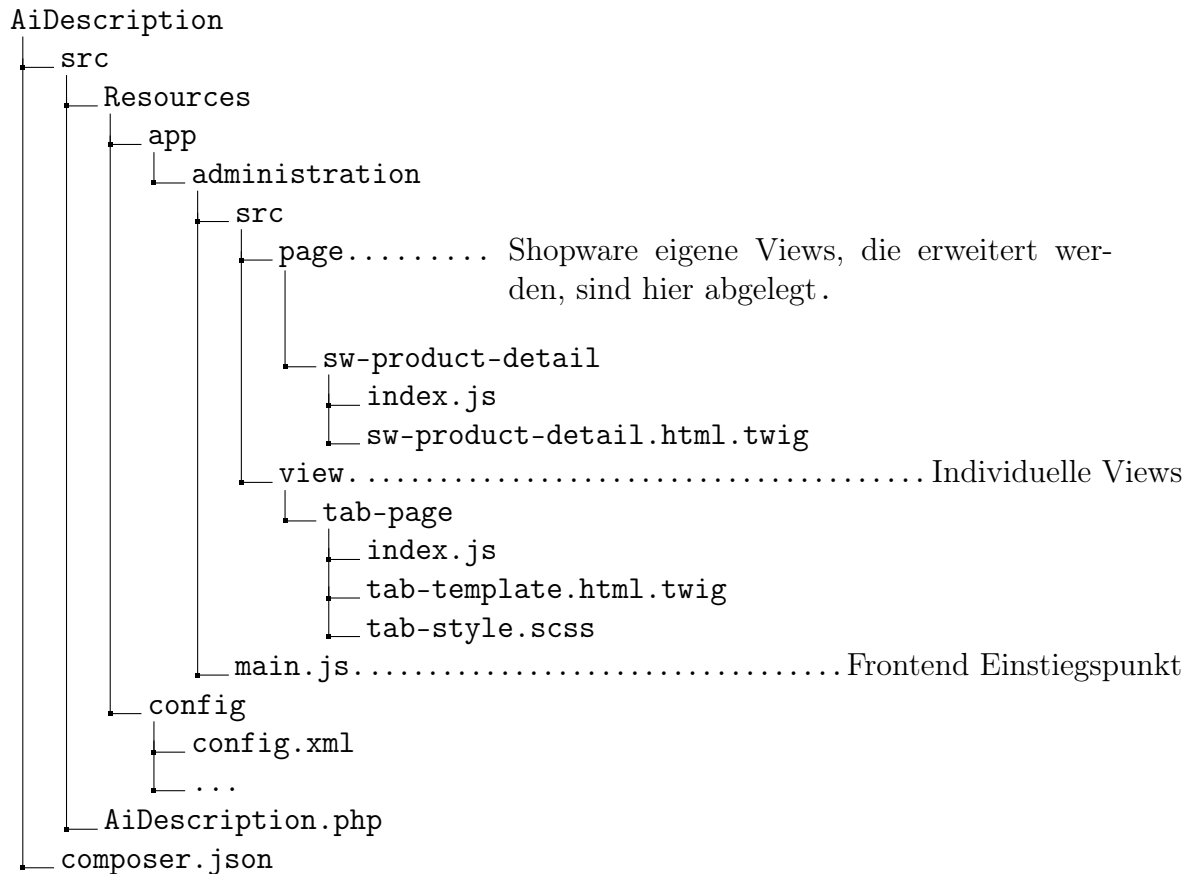


Abbildung 21: Die Dateistruktur für das Administrations-Frontend

schrieben wird. Die Route, welche mit dem neuen Tab verknüpft ist, wird in der `main.js`⁴⁴ angelegt.

```

1 {% block sw_product_detail_content_tabs_general %}
2
3     {% parent %}
4
5     <sw-tabs-item :route="{ name: 'sw.product.detail.ai', params: { id:
6         $route.params.id } }" title="AI Description">
7         AI Description
8     </sw-tabs-item>
9 {% endblock %}

```

Quellcode 4: Die Tab-Erweiterung in sw-product-detail.html.twig

Zusammengefasst sind folgende Veränderungen vorgenommen worden: In der `main.js` wurde eine neue Frontend-Route definiert (`/sw/product/detail/:id/ai`). Beim Aufruf dieser Route wird der neue View (`tab-page`) gerendert. Das Erreichen der Route wird für den

⁴⁴Siehe Anhang F.IV

Benutzer durch eine Erweiterung der Tabs auf der Produkt-Detailseite ermöglicht.

5.3.3. Erweiterung des Shopware WYSIWYG-Editor

Die Grundlage für das Bearbeiten ist der Shopware eigene WYSIWYG-Editor. Dieser ist eine individuelle Implementierung von Shopware, welcher die Web API `execCommand` nutzt, um das HTML-Dokument zu verändern. Mit der `execCommand` Funktion werden die aktuell ausgewählten Texte manipuliert. Beispielsweise innerhalb eines Textfeldes, eines Inputs oder jedem anderen Element mit dem Attribute `contenteditable`. Über den ersten Parameter wird die gewollte Aktion übergeben.

```
1 // Der Befehl für die Kursivschrift...
2 document.execCommand('italic')
3
4 // ...resultiert in folgendem HTML-Markup:
5 Text mit einem <i>wichtigen</i> Wort.
```

Quellcode 5: Die `execCommand` Funktion

Zum Zeitpunkt dieser Arbeit sind die `execCommand` Funktionen veraltet und werden nicht mehr empfohlen. Eine Alternative für moderne Browser existiert nicht [18]. Zudem kündigt ein bestehender Architecture decision record (ADR) von Shopware eine neue Implementierung des Editors an [2]. Daher wird eine vom `execCommand` unabhängige Implementierung angestrebt. Das erleichtert auch eine Portierung auf zukünftige Shopware Versionen, wenn der WYSIWYG-Editor ausgetauscht wird.

Das Markieren einer Textpassage für eine Umformulierung durch das LLM, wurde in Kapitel 4.4 bereits ausführlich visualisiert. Aus dieser Visualisierung ergibt sich eine Anforderung: Eine ausgewählte Textpassage muss im HTML-Markup umschlossen werden, um sie einerseits visuell vom Rest abzugrenzen und andererseits extrahieren zu können. Bei

Dieser zauberhafte Weißwein ist ein Meisterwerk. Der Jahrgang 2021 schimmert in einem hellen Gelb mit grünlichen Reflexen, ein Spiegelbild seiner Frische und Lebendigkeit.

Abbildung 22: Die Auswahl einer Textpassage (in blau Hinterlegt) und eine Textpassage die innerhalb eines `span`-Elements liegt (Hervorhebung). Eigene Darstellung.

einer näheren Untersuchung wird offensichtlich, dass diese auf den ersten Blick einfache

Interaktion eine Vielzahl von zusätzlichen Implikationen mit sich bringt, die im folgenden Abschnitt erörtert werden. Zur besseren Verständlichkeit der nachfolgenden Ausführungen müssen zunächst zwei Begriffe eingeführt werden. Es wird von **ausgewähltem Text** und von **markiertem Text (einer Hervorhebung)** gesprochen. Der ausgewählte Text entspricht dem, was ein Nutzer mit einem Eingabegerät ausgewählt hat, in dem er zum Beispiel mit gedrückter Maustaste über einen Text fährt (in Abbildung 22 die blau hinterlegte Passage). Der markierte Text hingegen repräsentiert das, was bereits durch den Nutzer für eine Umformulierung hervorgehoben wurde (in Abbildung 22 umschlossen mit einem span-Element). Für das bessere Verständnis wird in der folgenden Auflistung von **Hervorhebung** gesprochen. Hiermit ist die Textpassage gemeint, die durch einen Nutzer für eine Umformulierung ausgewählt wurde. Ebenfalls wird hinter jedem Fall ein Beispiel gegeben, wie der Inhalt des aktuell ausgewählten Textes sein kann.

Die einzelnen Möglichkeiten die hierbei auftreten können sind wie folgt:

Fall 1: Der aktuell ausgewählte Text beinhaltet nur Text:

```
lorem ipsum
```

Fall 2: Der aktuell ausgewählte Text beinhaltet eine vollständige Hervorhebung:

```
<span>lorem ipsum</span>
```

Fall 3: Der aktuell ausgewählte Text beinhaltet sowohl Text, als auch einen unvollständigen Teil einer Hervorhebung:

```
lorem <span>ipsum non
```

Fall 4: Der aktuell ausgewählte Text beinhaltet eine vollständige Hervorhebung mit zusätzlichem Text, welcher eine oder beide Seiten erweitern:

```
lorem <span> ipsum ipsum</span> non
```

Fall 5: Der aktuell ausgewählte Text mehr als ein span-Element:

```
<span>ipsum</span> non<span>lorem</span>
```

Fall 6: Der aktuell ausgewählte Text beinhaltet andere DOM-Elemente:

```
lorem <b>ipsum</b>
```

Fall 7: Der aktuell ausgewählte Text beinhaltet nur einen unvollständigen Teil anderer DOM-Elemente.

```
lorem <b>ipsum
```

Für jeden Fall ist die Nutzerintention eine andere. Fall 1 ist eindeutig: Wenn lediglich Text ausgewählt wird, beabsichtigt der Nutzer, diesen Text zu manipulieren. Der aktuell ausgewählte Text wird daher mit einem span-Element umschlossen. Hierdurch kann er

visuell über eine CSS-Klasse hervorgehoben werden und ebenfalls extrahiert werden, da er eindeutig angesprochen werden kann. Fall 2 ist ebenfalls eindeutig: Der Nutzer möchte einen vorher markierten Text nicht mehr hervorheben. Die betroffenen span-Elemente werden entfernt, der Inhalt der span-Elemente wird an derselben Stelle eingefügt.

Sobald der aktuell ausgewählte Text aber Teile von span-Elementen oder vollständige DOM-Elemente enthält, sind die Intensionen der Nutzer nicht mehr eindeutig. Eine Teilauswahl kann bedeuten, dass der Nutzer eine Hervorhebung erweitern oder sie reduzieren möchte. Wenn die Länge des aktuell ausgewählten Textes beispielsweise kürzer ist, als die Länge der Markierung, könnte dies bedeuten, dass eine Erweiterung erwartet wird. Hierbei entstehen eine Reihe an Grenzfällen, die den Implementierungsaufwand erheblich vergrößern. Aus praktischen Gründen und unter Berücksichtigung des Anwendungsbereichs wird eine Zwischenlösung gewählt, die nicht jeden Grenzfall individuell behandelt: Wenn die Auswahl ausschließlich Text enthält, wird sie umschlossen. In allen anderen Fällen führt die Interaktion dazu, dass die hervorgehobene Textpassage nicht mehr hervorgehoben ist. Das umschließende span-Element wird entsprechend entfernt. Die hieraus resultierenden Einschränkungen für einen Nutzer sind im Allgemeinen geringfügig. Wenn tatsächlich eine Erweiterung gewünscht ist und die Interaktion die Hervorhebung entfernt, muss lediglich ein zweiter Klick durchgeführt werden, um die Auswahl wieder zu umschließen und damit hervorzuheben.

Der einzige Sonderfall, der berücksichtigt werden muss, ist Fall 7. Würde man ein Teil eines anderen DOM-Elements mit einem span-Element umschließt, führt das zu einem fehlerhaften HTML-Markup⁴⁵, was unbedingt vermieden werden muss.

Für die Umsetzung der Anforderungen muss der aktuell ausgewählte Text extrahiert werden, sowie die Inhalte einzelner DOM-Elemente. Hierfür bieten sich die beiden Web-APIs **Selection** und **Range** an:

Das **Range** Interface repräsentiert ein Fragment eines Dokumentes. Es besteht aus einer bis mehreren Nodes. Eine leere **Range** wird über `document.createRange()` erstellt und kann daraufhin mit einer Start- und End-Node initialisiert werden:

⁴⁵Hierbei könnte folgendes fehlerhaftes Markup entstehen: `Ein Satz mit <i> italic Text der eigentlich bis hier geht </i>`. Das i-Element wird innerhalb des span-Elements geöffnet, aber erst nach dem span-Element geschlossen.

```
1 let range = document.createRange();  
2  
3 range.setStart(startNode, startOffset);  
4 range.setEnd(endNode, endOffset);
```

Quellcode 6: Die initialisierung eines Range Interfaces. Entnommen aus [17].

Innerhalb des **Range Interface** existieren verschiedene Properties und Funktionen, die sich alle auf die Inhalte der **Range** beziehen. Es wird ermöglicht diese auszulesen, zu verändern und zu manipulieren. Für das Verständnis des folgenden Quellcodes sind zwei Funktionen aus dem **Range Interface** relevant: Die Funktion `range.intersectsNode(someNode)` prüft, ob die übergebene Node innerhalb der **Range** vollständig oder unvollständig vorhanden ist [1]. Hiermit werden die Fälle abgedeckt, bei denen es darum geht, das der aktuell ausgewählte Text andere DOM-Elemente enthält. In Quellcode 7 Zeile 17 - 22 ist die zugehörige Implementierung: Es werden für alle span-Elemente im WYSIWYG-Editor geprüft, ob diese innerhalb der aktuellen Textauswahl vorhanden sind.

Die Funktion `range.extractContents()` entfernt alle Nodes in der **Range** und gibt diese als Fragment (**DocumentFragment**) zurück. Enthält die **Range** nur Teile einer Node (siehe Fall 7) werden automatisch die relevanten **tags** geschlossen, um sowohl im Fragment als auch im ursprünglichen Dokument eine valide HTML-Struktur zu erhalten [1]. Das bringt für die Implementierung den Vorteil, dass nicht geprüft werden muss, ob in der **Range** andere DOM-Elemente als spans vorhanden sind.

Das **Selection** Objekt repräsentiert eine Auswahl, die durch einen Nutzer getroffen wurde. Zum Beispiel, indem die linke Maustaste gehalten wird und man über einen Text fährt. Mit dem Aufruf von `window.getSelection()` kann das aktuelle **Selection** Objekt geholt werden (siehe Zeile 2 in Quellcode 7). Das Objekt beinhaltet eine Reihe an Funktionen, welche für das Manipulieren und untersuchen einer Auswahl genutzt werden können. Für die Implementierung sind diese nicht relevant, daher werden sie an dieser Stelle nicht vorgestellt. Das **Selection** Objekt wird direkt in ein **Range** Interface konvertiert, um Zugriff auf die vorher dargelegten Funktionen zu haben.

```
1 function toggleSpan() {
2     const selection = window.getSelection();
3     const editorElement = document.querySelector("#aidescription-
4 generated-editor");
5     if (!editorElement && selection.rangeCount === 0) return;
6
7     const userSelectedRange = selection.getRangeAt(0);
8     const spanElements = editorElement.querySelectorAll("span");
9
10    let includesAnotherSpan = false;
11    let selectedText = "";
12
13    if (userSelectedRange.toString().trim() !== "") {
14        selectedText = userSelectedRange.toString();
15    }
16
17    // check if a span or part of a span are inside of the selection
18    for (const spanElement of spanElements) {
19        if (userSelectedRange.intersectsNode(spanElement)) {
20            includesAnotherSpan = true;
21            break;
22        }
23    }
24
25    if (!includesAnotherSpan && selectedText !== "") {
26        // case 1: the selection only contains text and is not a partial
27        // of another span.
28        const newSpan = document.createElement("span");
29        newSpan.setAttribute("data-change", "true");
30        // edge case: the selection contains another DOM-Element
31        newSpan.appendChild(userSelectedRange.extractContents());
32        userSelectedRange.insertNode(newSpan);
33    } else {
34        // turns out we can combine the other cases: just remove the
35        // span and preserve its content
36        for (const spanElement of spanElements) {
37            const spanRange = document.createRange();
38            spanRange.selectNodeContents(spanElement);
39
40            if (userSelectedRange.intersectsNode(spanElement)) {
```

```
38     const fragment = spanRange.extractContents();
39     const parent = spanElement.parentNode;
40     parent.insertBefore(fragment, spanElement);
41     parent.removeChild(spanElement);
42 }
43 }
44 }
45 }
46
```

Quellcode 7: Die JavaScript Implementierung für das Umschließen und Entfernung der Umschließung einer Auswahl mit einem span Element

Im Quellcode 7 sind die beiden Unterscheidungen implementiert: Enthält die **Selection** nur Text, wird ein neues span-Element erstellt und mit dem Inhalt der **Selection** gefüllt. Daraufhin wird dieses an derselben Stelle platziert (Siehe Zeile 24-30 in Quellcode 7).

Für alle anderen Fälle wurde festgelegt, dass alle span-Elemente innerhalb der **Selection** entfernt werden. Es wird über alle span-Elemente iteriert und geprüft, ob diese innerhalb der **Selection** liegen. Ist dies der Fall, wird der Inhalt des span-Elements extrahiert und ohne das span-Element an derselben Stelle platziert (siehe Zeile 32-43 in Quellcode 7).

Während der Interaktion mit der Implementierung ist ein weiterer Sonderfall aufgetreten: Es wird nicht geprüft, ob die Hervorhebung einen vollständigen Satz enthält. Derzeit ist es möglich, nur einzelne Wörter für eine Umformulierung auszuwählen. Abhängig von der Anzahl der hervorgehobenen Wörter kann dies zu unzureichenden Ergebnissen führen, da der Kontext für eine Umformulierung fehlt. Eine mögliche Lösung wäre die Überprüfung, ob die Hervorhebung einen vollständigen Satz enthält. Dafür könnte beispielsweise geprüft werden, ob der Satzanfang vor einem Leerzeichen und Punkt liegt und das Ende mit einem Punkt abschließt. Jedoch könnte dies die Nutzer verwirren, da die Hervorhebung dann scheinbar ohne ersichtlichen Grund größer oder kleiner wird. Daher wurde entschieden, diesen Sonderfall nicht zu implementieren, sondern ihn stattdessen durch einen Hinweis oder eine Erklärung abzufangen.

Die tiefgreifende Integration in den WYSIWYG-Editor ist eine wichtige Erweiterung, um eine effektive Änderung und dadurch Evaluierung der Beschreibungstexte durch die Redakteure zu ermöglichen. Mit den Visualisierungen und der daraus folgenden Implementierung sind die verschiedenen Möglichkeiten der Gestaltung eines Userinterface abgedeckt worden, um eine fundierte Entscheidung treffen zu können, wie dieses gestaltet werden muss.

6. Bewertung durch Experten

In diesem Kapitel wird eine Evaluation der generierten Beschreibungstexte durchgeführt. Ziel ist es zu ermitteln, ob sie einer Prüfung durch einen Experten standhalten. Dabei sind zwei Dimensionen besonders interessant: Erstens, ob die Beschreibungstexte entsprechend der Zielgruppe formuliert sind und zweitens, ob die Inhalte korrekt dargestellt werden. Letzteres bezieht sich insbesondere auf Aromen und Düfte, da diese Informationen nicht in den strukturierten Daten vorliegen und eine Transferleistung darstellen. Der Vergleich in Kapitel 5.1 hat bereits gezeigt, dass die Informationen aus den strukturierten Daten in der Regel korrekt in die Beschreibungen integriert werden.

Für die Auswahl eines Experten sind verschiedene Kriterien relevant. Die Produkte müssen im Vindor-Katalog vertreten sein, um auf die strukturierten Daten zugreifen zu können. Ist dies nicht der Fall, müssten diese Daten zusammengestellt und in ein lesbares Format gebracht werden. Zusätzlich gibt es bereits Beschreibungstexte für Produkte im Vindor-Katalog, die für eine erste Validierung herangezogen werden können. Dies soll sicherstellen, dass die generierten Beschreibungen für die Evaluierung, zumindest in Grundzügen korrekt sind. Darüber hinaus sollte es sich um ein deutsches Weingut handeln, denn der Anwendungsbereich gibt vor, dass deutsche Texte generiert werden sollen. Es bietet sich an, einen Winzer für die Bewertung heranzuziehen, da dieser seine Produkte kennt und wenig Transferleistung für eine Bewertung erbringen muss. Nicht zu vernachlässigen ist die zeitliche Komponente: Da das Plugin erst im Laufe dieser Arbeit entstanden ist, ist das Zeitfenster, in welchem Texte vorhanden sind, sehr begrenzt.

Der Partner dieser Arbeit hat Kontakt zu einem deutschen Winzer hergestellt, der sich bereit erklärt hat, die Beschreibungstexte zu bewerten. Es wurde entschieden, dem Winzer eine Auswahl von Texten zur Verfügung zu stellen, ohne Informationen über die Herkunft und deren Ersteller. Damit wurde sich erhofft, dass eine unvoreingenommene Rückmeldung entsteht. Der Winzer wurde im ersten Schritt nur angefragt, ob die Beschreibungen in dieser Form für den Online-Shop genutzt werden können. Die erste Rückmeldung war äußerst positiv.

Anschließend wurde ein Fragenkatalog erstellt und dem Winzer zugesandt. Die Fragen wurden in einer persönlichen Korrespondenz beantwortet. Das vollständige Gedächtnisprotokoll ist in Anhang G.III, im Folgenden werden die wichtigsten Erkenntnisse zusammengefasst:

Der Winzer hat nicht erkannt, dass die Texte automatisch generiert wurden. Dies führt

er auf zwei Punkte zurück: Erstens gibt es immer Experten, die einen Wein nicht perfekt beschreiben und zweitens sind die Aromen und Düfte in den generierten Texten sehr allgemein gehalten und daher zutreffend. Dennoch lässt sich daraus ableiten, dass die Inhalte der Beschreibungstexte korrekt generiert werden.

Des Weiteren wurden die Lesbarkeit und die Formulierungen überprüft. Im Allgemeinen war die Rückmeldung darauf ebenfalls positiv; der Sprachfluss und die Lesbarkeit waren bei allen Beschreibungen gegeben. Dennoch gibt es Formulierungen, welche der Winzer in der Form nicht verwenden würde. Dies bezog sich auf die Beschreibung des Weins *Stairs n' Roses, Stairs n' Roses Riesling feinherb, QbA Mosel*⁴⁶, in dem die Formulierung *entführt Sie auf eine sinnliche Reise durch die romantischen Weinberge Deutschlands* vorkam. Solche Formulierungen implizierten für ihn immer einen Verschnitt von Rebsorten, was kein Qualitätsmerkmal für einen Wein ist. Daher würde er auf solche Formulierungen verzichten. Dies gilt auch für Formulierungen, die sich auf viele verschiedene Regionen beziehen. Das Verhindern von solchen Formulierungen in den generierten Beschreibungen lässt sich nicht ohne Weiteres umsetzen. Ein Ansatz hierfür könnten erneute Kontexterweiterungen in der Prompt sein. Beispielsweise eine Anweisung wie, *schreibe nur über eine Region, auch wenn der Wein aus mehreren Regionen kommt*. Es stellt sich jedoch die Frage, wie viele weitere Formulierungen verhindert werden sollen, wenn die Anzahl der Experten erhöht wird. Das Problem, das entstehen kann, ist die maximal mögliche Länge einer Prompt. Die aktuelle Implementierung unterstützt noch keine Möglichkeit, einzelne Phrasen auszuschließen. Hierfür muss auf den Redakteur zurückgegriffen werden, der die Beschreibungstexte erstellt und prüft.

Abgesehen von den Formulierungen bemerkte der Winzer, dass ihm bei einigen Weinen interessante Informationen fehlen, die für ihn ein Kaufargument darstellen würden. Dies bezieht sich beispielsweise auf die Weine der *Next Generation*⁴⁷. Dabei handelt es sich um eine Rebsorte, die einen höheren Pilzwiderstand aufweist und daher weniger gespritzt werden muss. Sie sind nachhaltiger und ressourcenschonender. Während der persönlichen Korrespondenz wurde die Ursache dafür identifiziert: Diese Informationen sind nicht in den strukturierten Daten vorhanden und werden nur im Rückenetikett des Weins beschrieben. Die Rückenetiketten werden zum jetzigen Zeitpunkt nicht in die Generierung mit einbezogen. Da es wahrscheinlich ist, dass auch bei anderen Weinen interessante Informationen

⁴⁶Siehe Anhang G.IV.2

⁴⁷Siehe Anhang G.IV.6

nicht in den strukturierten Daten vorhanden sind, ist hieraus eine neue Anforderung entstanden: (A20) Das System wird den Nutzenden die Möglichkeit bieten, Informationen, die nicht in den strukturierten Daten vorhanden sind, zu ergänzen.

Darüber hinaus fand eine Bewertung in Zusammenarbeit mit dem Partner dieser Arbeit statt, insbesondere mit dem Redakteur, der die Texte einpflegt und den Einkauf durchführt. Die Rückmeldungen waren ähnlich wie die des Winzers. Die Beschreibungen sind zutreffend und lesen sich gut. Vor allem mit Blick auf die Zielgruppe, welche sowohl der Winzer als auch der Redakteur ähnlich beschreiben: Es handelt sich nicht nur um Gelegenheitstrinker, sondern auch um anspruchsvolle Kunden, die bereit sind, Wein im höheren Preissegment zu kaufen und diesen als ein Genussmittel betrachten.

Es ist anzumerken, dass die Evaluation der Ergebnisse eine komplexe Aufgabe darstellt. Insbesondere gestaltet sich die objektive Bewertung als problematisch. Einige Aspekte, wie der Lesefluss und die Formatierung, können zwar verallgemeinert werden, bleiben jedoch letztendlich subjektiv. Im Rahmen dieser Arbeit sind die festgelegten Kriterien eher als äußerliche Vorgaben zu betrachten. Hierunter fallen die Textlänge sowie das Ausschließen bestimmter Attribute in den Beschreibungstexten. Darüber hinaus war es schwierig, eine universelle Zielgruppe zu definieren. Zum aktuellen Zeitpunkt wird angenommen, dass die Zielgruppe mit dem Produkt in Zusammenhang steht, speziell im Online-Shops von Vindor, wo unterschiedliche Zielgruppen anzutreffen sind. Ein Ansatz, diesem Problem entgegenzuwirken, besteht darin, die Möglichkeit zur Anpassung der Formulierung (Tonalität) zu bieten. Dies ermöglicht es den Redakteuren, bei jeder Beschreibung eine Auswahl zu treffen, wie die Tonalität und damit der Beschreibungstext formuliert ist.

7. Fazit

Diese Arbeit hat sich mit der automatisierten Generierung von Beschreibungstexten für Wein im E-Commerce befasst. Dafür wurde sich zu Beginn mit den Strukturen im Einzelhandel auseinandergesetzt, um sich der Bedeutung von Beschreibungstexten bewusst zu machen. Die Bedeutung von Beschreibungstexten im E-Commerce ist vielschichtig und reicht von der Bereitstellung von Informationen über die Beeinflussung der Kaufentscheidung bis hin zur Reduzierung von Retouren. Beschreibungstexte dienen nicht nur als Schnittstelle zwischen dem Produkt und dem Kunden, sondern haben auch Auswirkungen auf die Darstellung von Produkten, die visuell schwer erfassbar sind.

Um eine effektive Unterstützung der Redakteure zu ermöglichen, wurde der aktuelle Ablauf im Unternehmenskontext intensiv betrachtet. Hierbei hat sich vor allem herauskristallisiert, dass Beschreibungstexte bis dato in einem zeitaufwändigen Prozess manuell erstellt wurden. Dieser Vorgang ist wichtig, um sich von anderen Online-Händlern abgrenzen zu können.

Für die fundierte Auswahl eines LLM sind die Anwendungsgebiete und Aufbauten der aktuellen Modelle dargelegt worden. Es hat sich herausgestellt, dass die Transformer Architektur allen modernen LLMs zugrunde liegt. Auf Grund der hohen Anzahl an Anbietern und Modellen ist eine Eingrenzung während der Analyse vorgenommen worden. Hierbei wurde sich vor allem auf Anbieter konzentriert, die Einstellungsmöglichkeiten an ihren Modellen ermöglichen. Das ermöglichte eine Anpassung und Justierung an den Anwendungsfall. Im Zuge des tiefgreifenden Prompt Engineerings konnte festgestellt werden, dass das aktuellste Modell von OpenAI (GPT-4) sich am besten für die automatische Generierung von Beschreibungstexten eignen. Es konnte sich einerseits von Konkurrenzprodukten abgrenzen, da es umfassende Konfigurationen ermöglicht und andererseits dadurch, dass es mit der geringsten Dichte an Informationen die besten und konsistentesten Ergebnisse liefert. Des Weiteren wurde festgestellt, dass das GPT-4 Modell bereits passende Ergebnisse liefert, sodass sich der zusätzliche Aufwand für ein Tuning nicht als sinnvoll erwiesen hat. Hier muss jedoch beachtet werden, dass für das Tuning nur bereits vorhandene Daten genutzt wurden. Es kann davon ausgegangen werden, dass eine Arbeit, die darauf abzielt Tuningdaten zu erstellen, eine Verbesserung der Ergebnisse erzielt.

Die Grundlagen für die Implementierung wurden durch eine umfassende Konzeption der automatischen Lösung ermöglicht. Die Anforderungsermittlung ist der erste Schritt für die Entscheidung, wie ein Userinterface aufgebaut ist, das eine effektive Generierung und Eva-

luierung ermöglicht. Die darauf folgende Visualisierung hat es ermöglicht, die Forschungsfrage zu beantworten, wie ein Userinterface gestaltet werden muss, dass eine effektive Generierung und Evaluierung ermöglicht. Hierbei haben sich zwei Punkte manifestiert: Es muss ein Ablauf ermöglicht werden, der einerseits alle Interaktionen an einer Stelle bündelt und andererseits bestehende Komponenten so erweitert, dass Funktionalitäten ergänzt und nicht eliminiert werden. Dies wurde auch durch den Partner dieser Arbeit validiert.

Mithilfe der Evaluation durch einen Winzer konnte zwar bestätigt werden, dass die Beschreibungstexte an sich genutzt werden können, dennoch war es nicht möglich, allgemeingültige Kriterien für den Anwendungsfall aufzustellen, die eine objektive Bewertung ermöglichen. Es muss immer individuell entschieden werden, ob und in welchem Umfang ein generierter Beschreibungstext für einen Wein und deren Zielgruppe geeignet ist. Dementsprechend ist eine Lösung entstanden, die nicht als vollautomatisch angesehen werden kann, sondern als Unterstützung für die Redakteure.

Das Ergebnis ist ein Plugin für Shopware 6, das es Redakteuren ermöglicht, anhand der strukturierten Daten Beschreibungstexte für Wein automatisch erstellen zu lassen. Es ist eine tiefe Integration im Shopware-Ecosystem entstanden, dennoch wurden für die wichtigsten Interaktionen Lösungen gewählt, welche sich mit wenig Aufwand abstrahieren und entkoppeln lassen. Dabei wurde darauf geachtet, auf Standardkonventionen aufzubauen, um die Nutzung für Redakteure zu erleichtern. Es wird davon ausgegangen, dass die Einarbeitungszeit gering ist, da den Redakteuren bekannte Funktionen auch im Plugin funktionieren.

Aktuell ist das Plugin noch fest an den Anwendungsfall des Weinhandels gekoppelt. Es bietet sich an, in einer aufbauenden Arbeit zu untersuchen, wie sich Aufbauten und vor allem das Prompting verändern muss, um andere Anwendungsfälle abzudecken. Hier gilt es zu klären, ob es Prompts gibt, die in mehreren Anwendungsfällen funktionieren oder ob es nur möglich ist, weitere Anwendungsfälle durch individuell zugeschnittene Anweisungen zu erschließen.

Literaturverzeichnis

- [1] Web Hypertext Application Technology Working Group (WHATWG). *DOM Standard*. URL: <https://dom.spec.whatwg.org/#interface-range> (besucht am 11.09.2023).
- [2] Shopware AG. *Admin text editor evaluation*. en. März 2023. URL: <https://github.com/shopware/platform/blob/trunk/adr/2023-03-27-admin-text-editor-evaluation.md> (besucht am 11.09.2023).
- [3] Shopware AG. *Home*. en. URL: <https://developer.shopware.com/docs/> (besucht am 04.09.2023).
- [4] Shopware AG. *Pläne & Preise*. de-DE. URL: <https://www.shopware.com/de/preise/> (besucht am 04.09.2023).
- [5] Shopware AG. *Shopware 6 Admin Design System*. URL: <https://shopware.design/> (besucht am 08.09.2023).
- [6] Jay Alammam. *A Visual and Interactive Guide to the Basics of Neural Networks*. Dez. 2016. URL: <https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/> (besucht am 12.06.2023).
- [7] *Aleph Alpha GmbH, Heidelberg*. de. URL: <https://www.northdata.de/Aleph+Alpha+GmbH,+Heidelberg/Amtsgericht+Mannheim+HRB+732804> (besucht am 02.09.2023).
- [8] Mariette Awad und Rahul Khanna. “Machine Learning”. en. In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Hrsg. von Mariette Awad und Rahul Khanna. Berkeley, CA: Apress, 2015. ISBN: 978-1-4302-5990-9. DOI: 10.1007/978-1-4302-5990-9_1. URL: https://doi.org/10.1007/978-1-4302-5990-9_1 (besucht am 28.06.2023).
- [9] Greg Brockman und Ilya Sutskever. *OpenAI LP*. en-US. URL: <https://openai.com/blog/openai-lp> (besucht am 24.08.2023).
- [10] Tom B. Brown u. a. *Language Models are Few-Shot Learners*. arXiv:2005.14165 [cs]. Juli 2020. DOI: 10.48550/arXiv.2005.14165. URL: <http://arxiv.org/abs/2005.14165> (besucht am 24.08.2023).

- [11] Katharina Buchholz. *Infographic: Threads Shoots Past One Million User Mark at Lightning Speed*. en. Juli 2023. URL: <https://www.statista.com/chart/29174/time-to-one-million-users> (besucht am 18.09.2023).
- [12] Andriy Burkov. *Machine learning kompakt: alles, was Sie wissen müssen*. ger. Übers. von Knut Lorenzen. 1. Auflage. Frechen: mitp, 2019. ISBN: 978-3-95845-997-7 978-3-95845-995-3.
- [13] Aakanksha Chowdhery u. a. *PaLM: Scaling Language Modeling with Pathways*. en. arXiv:2204.02311 [cs]. Okt. 2022. URL: <http://arxiv.org/abs/2204.02311> (besucht am 24.09.2023).
- [14] Datanyze. *Market share of leading e-commerce software platforms and technologies worldwide as of July 2023*. en. März 2023. URL: <https://www.statista.com/statistics/710207/worldwide-ecommerce-platforms-market-share/> (besucht am 01.10.2023).
- [15] Frank Deges. “Kaufprozess und Kaufverhalten im E-Commerce”. de. In: *Grundlagen des E-Commerce: Strategien, Modelle, Instrumente*. Hrsg. von Frank Deges. Wiesbaden: Springer Fachmedien, 2020, S. 71–84. ISBN: 978-3-658-26320-1. DOI: 10.1007/978-3-658-26320-1_4. URL: https://doi.org/10.1007/978-3-658-26320-1_4 (besucht am 08.04.2023).
- [16] Thomas Foscht, Bernhard Swoboda und Hanna Schramm-Klein. *Käuferverhalten: Grundlagen - Perspektiven - Anwendungen*. de. Wiesbaden: Springer Fachmedien, 2015. ISBN: 978-3-658-08548-3 978-3-658-08549-0. DOI: 10.1007/978-3-658-08549-0. URL: <https://link.springer.com/10.1007/978-3-658-08549-0> (besucht am 19.06.2023).
- [17] Mozilla Foundation. *Document: createRange() method - Web APIs | MDN*. en-US. Apr. 2023. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Document/createRange> (besucht am 11.09.2023).
- [18] Mozilla Foundation. *Document: execCommand() method - Web APIs | MDN*. en-US. Juli 2023. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Document/execCommand> (besucht am 11.09.2023).
- [19] *Generative KI - Weltweit | Statista Marktprognose*. de. URL: <https://de.statista.com/outlook/tmo/kuenstliche-intelligenz/generative-ki/weltweit> (besucht am 18.09.2023).

- [20] Westwein WHG Weinhandels und Genussmittel GmbH. *Vindor.de*. de. URL: <https://www.vindor.de/> (besucht am 18.09.2023).
- [21] Aleph Alpha GmbH. *Model Card Luminous / Aleph Alpha API*. en. URL: <https://docs.aleph-alpha.com/docs/introduction/model-card/> (besucht am 30.06.2023).
- [22] Aleph Alpha GmbH. *What is Luminous? / Aleph Alpha API*. en. URL: <https://docs.aleph-alpha.com/docs/introduction/luminous/> (besucht am 30.09.2023).
- [23] gjuce GmbH. *gjuce GmbH*. URL: <https://www.gjuce.com/> (besucht am 18.09.2023).
- [24] Weinkontor Freund GmbH. *BDX REVOLUTION Sauvignon Blanc Bordeaux AOC » Producta Vignoble » Wein aus Frankreich Bordeaux St. Vincent de Pertignas*. URL: <https://www.weinkontor-freund.de/wein/frankreich/producta.vignobles/4034130> (besucht am 26.09.2023).
- [25] Brockman Greg und Ilya Sutskever. *Introducing OpenAI*. en-US. URL: <https://openai.com/blog/introducing-openai> (besucht am 24.08.2023).
- [26] HDE. *Wein & Sekt: Online-Umsatzanteile im Handel bis 2022*. de. Mai 2023. URL: <https://de.statista.com/statistik/daten/studie/1387418/umfrage/online-umsatzanteile-im-handel-mit-wein-und-sekt-in-deutschland/> (besucht am 19.06.2023).
- [27] Gerrit Heinemann. *Der neue Online-Handel: Geschäftsmodell und Kanalexzellenz im Digital Commerce*. de. Wiesbaden: Springer Fachmedien, 2015. ISBN: 978-3-658-06785-4 978-3-658-06786-1. DOI: 10.1007/978-3-658-06786-1. URL: <https://link.springer.com/10.1007/978-3-658-06786-1> (besucht am 20.06.2023).
- [28] Ziwei Ji u. a. “Survey of Hallucination in Natural Language Generation”. In: *ACM Computing Surveys* 55.12 (Dez. 2023). arXiv:2202.03629 [cs], S. 1–38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3571730. URL: <http://arxiv.org/abs/2202.03629> (besucht am 31.08.2023).
- [29] Takeshi Kojima u. a. *Large Language Models are Zero-Shot Reasoners*. arXiv:2205.11916 [cs]. Jan. 2023. DOI: 10.48550/arXiv.2205.11916. URL: <http://arxiv.org/abs/2205.11916> (besucht am 31.08.2023).

- [30] Alexandra Sasha Luccioni, Sylvain Viguier und Anne-Laure Ligozat. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*. arXiv:2211.02001 [cs]. Nov. 2022. URL: <http://arxiv.org/abs/2211.02001> (besucht am 30.06.2023).
- [31] *Marktanteil des Online-Handels am Umsatz mit Lebensmitteln in Deutschland bis 2022*. de. URL: <https://de.statista.com/statistik/daten/studie/475255/umfrage/marktanteil-des-online-handels-am-umsatz-mit-lebensmitteln-in-deutschland/> (besucht am 19.06.2023).
- [32] Meta. *Llama 2 - Meta AI*. URL: <https://ai.meta.com/llama/> (besucht am 02.10.2023).
- [33] Noir Mohd, Gushan Dahsmana und Deepak Upadhyay. "Implementation of Traditional Vs. Transformer MachineLearning Models". In: *Webology* (2021). DOI: 10.29121/WEB/V18I4/144. URL: <https://www.webology.org/data-cms/articles/20220922073631pmPaper144.pdf> (besucht am 08.05.2023).
- [34] *Neural machine translation with a Transformer and Keras*. en. URL: <https://www.tensorflow.org/text/tutorials/transformer> (besucht am 22.08.2023).
- [35] Anna Nordhoff. *AI-based ecommerce platform: new AI-based tools*. en-DE. Feb. 2023. URL: <https://www.shopware.com/en/news/shopware-new-ai-feature/> (besucht am 18.09.2023).
- [36] OpenAI. *GPT-4 Technical Report*. arXiv:2303.08774 [cs]. März 2023. DOI: 10.48550/arXiv.2303.08774. URL: <http://arxiv.org/abs/2303.08774> (besucht am 24.08.2023).
- [37] *OpenAI Platform*. en. URL: <https://platform.openai.com> (besucht am 30.08.2023).
- [38] Jared Palmer, Shu Ding und Max Leiter. *Introducing the Vercel AI SDK: npm i ai - Vercel*. en. Mai 2023. URL: <https://vercel.com/blog/introducing-the-vercel-ai-sdk> (besucht am 29.08.2023).
- [39] Joe Price. *Introducing AI-Generated Product Descriptions Powered by Shopify Magic*. en. Apr. 2023. URL: <https://www.shopify.com/blog/ai-product-descriptions> (besucht am 18.09.2023).
- [40] Alec Radford u. a. *Better language models and their implications*. en-US. URL: <https://openai.com/research/better-language-models> (besucht am 24.08.2023).

- [41] Chris Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering. *Requirements-Engineering und -Management: das Handbuch für Anforderungen in jeder Situation*. ger. 7., aktualisierte und erweiterte Auflage. München: Hanser, 2021. ISBN: 978-3-446-46430-8 978-3-446-45587-0.
- [42] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3 (Juli 1959). Conference Name: IBM Journal of Research and Development, S. 210–229. ISSN: 0018-8646. DOI: 10.1147/rd.33.0210.
- [43] Elvis Saravia. “Prompt Engineering Guide”. In: <https://github.com/dair-ai/Prompt-Engineering-Guide> (Dez. 2022).
- [44] Murat H. Sazli. “A brief review of feed-forward neural networks”. en. In: *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 50.01 (Jan. 2006). Number: 01 Publisher: Ankara Üniversitesi, S. 0–0. ISSN: 2618-6462. DOI: 10.1501/commua1-2_0000000026. URL: <https://dergipark.org.tr/tr/pub/aupse/issue/60555/890416> (besucht am 01.09.2023).
- [45] Kevin Scott. *Microsoft teams up with OpenAI to exclusively license GPT-3 language model*. en-US. Sep. 2020. URL: <https://blogs.microsoft.com/blog/2020/09/22/microsoft-teams-up-with-openai-to-exclusively-license-gpt-3-language-model/> (besucht am 24.08.2023).
- [46] Pichai Sundai. *An important next step on our AI journey*. en-us. Feb. 2023. URL: <https://blog.google/technology/ai/bard-google-ai-search-updates/> (besucht am 30.09.2023).
- [47] *Umsatz mit Lebensmitteln im deutschen Online-Handel bis 2022*. de. URL: <https://de.statista.com/statistik/daten/studie/894997/umfrage/umsatz-mit-lebensmitteln-im-deutschen-online-handel/> (besucht am 19.06.2023).
- [48] Ashish Vaswani u. a. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Dez. 2017. DOI: 10.48550/arXiv.1706.03762. URL: <http://arxiv.org/abs/1706.03762> (besucht am 04.04.2023).
- [49] Ashish Vaswani u. a. “Tensor2Tensor for Neural Machine Translation”. In: *CoRR* abs/1803.07416 (2018). URL: <http://arxiv.org/abs/1803.07416>.

- [50] Deutsches Weininstitut. *Absatz von Wein in Deutschland nach Vertriebsformen 2021*. de. Okt. 2022. URL: <https://de.statista.com/statistik/daten/studie/807775/umfrage/wein-absatz-in-deutschland-nach-einkaufsstuetten/> (besucht am 19.06.2023).

Anhang A Minimale Prompt für das Erstellen einer Beschreibung von Wein

Konfiguration: ChatGPT (Modell GPT-3.5-Turbo), ohne zusätzlichen Einstellungen.

User: Erstelle eine Beschreibung für einen Wein. Die Beschreibung wird in einem Onlineshop genutzt. Es geht um folgenden Wein: BDX REVOLUTION Sauvignon Blanc, Sauvignon Blanc, Weißwein, Französisch.

System: Tauchen Sie ein in die erfrischende Revolution des Geschmacks mit unserem BDX REVOLUTION Sauvignon Blanc. Dieser herausragende französische Weißwein verkörpert die Essenz eines modernen Sauvignon Blancs und entführt Ihre Sinne auf eine unvergessliche Reise.

Der BDX REVOLUTION Sauvignon Blanc beeindruckt mit seiner leuchtend blassen Farbe, die die Lebendigkeit und Frische dieses Weins widerspiegelt. Die Nase wird von einem faszinierenden Bouquet aus Aromen von saftigen Stachelbeeren, spritziger Limette und reifen weißen Pfirsichen umworben. Diese duftenden Noten werden durch zarte Anklänge von frischen Kräutern und einer subtilen mineralischen Note ergänzt, die dem Wein eine unverwechselbare Komplexität verleihen.

Am Gaumen entfaltet der BDX REVOLUTION Sauvignon Blanc eine harmonische Symphonie aus knackiger Säure und fruchtiger Fülle. Die Stachelbeer- und Zitrusaromen setzen sich fort und werden von Nuancen von Passionsfrucht und grüner Paprika begleitet, die dem Wein eine aufregende Vielschichtigkeit verleihen. Die erfrischende Textur und das lebendige Spiel der Aromen machen diesen Wein zu einem wahren Genussmoment.

Genießen Sie den BDX REVOLUTION Sauvignon Blanc gut gekühlt als Begleitung zu leichten Sommergerichten, Meeresfrüchten oder einfach als erhebendes Glas Wein für besondere Augenblicke. Tauchen Sie ein in die neue Ära des Sauvignon Blancs mit diesem bemerkenswerten französischen Meisterwerk, das Ihnen unvergessliche Genussmomente schenkt. Willkommen in der Revolution des Geschmacks!

Anhang B Stakeholderanalyse und Stakeholdermatrix

- Shopbetreiber: Diese Stakeholder sind direkte Nutzer des Plugins und haben ein Interesse daran, dass die generierten Beschreibungstexte ihre Kunden ansprechen und zu Kaufentscheidungen führen. Sie möchten möglicherweise auch eine individuelle Anpassung der Texte vornehmen können.
- Weinproduzenten (Winzer): Die Produzenten liefern den Wein und die strukturierten Produktdaten und haben ein Interesse daran, dass ihre Weine ansprechend und präzise beschrieben werden.
- Kunden: Sie erwarten informative und ansprechende Beschreibungstexte, die ihnen bei der Kaufentscheidung helfen. Kundenfeedback könnte verwendet werden, um die Qualität der generierten Texte zu bewerten und zu verbessern.
- Weinexperten: Experten aus dem Weinhandel sind wichtige Stakeholder, um die Qualität und Relevanz der generierten Texte zu bewerten.
- Redakteure: Im Kontext des Plugins könnten Redakteure an der Erstellung von Vorlagen oder Richtlinien für die generierten Texte mitwirken. Sie könnten Feedback zu den Texten geben und sicherstellen, dass die generierten Beschreibungstexte stilistisch und inhaltlich den Anforderungen entsprechen. Redakteure können auch bei der Entwicklung der sprachlichen Nuancen und Tonalität des generierten Inhalts unterstützen, um sicherzustellen, dass die Texte die gewünschte Markenidentität und Positionierung reflektieren.
- Marketing-Personal: Das Marketing-Personal hat ein Interesse daran, dass die generierten Texte die gewünschte Botschaft und Markenidentität transportieren. Das Marketing-Personal kann auch bei der Festlegung der Zielgruppen und Marketingstrategien unterstützen, um sicherzustellen, dass die generierten Texte die relevanten Kunden ansprechen. Sie können auch Feedback zu den generierten Texten geben und gegebenenfalls Anpassungen oder Optimierungen vorschlagen, um die Marketingziele zu erreichen.

Neben der Aufschlüsselung wurde eine Einfluss-Interessen-Matrix aufgestellt:

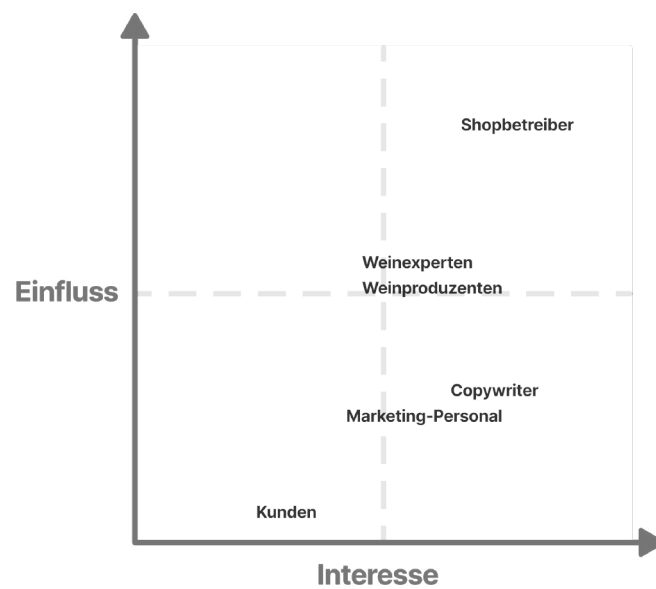


Abbildung 23: Einfluss-Interesse Matrix der Stakeholder. Eigene Darstellung.

Anhang C Beispiele aus der manuellen Interaktion

Für die dargelegten Beispiele mit Bard und dem LLaMa Modell wurde folgende Prompt genutzt. Durch die Gegebenheit, dass sich gegen eine weitere Nutzung der Modelle entschieden wurde, ist nur ein Beispiel dargelegt.

Prompt: Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Du formulierst die Text förmlich. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Nehme maximal drei passende Speisen in den Text. Nehme die Aromen und Düfte mit in die Beschreibung auf. Die Trinktemperatur und der Alkoholgehalt sollen nicht in der Beschreibung vorkommen.

Das sind die Informationen zu dem Wein:

Herkunftsland: Chile Anbauregion: Valle Central Rebsorte: Carmenere Weingut: Concha y Toro Marke: Casillero del Diablo Produktkategorie: Rotwein Jahrgang: 2020 Geschmacksrichtung: trocken Farbnuance: dunkelrot Flaschengröße: 0,75 l Alkoholgehalt: 13,5 % Restzucker: 2,3 g/l Säure: 5,2 g/l Trinktemperatur: 16-18 °C Passende Speisen: Knoblauch-Champignons, Sbrinz, Gruyère, leichte Fleischgerichte, Barbecue Hersteller: Concha y Toro, Av. Nueva Tajamar 481, Torre Norte, P 15, Las Condes, Santiago de Chile - CHILE

C.I Beispielantwort von Bard

Generierte Beschreibung: Der Casillero del Diablo Carmenere Reserva 2020 ist ein edler Rotwein aus Chile, der aus der Rebsorte Carmenère hergestellt wird. Er ist trocken und hat eine tiefrote Farbe. In der Nase entfalten sich Aromen von reifen Pflaumen, schwarzen Johannisbeeren, Gewürzen und Kaffee. Am Gaumen ist der Wein vollmundig und weich mit einer guten Balance zwischen Säure und Tanninen. Der Casillero del Diablo Carmenere Reserva 2020 ist ein idealer Begleiter zu würzigen Speisen wie Knoblauch-Champignons, Sbrinz oder Gruyère. Er passt auch gut zu leichten Fleischgerichten wie einem Grillabend mit Freunden. Fazit: Ein eleganter und ausgewogener Rotwein, der mit seinen Aromen und seinem Körper begeistert.

C.II Beispielantwort LLaMa

Anfrage an das Modell Meta/llama70b-v2-chat über den Vercel SDK Playground, mit der Konfiguration `temperature: 0.5` und `top_p: 0`. Wiederholte Anfragen haben nahezu

identische Texte generiert.

Generierte Beschreibung: Señorío de Casillero del Diablo: A Rich and Elegant Carmenere from Chile's Valle Central

Experience the richness and elegance of Chile's finest Carmenere with Señorío de Casillero del Diablo, a premium rotwein from the renowned Concha y Toro winery. This 2020 vintage boasts a deep, dark red color and a dry, full-bodied taste, making it the perfect accompaniment to a variety of dishes.

The aroma is a delightful blend of black fruit, spices, and subtle hints of oak, while the palate is rich and velvety, with flavors of dark berries, cherries, and a touch of minerality. The finish is long and satisfying, with a subtle sweetness and a hint of spice.

This wine pairs beautifully with a variety of dishes, including light meat fare, such as grilled steak or chicken, as well as rich, creamy cheeses like Gruyère. It also complements hearty mushroom dishes, like Knoblauch-Champignons or Sbrinz, and is an excellent choice for a sophisticated barbecue.

With an alcohol content of 13.5% and a rest sugar level of 2.3 g/l, this wine is a true delight for the senses. Its acidity is well-balanced, with a level of 5.2 g/l, making it a versatile choice for a variety of occasions.

Don't miss out on the opportunity to taste the richness and elegance of Señorío de Casillero del Diablo. Order now and experience the unparalleled quality of this premium Chilean rotwein.

Anhang D Prompt Iterationen

Im Folgenden werden die wichtigsten Prompt Iterationen aufgelistet. Es sind nur größerer Änderungen gelistet. Änderungen von einzelnen Worten werden nicht aufgelistet. Die *Liste der Eigenschaften* sind die strukturierten Daten über einen Wein. Der Platzhalter TONALITÄT kann verschiedene Ausprägungen haben.

1. Schreibe eine Beschreibung für diesen Wein: *Liste der Eigenschaften*.
2. Schreibe eine Beschreibung für diesen Wein: *Liste der Eigenschaften*. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll 10 Sätze lang sein.
3. Schreibe eine Beschreibung für diesen Wein: *Liste der Eigenschaften*. Die Beschreibung wird in einem Onlineshop genutzt. Schreibe 10 Sätze über den Wein. Das ist

ein Beispieltext für diesen Wein, nehme wenn vorhanden nur die Aromen aus diesem Text auf: Ursprüngliche Beschreibung.

4. Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es soll nur eine Auswahl der passenden Speisen vorkommen. *Liste der Eigenschaften.*
5. Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Du formulierst die Texte TONALITÄT. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es soll nur eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung auf. Die Trinktemperatur und der Alkoholgehalt sollen nicht in der Beschreibung vorkommen. Das sind die Informationen zu dem Wein: *Liste der Eigenschaften.*
6. Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Du formulierst die Texte freundlich. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es soll nur eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung auf. Das sind die Informationen zu dem Wein: *Liste der Eigenschaften.* Die folgenden Informationen sind auch über den selben Wein, sollen aber nicht explizit in der Beschreibung genannt werden. Nutze sie um den Wein besser zu verstehen: *Liste der Eigenschaften, die der Redakteur ausgewählt hat.*

Anhang E Ergebnisse der automatisierten Interaktion

An dieser Stelle werden exemplarische Antworten der Modelle dargelegt. Die meisten Anfragen an die Modelle sind nur genutzt worden, um die Prompt zu validieren. Da diese keinen Mehrwert bieten werden nur die Antworten von Prompts dargelegt, die auch zu einem Ergebnis geführt haben. Für die nun folgenden Beispiele sind diese Prompts genutzt worden:

Prompt (A): Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Formuliere den Text TONALITÄT. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es sollen nur eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung

auf. Hier sind die Informationen zu dem Wein:

name: Piedemonte, Gamma Blanco, DO Navarra Herkunftsland: Spanien, Anbauregion: Navarra, Weingut: Bodegas Piedemonte, Marke: Piedemonte, Produktkategorie: Weißwein, Qualitätsstufe: D.O., Cuvee: Chardonnay, Moscatel, Viura, Jahrgang: 2022, Geschmacksrichtung: trocken, Farbnuance: strohgelb, Passende Speisen: Sehr gut zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten, Hinweis auf Allergene: enthält Sulfite, Hersteller: Bodegas Piedemonte, Rua Romana, 0 - ES 31390 Olite, Navarra, Spanien,

Prompt (B): Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Formuliere den Text TONALITÄT. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es sollen nur eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung auf. Hier sind die Informationen zu dem Wein:

name: Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, Herkunftsland: Italien, Anbauregion: Friaul, Weingut: Albino Armani, Marke: Albino Armani, Produktkategorie: Weißwein, Qualitätsstufe: DOC, Cuvee: , Jahrgang: 2022, Geschmacksrichtung: trocken, Farbnuance: strahlendes Strohgelb, Passende Speisen: , Hinweis auf Allergene: enthält Sulfite, Hersteller: Cantina Vini Armani , Località Ceradello, Via Ceradello 401, IT-37020 Dolcè - Italien,

Die Tonalität hatte die Ausprägung freundlich, professionell, verspielt und dramatisch.

E.I Luminous-Base-Control

E.I.1 Freundlich (Prompt A)

Der Weißwein Piedemonte, Gamma Blanco, DO Navarra ist ein perfekter Begleiter für alle Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten. Mit seinem frischen und fruchtigen Aroma von Aprikose, Ananas und Zitrusfrüchten passt er perfekt zu den Aromen der Speisen und bringt sie zum Vorschein. Mit seiner leichten Süße und knackigen Säure ist er ein idealer Begleiter für leichte Gerichte. Der Wein wurde aus drei Weißweintruben hergestellt, aus denen er seine einzigartige Persönlichkeit und seinen unverwechselbaren Geschmack bezieht. Er ist trocken, mit einem niedrigen Alkoholgehalt von 12,5 % und enthält Sulfite.

E.1.2 Professionell (Prompt A)

#Weinbeschreibung: Du hast Lust auf einen frischen und fruchtigen Weißwein? Dann ist unser Piedemonte Gamma Blanco genau das Richtige für dich! Dieser spanische Weißwein aus Navarra zeichnet sich durch seine zarte strohgelbe Farbe und sein feines, fruchtiges Aroma aus.

Das Geschmacksprofil des Gamma Blanco ist trocken und ausgewogen mit Aromen von Zitrusfrüchten, Apfel und grünem Pfeffer. Dieser Weißwein passt perfekt zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten.

Unser Gamma Blanco ist ein unkomplizierter und erfrischender Wein, der sich hervorragend als Begleiter für jede Mahlzeit eignet. Probier ihn am besten leicht gekühlt und genieße die fruchtigen Noten und die erfrischende Qualität dieses Weißweins. #PiedemonteGammaBlanco #Weißwein #SpanischerWein #Navarra #RoteTraube #Wein #Weinliebhaber #Weinverkostung #Weinprobe #Weinreise #Weinempfehlung #Weinkenner

E.1.3 Verspielt (Prompt A)

Der Weißwein aus Navarra ist ein wahres Geschmacksfeuerwerk. Die Aromen von tropischen Früchten und Zitrusfrüchten, die an eine frische Kokosnuss erinnern, mischen sich mit den subtilen Noten von Blumen und Kräutern, die an einen Frühlingsmorgen erinnern. Der trockene Weißwein hat eine weiche Textur und einen langen Abgang, der die Sinne mit jeder Nuance dieses einzigartigen Geschmacks verwöhnt. Perfekt zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten, wird dieser Weißwein Sie mit Sicherheit beeindrucken.

E.1.4 Dramatisch (Prompt A)

Ein süßer und erfrischender Weißwein aus Navarra, Spanien - der Piedemonte Gamma Blanco DO Navarra 2022. Dieser Weißwein präsentiert sich in einem hellen Strohgelb mit goldenen Reflexen. In der Nase duftet es nach frischen und süßen tropischen Früchten wie Ananas und Mango, mit einem Hauch von Zitrusfrüchten wie Limette und Grapefruit. Am Gaumen ist er vollmundig und weich, mit einer schönen Balance aus Süße und Säure. Dieser Weißwein passt perfekt zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten. Genießen Sie ihn gut gekühlt und genießen Sie seine erfrischende und süße Fruchtnote.

E.1.5 Freundlich (Prompt B)

Der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb mit einem frischen und fruchtigen Duft, der an reife Pfirsiche, Aprikosen und Birnen erinnert. Am Gaumen ist er vollmundig und knackig mit einem frischen, lebhaften Abgang. Er passt hervorragend zu leichten Gerichten wie Salaten, Meeresfrüchten und Gemüse. Genießen Sie ihn gekühlt bei 8-10 Grad.

E.1.6 Professionell (Prompt B)

Der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb. In der Nase duftet er nach gelben Äpfeln, Ananas und Zitrusfrüchten. Am Gaumen ist er frisch, knackig und fruchtig mit einer leichten Säure. Er passt hervorragend zu Meeresfrüchten, Salaten und hellem Fleisch. Dieser Wein ist trocken, mit einem Alkoholgehalt von 12,5 % und einem Restzuckergehalt von 2,5 g/l. Er enthält Sulfite.

E.1.7 Verspielt (Prompt B)

Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb. Dieser trockene Weißwein hat ein reichhaltiges, anhaltendes Aroma mit Noten von gelben Äpfeln, Aprikosen und Pfirsichen. Dieser Weißwein passt hervorragend zu Meeresfrüchten wie gebratenem oder gegrilltem Fisch, Meeresfrüchten, Hummer, Jakobsmuscheln und vielen anderen Gerichten.

E.1.8 Dramatisch (Prompt B)

Der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb mit einem Hauch von grünem Apfel und Zitrusfrüchten. Dieser trockene Weißwein wird Sie mit seinen erfrischenden Aromen von grünem Apfel, Zitrusfrüchten und Mineralien überraschen. Genießen Sie ihn zu leichten Gerichten wie gegrilltem Fisch, Meeresfrüchten, Salaten oder Gemüse. Er wird auch zu hellem Fleisch wie Huhn, Pute und Kaninchen passen. Dieser Wein ist ein Muss für jeden Weinkenner und wird sicher zu Ihrem Lieblingswein werden!

E.II Luminous-Supreme-Control

E.II.1 Verspielt (Prompt A)

Der Weißwein Piedemonte Gamma Blanco stammt aus der Region Navarra in Spanien. Sein Geschmack ist trocken und eignet sich hervorragend zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten. Das Aroma dieses Weißweins ist blumig mit Noten von reifer Ananas und gelben Früchten. Sein Abgang ist vollmundig und lang anhaltend. Genießen Sie ihn bei einer angenehmen Trinktemperatur von 8-10 Grad.

E.II.2 Freundlich (Prompt A)

Der Wein Piedemonte Gamma Blanco ist ein trockener Weißwein aus der DO Navarra in Spanien. Er wird aus Chardonnay, Moscatel und Viura hergestellt und hat einen starken und angenehmen Duft nach tropischen Früchten mit einem Hauch von Mandeln. Am Gaumen ist er vollmundig und weich, mit einem langen Abgang. Er passt hervorragend zu allen Arten von Fisch, Schweinefleisch und Geflügel, zu Paellas und Nudelgerichten. Genießen Sie ihn bei einer Trinktemperatur von 8-10 Grad.

E.II.3 Professionell (Prompt A)

Chardonnay, Moscatel und Viura vereinen sich zu einem trockenen Weißwein aus Navarra, Spanien. Er zeigt Noten von tropischen Früchten und Zitrusfrüchten, die von blumigen und würzigen Aromen begleitet werden. Dieser leichte und erfrischende Wein passt perfekt zu allen Arten von Fisch, Schweinefleisch und Geflügel, aber auch zu Paellas und Nudelgerichten. Genießen Sie ihn bei einer Trinktemperatur von 8-10 Grad. Enthält Sulfite.

E.II.4 Dramatisch (Prompt A)

Der Piedemonte Gamma Blanco ist ein weißer Wein aus der D.O. Navarra in Spanien. Er wird von Bodegas Piedemonte hergestellt und ist ein wahres Vergnügen für die Sinne. Seine strohgelbe Farbnuance und seine trockene Geschmacksrichtung machen ihn zu einem Wein der Extraklasse. Der Piedemonte Gamma Blanco bringt Aromen von Süße und Frische mit sich, die an Pfirsiche, Aprikosen und weiße Blüten erinnern. Sein Duft ist einladend und erinnert an gelbe Früchte und Honig. Genießen Sie diesen Wein zu Fischgerichten, Meeresfrüchten, Geflügel und leichten Salaten. Er wird Ihnen ein unvergessliches

Geschmackserlebnis bereiten!

E.II.5 Verspielt (Prompt B)

Der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb mit einem blumigen und fruchtigen Aroma. Seine blumigen Noten von Jasmin und Rosenblättern werden durch zarte Anklänge von Apfel und Quitte ergänzt. Am Gaumen ist er trocken und knackig mit einem langen Abgang. Dieser Weißwein passt perfekt zu Meeresfrüchten, Geflügel, Salaten und Gemüse. Genießen Sie ihn bei einer kühlen Temperatur von 8-10 Grad.

E.II.6 Freundlich (Prompt B)

Der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ist ein strahlendes Strohgelb, mit einem subtilen Duft nach Apfel, Birne und Zitrusfrüchten. Am Gaumen ist er frisch und knackig mit einer angenehmen Säure, die den Geschmack von Zitrusfrüchten und Apfel unterstützt. Dieser trockene Weißwein passt gut zu Meeresfrüchten, Fisch, Geflügel und hellem Fleisch. Genießen Sie ihn bei einer Temperatur von 8-10 Grad.

E.II.7 Professionell (Prompt B)

Albino Armani ist ein strahlendes Strohgelb mit einer subtilen Perlung. Das Bukett ist elegant und komplex mit Aromen von gelben Früchten, Pfirsichen und Birnen, gefolgt von blumigen Noten von Jasmin und Akazienblüten. Am Gaumen ist er trocken und knackig mit einer guten Struktur und einem langen Abgang. Er passt hervorragend zu Gerichten wie gebratenem Fisch mit Gemüse oder zu hellem Fleisch wie Huhn oder Kaninchen. Genießen Sie ihn bei einer Trinktemperatur von 8-10 Grad.

E.II.8 Dramatisch (Prompt B)

Aromen: Noten von reifen Birnen, Ananas und weißer Pfirsich. Der Albino Armani Pinot Grigio Friuli Grave DOC ist ein wahres Geschenk Italiens. Dieser italienische Weißwein ist ein Jahrgang 2022 und überzeugt mit seinem strahlenden Strohgelb. Er verführt die Sinne mit seinen reifen Birnen-, Ananas- und weißen Pfirsicharomen. Genießen Sie diesen wunderbaren Wein zu einem leckeren Fischgericht, einem knackigen Salat oder einem saftigen Käse. Ein Genuss, der Sie verzaubern wird.

E.III GPT-3.5-Turbo

E.III.1 Verspielt (Prompt A)

Alkoholgehalt: 13 % vol. Der Piedemonte Gamma Blanco ist ein edler Weißwein aus der D.O. Navarra in Spanien. Sein trockener Geschmack und die strohgelbe Farbnuance machen ihn zu einem einzigartigen Genusserlebnis. Er ist eine Cuvee aus Chardonnay, Moscatel und Viura, die harmonisch miteinander verschmelzen. Er passt hervorragend zu Fisch, Schweinefleisch, Geflügel, Paellas und Nudelgerichten. Seine fruchtigen Aromen von Zitrusfrüchten, wie Limetten und Grapefruit, machen ihn zu einem unvergesslichen Geschmackserlebnis. Seine feinen Blütennoten und sein süßer Honigduft schaffen einen unverwechselbaren Duft, der die Sinne verzaubert. Der Piedemonte Gamma Blanco ist ein Wein, der jeden Anlass zu etwas Besonderem macht. Genießen Sie ihn mit einem Alkoholgehalt von 13 % vol. und lassen Sie sich von seiner vollendeten Eleganz verzaubern.

E.III.2 Freundlich (Prompt A)

Aromen und Düfte: Noten von frischen Früchten, Zitrusaromen, Kräuter und Blumen. Der Piedemonte Gamma Blanco D.O. Navarra ist ein Weißwein aus Spanien, der in der Region Navarra hergestellt wird. Er wird von Bodegas Piedemonte unter der Marke Piedemonte produziert. Der 2022er Jahrgang ist ein trockener Wein mit einer strohgelben Farbnuance. Er verfügt über einzigartige Aromen und Düfte von frischen Früchten, Zitrusaromen, Kräutern und Blumen.

Der Piedemonte Gamma Blanco D.O. Navarra ist ein vielseitiger Wein, der sich besonders gut zu allen Arten von Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten eignet. Er ist ein einzigartiger Genuss, der den Gaumen in ein wahres Gartenparadies verwandeln wird. Lasse dich von seinen frischen Aromen und Düften verzaubern und genieße ein unvergessliches Geschmackserlebnis.

Der Piedemonte Gamma Blanco D.O. Navarra ist ein unvergleichlicher Wein, der einzigartige Geschmackserlebnisse bietet. Er ist ein perfekter Begleiter für ein romantisches Abendessen oder ein entspanntes Mittagessen. Genieße seine einzigartigen Aromen und Düfte und erlebe ein unvergessliches Geschmackserlebnis.

E.III.3 Professionell (Prompt A)

Der Piedemonte Gamma Blanco D.O. ist ein trockener Weißwein aus der spanischen Region Navarra. Er ist ein Cuvée aus Chardonnay, Moscatel und Viura und hat eine strohgelbe Farbnuance. Sein Bouquet ist eine wahre Augenweide, die Aromen von frischen Früchten und Blumen erfreuen das Aroma.

Genießen Sie den Piedemonte Gamma Blanco D.O. mit Fisch, Schweinefleisch, Geflügel, Paellas und Nudelgerichten. Seine frischen Fruchtaromen werden jedes Gericht aufwerten. Genießen Sie ein Glas des Piedemonte Gamma Blanco D.O. und erfreuen Sie sich an seinem einzigartigen Geschmack.

Der Piedemonte Gamma Blanco D.O. stammt aus dem Weingut Bodegas Piedemonte in der Region Navarra. Er ist ein qualitativ hochwertiger Wein und enthält Sulfite. Genießen Sie diesen exquisiten Wein und lassen Sie sich von seinem Geschmack und seinen Aromen begeistern.

E.III.4 Dramatisch (Prompt A)

Lassen Sie sich von dem Piedemonte Gamma Blanco, D.O. Navarra verzaubern. Dieser exquisite Weißwein aus dem Weingut Bodegas Piedemonte in Navarra, Spanien, ist eine Cuvée aus Chardonnay, Moscatel und Viura und überzeugt durch seine strohgelbe Farbnuance. Sein trockener Geschmack und seine angenehmen Aromen von frischen Äpfeln, Birnen und Zitrusfrüchten machen ihn zu einem besonderen Genuss. Der Piedemonte Gamma Blanco ist der perfekte Begleiter für Fisch, Schweinefleisch und Geflügel, Paellas und Nudelgerichten. Genießen Sie ihn und lassen Sie sich von seinem einzigartigen Duft und Geschmack verzaubern. Dieser köstliche Wein enthält Sulfite und ist aus dem Weingut Bodegas Piedemonte in Olite, Navarra, Spanien. Erleben Sie ein unvergleichliches Geschmackserlebnis und genießen Sie den Piedemonte Gamma Blanco, D.O. Navarra!

E.III.5 Verspielt (Prompt B)

Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist ein wahres Meisterstück aus Italien. Dieser Weißwein aus dem Jahrgang 2022 überzeugt durch seine strahlende Strohgelb-Farbnuance und sein trockenes Aroma. Der Geschmack ist vollmundig, leicht fruchtig und erinnert an Aromen von Birne, weißer Johannisbeere und einem Hauch von Zitrusfrüchten. Sein Abgang ist frisch und anhaltend. Er passt hervorragend zu Fischgerichten, leichten Salaten und Geflügel. Genießen Sie diesen Wein und lassen Sie sich von

seinen Aromen und Düften verzaubern!

E.III.6 Freundlich (Prompt B)

Aromen: Aromen von weißen Blüten, frischen Äpfeln und Zitronen Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist ein Weißwein aus Italien, der auf den Hügeln des Friaul angebaut wird. Sein strahlendes Strohgelb und seine trockene Geschmacksrichtung machen ihn zu einem wahren Genuss. Sein Bouquet ist reich an Aromen von weißen Blüten, frischen Äpfeln und Zitronen. Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist ein einzigartiger Wein, der sich perfekt als Begleiter zu Fischgerichten, Pasta und Salaten eignet. Genießen Sie ihn bei einem Abendessen mit Freunden und lassen Sie sich von seinem unvergleichlichen Geschmack verzaubern.

E.III.7 Professionell (Prompt B)

Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist ein herrlicher Weißwein aus dem italienischen Friaul. Der Jahrgang 2022 präsentiert sich in einem strahlenden Strohgelb und verströmt ein Aroma von frischen Zitrusfrüchten, Apfel und weißen Blüten. Am Gaumen erfreut er mit einer ausgewogenen Säure und einer lebhaften Fruchtigkeit. Sein trockener Geschmack mit einem Hauch von Mandeln und Honig macht ihn zu einem idealen Begleiter für Fisch- und Geflügelgerichte, aber auch für Salate und leichte Vorspeisen. Ein wunderbarer Wein, der ein wahres Fest für die Sinne ist. Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist eine Freude für jeden Weinliebhaber.

E.III.8 Dramatisch (Prompt B)

Aromen: Aromen von Pfirsich, Birne und Zitrusfrüchten. Der Albino Armani Pinot Grigio Friuli Grave DOC Friuli ist ein edler Weißwein aus Italien. Er wird in der Anbauregion Friaul von dem Weingut Albino Armani produziert. Der Wein ist trocken und hat eine strahlende Strohgelb Farbnuance. Seine Aromen von Pfirsich, Birne und Zitrusfrüchten machen ihn zu einem idealen Begleiter für Fisch, Meeresfrüchte und leichte Gemüsegerichte. Er ist ein toller Wein, der einzigartige Düfte und Aromen verströmt und ein unvergessliches Geschmackserlebnis bietet.

E.IV GPT-4

E.IV.1 Freundlich (Prompt A)

Entdecken Sie den Piedemonte, Gamma Blanco, DO Navarra - ein herrlicher Weißwein aus dem sonnigen Spanien. Dieser strohgelbe Tropfen, liebevoll hergestellt von Bodegas Piedemonte, ist eine harmonische Cuvée aus Chardonnay, Moscatel und Viura. Sein trockener Geschmack und die feinen Aromen machen ihn zum perfekten Begleiter für Fischgerichte, Schweinefleisch, Geflügel, Paellas und Nudelgerichte. Lassen Sie sich von diesem spanischen Juwel verzaubern!

E.IV.2 Professionell (Prompt A)

Der Piedemonte, Gamma Blanco, DO Navarra ist ein exzellenter Weißwein aus der renommierten Anbauregion Navarra in Spanien. Dieser Wein der Qualitätsstufe D.O. präsentiert sich in einem eleganten Strohgelb und besticht durch seine trockene Geschmacksrichtung. Die Cuvée aus Chardonnay, Moscatel und Viura entfaltet ein komplexes Aromenspektrum. Er harmoniert besonders gut mit Fisch, Schweinefleisch, Geflügel, Paellas und Nudelgerichten. Bitte beachten Sie, dass der Wein Sulfite enthält.

E.IV.3 Verspielt (Prompt A)

Tauchen Sie ein in die Welt des Piedemonte, Gamma Blanco, DO Navarra! Dieser spritzige Weißwein aus Spanien ist wie ein sonniger Tag am Strand. Seine strohgelbe Farbe erinnert an Sonnenstrahlen, während die Cuvée aus Chardonnay, Moscatel und Viura ein wahres Feuerwerk der Aromen entfacht. Ob zu Fisch, Schwein, Geflügel, Paella oder Pasta - dieser Wein macht jedes Essen zu einem Fest.

E.IV.4 Dramatisch (Prompt A)

Der Piedemonte, Gamma Blanco, DO Navarra ist kein gewöhnlicher Weißwein. Er ist ein Meisterwerk aus dem Herzen Spaniens, genauer gesagt aus der Anbauregion Navarra. Seine strohgelbe Farbe ist so intensiv wie das spanische Sonnenlicht. Die Cuvée aus Chardonnay, Moscatel und Viura entfaltet eine Symphonie von Aromen, die Sie in ihren Bann ziehen wird. Er ist der perfekte Begleiter für Fisch, Schweinefleisch, Geflügel, Paellas und Nudelgerichte.

E.IV.5 Freundlich (Prompt B)

Entdecken Sie den unverwechselbaren Charme des Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli. Dieser italienische Weißwein aus dem renommierten Weingut Albino Armani in Friaul ist ein wahrer Genuss für die Sinne. Seine strahlend strohgelbe Farbe und sein trockener Geschmack machen ihn zu einem perfekten Begleiter für verschiedene Gerichte.

E.IV.6 Professionell (Prompt B)

Präsentiert wird hier der Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli, ein Weißwein von höchster Qualitätsstufe aus dem Hause Albino Armani. Herkunftsort dieses edlen Tropfens ist die italienische Anbauregion Friaul. Der Wein besticht durch seine strahlend strohgelbe Farbe und seine trockene Geschmacksrichtung.

E.IV.7 Verspielt (Prompt B)

Lassen Sie sich verzaubern vom Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli. Dieser strahlend strohgelbe Weißwein aus Italien ist wie ein Sonnenstrahl in Ihrem Weinglas. Sein trockener Geschmack und seine Herkunft aus dem sagenhaften Friaul machen ihn zu einem echten Abenteuer für Ihren Gaumen.

E.IV.8 Dramatisch (Prompt B)

Treten Sie ein in die Welt des Albino Armani, Pinot Grigio Friuli Grave, DOC Friuli. Dieser Weißwein aus dem Hause Albino Armani ist eine wahre Offenbarung. Seine strahlend strohgelbe Farbe ist wie das Licht der italienischen Sonne, sein trockener Geschmack ein Echo der kargen Schönheit Friauls.

Anhang F Quellcode Beispiele der Plugin Implentierung

An dieser Stelle sind die wichtigsten Quellcode Beispiele dargestellt. Der vollständige Quellcode ist in folgendem GitHub Repository einsehbar: <https://github.com/WasMachenSachen/master-thesis-tim-loges>

F.I Die Service XML Datei für die Dependency Injection

```

1 <?xml version="1.0"?>
2 <container xmlns="http://symfony.com/schema/dic/services"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://symfony.com/schema/dic/services http://
   symfony.com/schema/dic/services/services-1.0.xsd">
5   <services>
6     <service id="AiDescription\AiDescription" />
7     <service id="AiDescription\Service\ReadingData">
8       <argument type="service" id="product.repository" />
9       <argument type="service" id="ai_description_content.repository" />
10    </service>
11    <service id="AiDescription\Service\HistoryService">
12      <argument type="service" id="ai_description_content.repository" />
13    </service>
14    <service id="AiDescription\Service\CallApi">
15      <argument type="service" id="Shopware\Core\System\SystemConfig\
16        SystemConfigService" />
17    </service>
18    <service id="AiDescription\Service\ComposePrompt" class="
19      AiDescription\Service\ComposePrompt">
20    </service>
21    <service id="AiDescription\Controller\SettingsController" public="
22      true">
23      <argument type="service" id="AiDescription\Service\ReadingData" />
24      <argument type="service" id="Shopware\Core\System\SystemConfig\
25        SystemConfigService" />
26      <call method="setContainer">
27        <argument type="service" id="service_container" />
28      </call>
29    </service>
30    <service id="AiDescription\Controller\GeneratorController" public="
31      true">
32      <argument type="service" id="AiDescription\Service\CallApi" />
33      <argument type="service" id="AiDescription\Service\ComposePrompt"
34      />
35      <argument type="service" id="AiDescription\Service\HistoryService"
36      />
37      <call method="setContainer">
38        <argument type="service" id="service_container" />

```

```
32     </call>
33 </service>
34 <service id="AiDescription\Core\Content\AiDescriptionDefinition">
35     <tag name="shopware.entity.definition" entity="
ai_description_content" />
36 </service>
37
38 </services>
39 </container>
```

Quellcode 8: Die service.xml

F.II ComposePrompt Service

```
1 <?php declare(strict_types=1);
2
3 namespace AiDescription\Service;
4
5 /**
6  * Class ComposePrompt
7  *
8  * This class is responsible for composing prompts and messages for
9  * generating and rephrasing descriptions.
10 */
11 class ComposePrompt
12 {
13     /**
14      * Compose instructions for generation a wine description.
15      *
16      * @param string $tonality The tonality of the description
17      * @param int $maxLength The maximum length of the description in
18      * words
19      * @return string The composed instructions
20      */
21     public function composeInstructionsForGeneration(string $tonality,
22         int $maxLength = 200): string
23     {
24         $instructions = <<<PROMPT
25         Du bist ein Experte für Wein und sollst eine Beschreibung für
26         einen Wein erstellen.
27
28         Du formulierst die Texte $tonality. Die Beschreibung wird in
29         einem Onlineshop genutzt.
30
31         Die Beschreibung soll $maxLength Wörter lang sein. Es soll eine
32         Auswahl der passenden Speisen vorkommen.
33         Nehme die Aromen und Düfte mit in die Beschreibung auf.
34         PROMPT;
35
36         return $instructions;
37     }
38
39     /**
40      * This method composes the the message with the given attributes.
41      *
42      */
43 }
```

```
34  * @param array $attributes The attributes of the wine
35  * @return string The composed message
36  */
37  public function composeMessageForGeneration(array $attributes):
    string
38  {
39      $includedAttributesString = "";
40      $excludedAttributesString = "";
41
42      foreach ($attributes as $property) {
43          // we use the checkbox state (checked or not) from the
44          // frontend, to determine if the attribute should be included or
45          // excluded
46          $property->checked === true ?
47              $includedAttributesString .= "{$property->name}: {
48              $property->options[0]->name}. \n" :
49              $excludedAttributesString .= "{$property->name}: {
50              $property->options[0]->name}. \n";
51      }
52      $initialMessage = <<<PROMPT
53      Hier sind die Informationen über den Wein:
54      $includedAttributesString
55
56      Die folgenden Informationen sind auch über den selben Wein,
57      sollen aber nicht explizit in der Beschreibung genannt werden.
58      Nutze sie um den Wein besser zu verstehen:
59      $excludedAttributesString
60      PROMPT;
61
62      return $initialMessage;
63  }
64
65  /**
66   * This method composes a prompt for rephrasing a wine description
67   * in a specific tonality.
68   *
69   * @param string $tonality The tonality in which the description
70   * should be rephrased
71   * @return string The composed prompt for rephrasing
72   */
```

```
66 public function composePromptForRephrasing(string $tonality):  
    string  
67 {  
68     $instructions = <<<PROMPT  
69     Lese den Text der dir zur Verfügung gestellt wird.  
70     Formuliere den Satz innerhalb der span Elemente mit dem data-  
    change Attribute neu, in einem $tonality Ton.  
71     Der Rest des Textes soll gleich bleiben.  
72     Behalte die HTML Formatierung bei. Entferne in der Ausgabe das  
    data-change Attribute.  
73     PROMPT;  
74  
75     return $instructions;  
76 }  
77 }
```

Quellcode 9: Der ComposePrompt Service.

F.III JSON Body für eine Anfrage und eine Antwort an die OpenAI API

```
1 {
2   "model": "gpt-4",
3   "messages": [
4     {
5       "role": "system",
6       "content": "Du bist ein Experte für Wein und sollst eine
7         Beschreibung für einen Wein erstellen.\nDu formulierst die Texte
8         verspielt. Die Beschreibung wird in einem Onlineshop genutzt.\n
9         Die Beschreibung soll 200 Wörter lang sein. Es soll eine
10        Auswahl der passenden Speisen vorkommen.\nNehme die Aromen und D
11        üfte mit in die Beschreibung auf."
12     },
13     {
14       "role": "user",
15       "content": "Hier sind die Informationen über den Wein:\n
16        \nAnbauregion: Valle Central. \nGeschmacksrichtung: trocken. \n
17        \nHerkunftsland: Chile. \nHinweis auf Allergene: enthält Sulfite. \n
18        \nInverkehrbringer: Mack & Schühle AG, Neue Straße 45, 73277
19        Owen Teck, Deutschland. \nJahrgang: 2013. \nPassende Speisen:
20        Fisch und Schalentiere, Geflügel und helles Fleisch, zarte
21        Gerichte von Schwein, Kalb und Rind, gekochtes Fleisch,
22        Pfannengerichte, Fleisch in hellen Soßen. \nProduktkategorie:
23        Weißwein. \nQualitätsstufe: D.O.. \nRebsorte: Sauvignon Blanc. \n
24        \nWeingut: Cono Sur. \n\n\nDie folgenden Informationen sind auch
25        über den selben Wein, sollen aber nicht explizit in der
26        Beschreibung genannt werden.\nNutze sie um den Wein besser zu
27        verstehen:\nFlaschengröße: 0,75 l. \nRestzucker: 3,8 g/l. \nSä
28        ure: 5,4 g/l. \nTrinktemperatur: 16-18 C. \n"
29     }
30   ],
31   "max_tokens": 500,
32   "temperature": 0.4,
33   "top_p": 0.2,
```

```
16 "frequency_penalty": 0,
17 "presence_penalty": 0
18 }
```

Quellcode 10: Beispiel Anfrage an die OpenAI API.

```
1 {
2   "id": "chatcmpl-81byFTVjJ9Bb4UHi3O6fuK2xGRWFu",
3   "object": "chat.completion",
4   "created": 1695394979,
5   "model": "gpt-4-0613",
6   "choices": [
7     {
8       "index": 0,
9       "message": {
10        "role": "assistant",
11        "content": "Tauchen Sie ein in die verzauberte Welt des
Cono Sur Sauvignon Blanc, ein trockener Weißwein, der in der
malerischen Anbauregion Valle Central in Chile geboren wurde.
Dieser edle Tropfen aus dem Jahr 2013 ist ein wahrer Genuss für
die Sinne, der Sie auf eine sinnliche Reise durch die Aromen und
Düfte der chilenischen Weinlandschaft entführt.\n\nMit jedem
Schluck dieses Weines entfaltet sich ein Bouquet von frischen,
lebendigen Aromen, die an einen sonnigen Tag in einem üppigen
Obstgarten erinnern. Die Rebsorte Sauvignon Blanc verleiht ihm
eine spritzige Säure und eine subtile Süße, die sich harmonisch
mit den Noten von grünen Äpfeln, Stachelbeeren und frisch
geschnittenem Gras verbindet."
12      }
13    }
14  ]
15 }
```

Quellcode 11: Beispiel Antwort der OpenAI API

F.IV Der Einstiegspunkt der Administrations Erweiterung

main.js

```
1 import "../page/sw-product-detail";
2 import "../view/tab-page";
3
4 Shopware.Module.register("sw-new-tab-ai", {
5   routeMiddleware(next, currentRoute) {
6     const customRouteName = "sw.product.detail.ai";
7     if (
8       currentRoute.name === "sw.product.detail" &&
9       currentRoute.children.every((currentRoute) => currentRoute.name
10        !== customRouteName)
11     ) {
12       currentRoute.children.push({
13         name: customRouteName,
14         path: "/sw/product/detail/:id/ai",
15         props: true,
16         component: "tab-page",
17         meta: {
18           parentPath: "sw.product.index",
19         },
20       });
21     }
22     next(currentRoute);
23   },
24 });
```

Quellcode 12: Die main.js des Plugins

Anhang G Evaluation durch einen Winzer

G.1 Fragenkatalog für persönliche Korrespondenz mit Winzer

Bezogen auf die generierten Texte:

1. Sind Aromen und Düfte der Weine korrekt dargestellt?
2. Welche Zielgruppe siehst du durch die Beschreibung angesprochen?

Allgemeine Fragen:

3. Welche Zielgruppe haben deine Weine?
 - Gelegenheitstrinker (kann vlt. schon durch einen „schönen“ Marketingtext überzeugt werden)
 - Weinkenner (versierter als der vorherige, aber immer noch kein Experte)
 - Experten (hat bspw. viel höheren Anspruch an Korrektheit der dargestellten Aromen und Düfte)
4. Welche Merkmale eines Weins sollten in einer Beschreibung hervorgehoben werden, um das Interesse der Zielgruppe zu wecken?
5. Erwartest du, dass Auszeichnungen und Pressebewertungen (wenn vorhanden) in der Weinbeschreibung aufgenommen werden?
6. Gibt es Attribute, die immer in einer Beschreibung vorkommen sollten?

Die Attribute, die wir zur Verfügung haben, sind folgende:

- Herkunftsland
- Region
- Anbauregion
- Rebsorte
- Weingut
- Marke
- Produktkategorie
- Qualitätsstufe

- Cuvee
- Jahrgang
- Geschmacksrichtung
- Farbnuance
- Flaschengröße
- Alkoholgehalt
- Restzucker
- Säure
- Trinktemperatur
- Passende Speisen
- Hinweis auf Allergene
- Auszeichnungen
- Hersteller

7. Welche Probleme siehst du oder erwartest du bei automatisch generierten Texten?

G.II Email Korrespondenz mit Winzer

Für ein unvoreingenommenes Feedback wurden im ersten Schritt eine Auswahl an Texten an einen Winzer geschickt. Ohne Informationen über Herkunft und Ersteller. Die Kontaktaufnahme wurde durch den Partner in dieser Arbeit durchgeführt. Anrede und Footer wurde entfernt.

Erste Anfrage per E-Mail am 20.08.23: um besser bei Google zu ranken haben wir noch mal individuelle Beschreibungstexte für Deine Weine erstellen lassen. Kannst Du mal drüber lesen, wie Du die Texte findest und was ggfs. angepasst werden sollte?

Antwort des Winzers am 03.09.23: Wow, da habt Ihr Euch richtig viel Arbeit gemacht! Das liest sich alles super und kann so bleiben!

Danke für Euren Einsatz!

G.III Persönliche Korrespondenz mit Winzer für Evaluation

Hierbei handelt es sich um ein Gedächtnisprotokoll von einem Telefonat am 11.09.2023 mit einem Winzer. Ursprünglich war vorgesehen, dass der in Anhang G.I aufgestellte Fragenkatalog asynchron beantwortet wird.

I: Ich schreibe aktuell meine Masterarbeit bei Gjuce über das automatische Generieren von Produktbeschreibungen. Die Möglichkeit haben wir genutzt, um für die Weine neue Texte zu erstellen. Dafür wird eine KI genutzt (ähnlich wie ChatGPT), die anhand der Steckbriefe von Vindor.de neue Texte erstellt. Ist dir das beim ersten Lesen der Texte aufgefallen?

W: Ich habe nicht bemerkt, dass die Texte automatisch generiert wurden. Jetzt beim zweiten Lesen würde ich sagen, das passt nicht 100% aber ist schon sehr nahe. Experten machen aber in der Regel auch Fehler, deswegen ist es mir nicht aufgefallen.

I: Sind Aromen und Düfte der Weine korrekt dargestellt?

W: Grundsätzlich sind Aromen und Düfte passend. Liegt wahrscheinlich daran, dass sie allgemein gehalten wurden und sich oft von der Rebsorte ableiten lassen.

I: Welche Merkmale eines Weins sollten in einer Beschreibung hervorgehoben werden, um das Interesse der Zielgruppe zu wecken?

W: Andersherum ist es einfacher zu sagen: Formulierungen wie die *Reise durch* würde man nicht schreiben. Das klingt oft nach einem Verschnitt von verschiedenen Sorten, was ungern gesehen wird. Ebenfalls Formulierungen die Regionen oder viele Regionen betreffen klingen nach Verschnitt. Teilweise waren Redundanzen vorhanden vor allem, wenn es um die Region Mosel ging.

I: Welche Zielgruppe haben deine Weine?

W: Eine genaue Zielgruppe gibt es nicht. Ich sieht aber die Zielgruppe zwischen 35 und 55. Finanziell müssen sie auch mal Wein über fünf Euro kaufen müssen, als Belohnung.

Menschen mit Kindern, bisschen genussorientierte.

I: Hast du sonst noch Anmerkungen?

W: Interessant wäre noch Informationen zur „Next Generation“. Dabei geht es um neue Rebsorte, die ein höheren Pilzwiderstand hat und deswegen weniger gespritzt werden muss, mit dem Argument der Nachhaltigkeit. Das sind aber Informationen, die nicht in der strukturierten Daten vorliegen, sondern in dem Fall im Rücken Etikett zu erkennen sind. Die habt ihr nicht genutzt oder? Ist vielleicht interessant. Ich sehe es auch so, dass die Analyse Daten nicht explizit im Text genannt werden müssen. Für die Profis ist es dann ja trotzdem erkennbar, das sind aber echt sehr wenige, die dann wirklich Interesse daran haben zu lesen wie hoch der Säuregehalt ist.

G.IV Beschreibungstexte für die Evaluierung

Die Prompt ist für alle folgenden Beschreibungstexte identisch: *Du bist ein Experte für Wein und sollst eine Beschreibung für Wein erstellen. Formuliere den Text verspielt. Die Beschreibung wird in einem Onlineshop genutzt. Die Beschreibung soll nicht länger als 200 Wörter sein. Es sollen nur eine Auswahl der passenden Speisen vorkommen. Nehme die Aromen und Düfte mit in die Beschreibung auf. Hier sind die Informationen zu dem Wein: Herkunftsland: Deutschland,*

Region: Mosel,

Anbauregion: Mosel,

Weingut: Stairs n' Roses,

Marke: Stairs n' Roses,

Hersteller: Stairs n' Roses, Julia Molitor-Justen und Dr. Daniel Molitor, In den Viermorgen 8, 54538 Kinheim,

Die folgenden Informationen sind auch über den selben Wein, sollen aber nicht explizit in der Beschreibung genannt werden. Nutze sie um den Wein besser zu verstehen:

Hinweis auf Allergene: enthält Sulfite,

Alle strukturierten Daten, die bei den Weinen unterschiedlich sind, werden im folgenden aufgelistet. Genutzt wurde das OpenAI Modell GPT-4 mit der Konfiguration: temperature: 0.5, top_p: 0 , presence_penalty: 0.2, frequency_penalty: 0.2.

G.IV.1 Stairs n' Roses, „Struwwelpitter“, QbA Mosel

URL: <https://www.vindor.de/stairs-n-roses-struwwelpitter-qba-mosel>

Strukturierte Daten:

Name: Stairs n' Roses, SStruwwelpitter", QbA Mosel,

Rebsorte: Cabernet Blanc, Sauvignac, Müller-Thurgau, Riesling,

Produktkategorie: Weißwein,

Qualitätsstufe: QbA,

Cuvee: 60% Riesling, 30% Müller-Thurgau, 5% Sauvignac, 5% Cabernet blanc,

Jahrgang: 2020,

Geschmacksrichtung: trocken,

Farbnuance: helles Gelb mit grünlichen Reflexen,

Passende Speisen: leichte Speisen, weißer Fisch mit Zitrone, Salate,

Generierte Beschreibung:

Der Wein „Struwwelpitter“ begeistert mit seinem frischen und fruchtigen Aroma von tropischen Früchten, Apfel, Zitrone und Ananas. Die sanfte Süße und die leichte Säure verleihen ihm einen erfrischenden Geschmack, der sich perfekt für leichte Gerichte wie weißen Fisch mit Zitrone oder Salat eignet. Mit seinen fruchtigen Aromen und seiner sanften Süße ist dieser Wein ein großartiger Begleiter für jede Gelegenheit. Probieren Sie ihn und genießen Sie einen Schluck Sommer auf der Zunge!

G.IV.2 Stairs n' Roses, Stairs n' Roses Riesling feinherb, QbA Mosel

URL:

<https://www.vindor.de/stairs-n-roses-stairs-n-roses-riesling-feinherb-qba-mosel>

Strukturierte Daten:

Name: Stairs n' Roses, Stairs n' Roses Riesling feinherb, QbA Mosel,

Rebsorte: Riesling,

Produktkategorie: Weißwein,

Qualitätsstufe: QbA,

Cuvee: ,

Jahrgang: 2021,

Geschmacksrichtung: feinherb,

Farbnuance: gelb,

Passende Speisen: Asiatische Speisen, fruchtige Desserts,

Generierte Beschreibung:

Tauchen Sie ein in die zauberhafte Welt des Stairs n' Roses Rieslings, ein feinherber Tropfen aus dem Herzen der Mosel. Dieser edle Weißwein verzaubert mit seiner leuchtend gelben Farbnuance und entführt Sie auf eine sinnliche Reise durch die romantischen Weinberge Deutschlands.

Seine Aromen sind ein verspieltes Ballett aus zarter Zitrusfrucht und knackigem grünen Apfel, die sich harmonisch mit subtilen Noten von Pfirsich und Aprikose vereinen. Der Duft ist eine liebeliche Symphonie, die an einen sonnigen Herbsttag erinnert, an dem die Trauben in den Weinbergen der Mosel ihre volle Reife erlangen.

Der Stairs n' Roses Riesling ist ein wahrer Charmeur, der mit seiner feinherben Geschmacksrichtung und der eleganten Säurestruktur verführt. Er ist der perfekte Begleiter

für asiatische Speisen, bei denen er mit seiner Frische und Fruchtigkeit hervorragend kontrastiert. Auch zu fruchtigen Desserts ist er ein wahrer Genuss und rundet jedes Mahl auf seine ganz eigene, unverwechselbare Weise ab.

Ein Glas dieses Rieslings ist wie ein Spaziergang durch die Weinberge der Mosel - voller Leichtigkeit, Fröhlichkeit und dem Versprechen von Genuss. Entdecken Sie den Stairs n' Roses Riesling und lassen Sie sich verzaubern!

G.IV.3 2020 Stairs n' Roses „Sweetest Devotion“

URL: <https://www.vindor.de/2020-stairs-n-roses-sweetest-devotion>

Strukturierte Daten:

Name: 2020 Stairs n' Roses „Sweetest Devotion“ ,

Rebsorte: Riesling,

Produktkategorie: Perlwein,

Qualitätsstufe: Qualitätsperlwein,

Cuvee: ,

Jahrgang: 2020,

Geschmacksrichtung: edelsüß,

Farbnuance: gelb,

Passende Speisen: ,

Generierte Beschreibung:

Der süße Duft von reifer Aprikose und Pfirsich, gepaart mit einer subtilen Zitrusnote, strömt aus diesem perligen Weißwein. Die fruchtigen Aromen werden von einer erfrischenden Säure unterstützt, die eine perfekte Balance zwischen Süße und Säure schafft. Dieser leichte und fruchtige Wein passt hervorragend zu leichteren Gerichten wie Meeresschnecken, Salaten oder hellem Fleisch. Genießen Sie ihn auf Eis oder mit einem Spritzer Mineralwasser als erfrischenden Aperitif.

G.IV.4 2020 Stairs n' Roses „Sweet Child“

URL: <https://www.vindor.de/2020-stairs-n-roses-sweet-child>

Strukturierte Daten:

Name: 2020 Stairs n' Roses „Sweet Child“,

Rebsorte: Riesling,

Produktkategorie: Weißwein,
Qualitätsstufe: Kabinett,
Cuvee: ,
Jahrgang: 2020,
Geschmacksrichtung: fruchtsüß,
Farbnuance: gelb,
Passende Speisen: Asiatische Speisen, fruchtige Desserts,

Generierte Beschreibung:

Lassen Sie sich von unserem 2020 Stairs n' Roses „Sweet Child“ verzaubern. Dieser fruchtsüße Riesling, liebevoll gekeltert im Herzen der Mosel, entfaltet ein Bouquet von sonnengereiften Äpfeln und Birnen, die sich mit einer leichten Zitrusnote und dem Duft von frischer Wiesenblüte vermischen. Seine Farbnuance, ein strahlendes Gelb, gleicht einem Sonnenuntergang im Spätsommer. Am Gaumen entfaltet er eine feine, elegante Süße, die von einer lebendigen Säure gekonnt balanciert wird. Ein wahrer Genuss, der Sie an warme Sommertage erinnert.

Ob zu scharfen asiatischen Speisen oder als süßer Kontrast zu fruchtigen Desserts, dieser Wein ist ein vielseitiger Begleiter. Doch auch solo getrunken, ist er ein wahrer Genuss. So wie eine süße Kindheitserinnerung, die immer wieder Freude bereitet.

G.IV.5 2019 Stairs n' Roses „Red Lion“

URL: <https://www.vindor.de/2019-stairs-n-roses-red-lion>

Strukturierte Daten:

Name: 2019 Stairs n' Roses „Red Lion“,
Rebsorte: Pinot Noir,
Produktkategorie: Rotwein,
Qualitätsstufe: QbA,
Cuvee: ,
Jahrgang: 2019,
Geschmacksrichtung: trocken,
Farbnuance: kaminrot,
Passende Speisen: hochwertige, vollmundige Speisen,

Generierte Beschreibung:

2019 Stairs n' Roses „Red Lion“ - Ein Rotwein mit Charakter Eine kräftige, rubinrote Farbe und ein angenehmer Duft nach Brombeeren und Erdbeeren machen Lust auf mehr. Der „Red Lion“ von Stairs n' Roses ist ein Rotwein mit Charakter, der sich hervorragend zu hochwertigen, vollmundigen Speisen wie gegrilltem Fleisch oder Wildbret eignet. Die Rebsorte Pinot Noir verleiht ihm seinen kräftigen Körper und seine fruchtige Süße, die durch die leichte Säure perfekt abgerundet wird. Der "Red Lion ist ein trockener Rotwein, der mit seinem weichen Abgang und seinen subtilen Tanninen für ein langes Finale sorgt. Der „Red Lion“ ist ein Wein, der mit seinem einzigartigen Geschmack und seinem lebhaften Aroma für Begeisterung sorgt. Probieren Sie ihn heute und genießen Sie einen Rotwein, der sich sehen lassen kann.

G.IV.6 Stairs n' Roses, „Next Generation“, QbA Mosel

URL: <https://www.vindor.de/stairs-n-roses-next-generation-qba-mosel>

Strukturierte Daten:

Name: Stairs n' Roses, „Next Generation“, QbA Mosel,

Rebsorte: Cabernet Blanc, Sauvignac,

Produktkategorie: Weißwein,

Qualitätsstufe: QbA,

Cuvee: Sauvignac, Cabernet blanc,

Jahrgang: 2021,

Geschmacksrichtung: trocken,

Farbnuance: helles Gelb mit grünlichen Reflexen,

Passende Speisen: würzige Speisen, asiatisches Essen,

Generierte Beschreibung:

Lassen Sie sich verzaubern von unserem Stairs n' Roses, „Next Generation“, QbA Mosel - ein Weißwein, der Ihre Geschmacksknospen auf eine sinnliche Reise schickt. Aus der malerischen Region Mosel stammend, verführt dieser Tropfen mit seiner hellen Gelbnuance, durchzogen von zarten grünlichen Reflexen. Er ist ein harmonisches Duett aus den Rebsorten Cabernet Blanc und Sauvignac, die eine Symphonie der Aromen spielen. Dieser Wein flüstert Geschichten von sonnenverwöhnten Weinbergen und der Leidenschaft seiner Winzer. In der Nase entfaltet er ein Bouquet von grünen Äpfeln, Zitrusfrüchten

und feinen Kräutern, das sich am Gaumen zu einer erfrischenden und zugleich komplexen Komposition entfaltet.

Der Stairs n' Roses ist ein wahrer Verführer, der mit seiner trockenen, aber fruchtigen Note gut zu würzigen Speisen und asiatischem Essen passt. Also nehmen Sie sich einen Moment Zeit, lassen Sie sich fallen und genießen Sie diesen Tropfen voller Leidenschaft und Hingabe. Ihr Gaumen wird es Ihnen danken.

G.IV.7 2018 Stairs n' Roses „Honeymoon“

URL: <https://www.vindor.de/2018-stairs-n-roses-honeymoon>

Strukturierte Daten:

Name: 2018 Stairs n' Roses „Honeymoon“,

Rebsorte: Riesling, Müller-Thurgau,

Produktkategorie: Weißwein,

Qualitätsstufe: Wein aus eingetrockneten Trauben,

Cuvee: ,

Jahrgang: 2018,

Geschmacksrichtung: edelsüß,

Farbnuance: tiefes gelb,

Passende Speisen: aperitiv, süße Desserts,

Generierte Beschreibung:

Der 2018er Stairs n' Roses „Honeymoon“ ist eine zauberhafte Symphonie aus Riesling und Müller-Thurgau, die aus dem malerischen Mosel-Tal stammt. Dieser Weißwein ist wie ein süßer Kuss im Mondschein, der Sie mit seinem tiefen Gelbton und edelsüßen Geschmack verzaubert. Beim Öffnen der Flasche entfaltet sich ein Bouquet aus sonnen gereiften Äpfeln, gepaart mit einer zarten Honignote und einem Hauch von exotischer Litschi. Am Gaumen entfaltet er eine beeindruckende Süße, die von einer lebhaften Säure perfekt ausbalanciert wird. Der Abgang ist lang und voller Eleganz. Der „Honeymoon“ ist der perfekte Begleiter für einen Aperitif oder süße Desserts. Er ist wie ein Flirt mit der Sonne, ein Tanz mit den Wolken, ein Lachen mit dem Wind. Ein Wein, der Sie auf eine Reise in das Herz der Mosel entführt, wo die Trauben in den Vormorgen von den Winzern Julia Molitor-Justen und Dr. Daniel Molitor mit Liebe und Leidenschaft gehegt und gepflegt werden.