

KI-gestützte Contenterstellung für kleine und mittelständische Verlage

MVP-Pragmatische Lösung mit 20.000€ Budget

Frank Eberle Consulting
Bürgermeister-Haidacher-Straße 10
D-82140 Olching

Mail: feberle@feberle.de
Phone: +49-176-36349575
<https://www.feberle.de>

Olching, November 2025
Version 1.0

Inhalt

Executive Summary	4
1. Ausgangssituation und Herausforderung.....	4
1.1 Ausgangslage bei KMU-Verlagen	4
1.2 Bisherige Barrieren für KI-Adoption	4
1.3 Grundannahme des MVP-Ansatzes	5
2. Strategischer Ansatz: MVP statt Komplettlösung	5
2.1 Scope-Beschränkung.....	5
2.2 Was IST im MVP enthalten	5
2.3 Technologie-Strategie: "Best of Breed" statt "Build Everything"	6
3. Pragmatische Lösung im Detail	6
3.1 Architektur - Vereinfacht	6
3.2 Workflow Schritt für Schritt	6
Phase 3.2.1: Datenerfassung & Vorfilterung	6
Phase 3.2.2: Relevanzfilter mit ML.....	7
Phase 3.2.3: KI-Textgenerierung	7
Phase 3.2.4: Redakteur Review & Entscheidung	8
Phase 3.2.5: CMS-Integration (pragmatisch)	8
4. Aufwandsreduzierung - konkrete Maßnahmen	9
4.1 Maßnahme 1: Fokus auf einen Titel	9
4.2 Maßnahme 2: Kein vollständiges Archiv-Training.....	9
4.3 Maßnahme 3: Vortrainierte Modelle nutzen (statt Custom-Training).....	9
4.4 Maßnahme 4: Halbautomatischer Prozess statt Vollautomation.....	9
4.5 Maßnahme 5: Verzicht auf Mehrsprachigkeit & Komplexfeatures.....	10
4.6 Zusammenfassung: Aufwandsreduktion.....	10
5. Projektplan - MVP Umsetzung.....	10
5.1 Phase 1: Initialisierung & Setup (2-3 Wochen)	10
5.2 Phase 2: Datenaufbereitung & Labeling (2-3 Wochen)	11
5.3 Phase 3: KI-Pipeline & Dashboard (3-4 Wochen)	12
5.4 Phase 4: Pilotbetrieb & Optimierung (4-6 Wochen)	13
5.5 Phase 5: Dokumentation & Deployment-Readiness (1-2 Wochen)	13
5.6 Gesamtprojektplan - Timeline.....	14
6. Budget und Kostenschätzung.....	15
6.1 Externe Kosten (Zielbudget: 20.000€)	15
6.2 Interne Personenkosten	15
6.3 Gesamtinvestition.....	15
6.4 Laufende Kosten ab Monat 5 (Regelbetrieb).....	16

7. KPIs und Erfolgsmessung	16
7.1 Operationale KPIs (täglich / wöchentlich gemessen)	16
7.2 Geschäfts-KPIs (monatlich gemessen)	16
7.3 Qualitäts-KPIs (wöchentlich / Stichprobe)	17
8. Risikoanalyse	17
8.1 Identifizierte Risiken & Mitigation	17
8.2 Risikobehandlung - Priorisierung	17
9. Implementierungs-Roadmap	18
9.1 MVP Phase (16-20 Wochen) - Nächste 4-5 Monate	18
9.2 Phase 2: Rollout auf weitere Titel (Monate 7-10)	18
9.3 Phase 3: Erweiterte Features (Monate 11-15)	18
9.4 Phase 4: Ganzheitliche Lösung (ab Monat 16)	18
10. Erfolgsfaktoren	19
10.1 Kritische Faktoren für MVP-Erfolg	19
11. Nächste Schritte	19
11.1 Entscheidungspunkt	19
11.2 Bei Ja - Sofortmaßnahmen (Woche 1)	20
11.3 Empfohlener Projekt-Ablauf	20
12. Kontakt & weitere Informationen	21
Anhang: Technische Details	21
A1. Technology Stack - MVP	21
A2. Datenfluss-Diagramm	22

Executive Summary

Die vollständige Implementierung einer KI-gestützten Contenterstellungslösung mit allen Funktionen (Mehrsprachigkeit, komplexe CMS-Integration, umfassende Automation) ist für kleine und mittelständische Verlage mit einem Budget von 20.000€ für externe Kosten unrealistisch.

Unsere Empfehlung: Ein **Minimal Viable Product (MVP)** mit fokussiertem Scope und pragmatischen Vereinfachungen ermöglicht dennoch einen wertvollen, erfolgreichen ersten Use Case:

- **Fokus:** Ein einzelner Titel, Pressemitteilungen als Datenquelle, einfache Automation ohne komplexe CMS-Integration
- **Technologie:** Moderne LLM-APIs (OpenAI, Claude) statt aufwendigem Custom-Training
- **Prozess:** Halbautomatisiert mit manueller Übernahme ins CMS statt vollständiger Automation
- **Timeline:** 16-20 Wochen statt 52 Wochen
- **Investition:** ca. 20.000€ externe Kosten + interne Ressourcen

Erwartete Ergebnisse nach MVP (Pilot):

- Zeitersparnis pro Artikel: 30-40% (von 35 Min auf 20 Min)
- Verwertungsquote: +50% (von 15% auf 22-25%)
- Zusätzliche nutzbare Kapazität: ca. 600 Stunden/Jahr
- Proof of Concept für Skalierung auf weitere Titel

Diese pragmatische Herangehensweise schafft einen messbaren ROI, generiert schnelle Erfolgsbeispiele und bildet die Grundlage für eine schrittweise Erweiterung zur ganzheitlichen Lösung.

1. Ausgangssituation und Herausforderung

1.1 Ausgangslage bei KMU-Verlagen

Typische Situation:

- 40-80 Pressemitteilungen pro Tag für einen Titel
- Manuelle Selektion, Sortierung und Umformulierung durch Redakteure
- 30-40 Minuten durchschnittlicher Aufwand pro verwerteter Pressemitteilung
- 10-20% Verwertungsquote (viel Material wird übersehen oder zu spät bearbeitet)
- Begrenzte IT-Ressourcen für komplexe Projekte
- Budget für externe Kosten typischerweise 15.000-25.000€

1.2 Bisherige Barrieren für KI-Adoption

1. **Finanzielle Hürde:** Geschätzte Kosten von 200.000€+ wirken abschreckend

2. **Komplexität:** Vollständige Archiv-Aufbereitung (5-10 Jahre) und Custom-Modellentwicklung erfordern viel Zeit und Expertise
3. **Technische Integration:** API-Entwicklung und CMS-Integration sind aufwendig
4. **Organisatorisch:** Große Pilotphasen mit mehreren Titeln erfordern mehr interne Ressourcen
5. **Fehlende Proof Points:** Schwer zu argumentieren, ohne erste Erfolgsbeispiele

1.3 Grundannahme des MVP-Ansatzes

"Ein kleiner, erfolgreicher Use Case schlägt einen großen, geplanten Use Case."

Besser: Ein funktionierendes Pilot-System auf einem Titel in 4-5 Monaten, das:

- Messbare Wertschöpfung bringt
- Akzeptanz bei Redakteuren aufbaut
- Als Sprungbrett für Skalierung dient

...als ein lange geplantes Großprojekt, das nie startet.

2. Strategischer Ansatz: MVP statt Komplettlösung

2.1 Scope-Beschränkung

Was ist NICHT im MVP enthalten:

Element	Warum nicht im MVP?	Zukünftig?
Mehrsprachigkeit	Komplexität, Trainingsaufwand	Phase 2
Automatische CMS-Integration	API-Entwicklung ist aufwendig	Phase 2
Komplettes Archiv-Training	5-10 Jahre Datenaufbereitung = zu viel Aufwand	Phase 2
Multiple Titel gleichzeitig	Zu viele Anforderungen parallel	Phase 2
Complex NER & Entity Linking	Nice-to-have, nicht essentiell	Phase 2
Echtzeitverarbeitung	Batch-Processing reicht	Phase 2

Table 1: Scope-Beschränkung MVP vs. Vollständige Lösung

2.2 Was IST im MVP enthalten

- ✓ **Ein Titel** (z.B. als Pilotierung)
- ✓ **Automatische Vorselektion** von Pressemitteilungen (Relevanzfilter)
- ✓ **KI-gestützte Textgenerierung** (Umformulierung, Ergänzung)
- ✓ **Halbautomatischer Workflow** (Redakteur akzeptiert/lehnt ab, befüllt CMS manuell)

- ✓ **Einfaches Dashboard** für Redakteure
- ✓ **Grundlegende Qualitätskontrolle** im Workflow
- ✓ **Basis-Monitoring** für KPIs

2.3 Technologie-Strategie: "Best of Breed" statt "Build Everything"

Anstatt:

- Eigene Modelle trainieren (teuer, lange)
- Komplexe Infrastruktur aufbauen

Nutzen wir:

- **OpenAI GPT-4 oder Claude 3** über APIs für Textgenerierung
- **HuggingFace transformers** für Klassifikation (einfach fine-tuned)
- **Einfache Vector DB** (Pinecone free tier) für Kontext-Retrieval
- **Cloud-basierte Workflows** (Azure Functions, AWS Lambda, oder Zapier)
- **Vorkonfigurierte Low-Code-Tools** statt Custom Development

Vorteil: 60-70% weniger Entwicklungsaufwand, schnellere Time-to-Value

3. Pragmatische Lösung im Detail

3.1 Architektur - Vereinfacht

Die Lösung folgt einem simplen, linearen Workflow:

1. **Datenerfassung:** Pressemitteilungen aus verschiedenen Quellen (E-Mail, RSS, APIs)
2. **Relevanzfilter:** ML-basierte Vorauswahl (nur relevante PM weitergeben)
3. **Textvorbereitung:** KI generiert Vorschlag (Umformulierung, Kürzung)
4. **Redakteur-Review:** Dashboard für Accept/Reject/Edit
5. **Export:** Artikel exportieren und manuell ins CMS übernehmen

3.2 Workflow Schritt für Schritt

Phase 3.2.1: Datenerfassung & Vorfilterung

Input: 40-80 Pressemitteilungen pro Tag

Quellen:

- E-Mail (Distributor)
- RSS-Feeds von Branchenquellen
- Direkte API-Integration (wenn vorhanden)

Automatische Vorverarbeitung:

- Strukturierung: HTML-Text in Textformat, Metadaten extrahieren (Sender, Datum, Links)

- Duplikatserkennung: Einfache String-Matching gegen letzte 30 Tage
- Spamfilter: Regel-basiert (bekannte Absender-Ausschlusslisten)

Output: Bereinigtes Daten-Set zur Relevanzprüfung

Phase 3.2.2: Relevanzfilter mit ML

Modell: Fine-getunetes HuggingFace BERT-Modell

Input: Pressemitteilungs-Text + Metadaten

Output: Relevanz-Score (0-1) mit Schwelle > 0.6

Training (Einmalig, ca. 1 Woche):

- Historische Daten: 500-1000 manuell gelabelte Pressemitteilungen
- "Relevant" = wurde in der Vergangenheit verwertete
- "Nicht relevant" = wurde ignoriert/abgelehnt
- 80/20 Train-Test Split
- Fine-Tuning auf 2-3 Epochen

Nutzen:

- Filtert ca. 40-50% der eingehenden PM als "nicht relevant" vor
- Reduziert manuelle Review auf ca. 20-30 PM/Tag
- Priorisiert wichtige Stories nach oben

Phase 3.2.3: KI-Textgenerierung

Basis: OpenAI GPT-4 oder Anthropic Claude 3 (API)

Prozess für jede Pressemitteilung mit Score > 0.6:

Prompt Engineering:

- Pressemitteilungs-Originaltext
- Titel/Kanal-Information
- Style Guide (kurz, prägnant, Fakten nicht übertrieben)
- Gewünschte Länge (200-400 Worte)
- Zielformat (Web, Newsletter, Heft)

Textgenerierung Anweisung:

"Du bist ein erfahrener Fachredakteur für [Industrie]. Reformuliere folgende Pressemitteilung zu einem ansprechenden Artikel. Struktur: Headline, Lead, Body. Keine erfundenen Daten, nur Fakten aus der Original-PM."

Output:

- Vorgeschlagene Headline
- Vorgeschlagener Artikel-Text
- Empfohlene Kategorie/Tags

- Veröffentlichungs-Timing

Kosten:

- Ca. 0,10-0,20€ pro generiertem Text (bei GPT-4, 40-80 Texte/Tag)
- Monatlich: ca. 2.000-2.500€ API-Kosten




Phase 3.2.4: Redakteur Review & Entscheidung

Interface: Einfaches Web-Dashboard

Redakteur sieht:

- Original-Pressemitteilung
- KI-Vorschlag (Headline + Text)
- Relevanz-Score und Kategorisierung
- Links zum Archiv (ähnliche frühere Artikel)

Redakteur wählt:

-  **Accept:** Text ist OK, wird zur Veröffentlichung empfohlen
-  **Edit:** Text braucht Anpassung, Redakteur überarbeitet
-  **Reject:** Nicht relevant, wird verworfen
- **Feedback:** Score war falsch / Text war schlecht

Zeitaufwand: 3-5 Min pro Review statt 35 Min Original (60% Reduktion)

Phase 3.2.5: CMS-Integration (pragmatisch)

Anstatt komplexe API-Entwicklung verwenden wir praktische Alternativen:

Option A - Export-basiert (empfohlen):

- Akzeptierte Artikel als CSV/JSON exportieren
- Redakteur/Assistent importiert manuell ins CMS
- Zeitaufwand: ca. 10 Min für 5-10 Artikel

Option B - Plugin-Integration:

- Einfaches WP-Plugin für Daten-Import
- One-Click Import mit vorausgefüllten Feldern
- Zeitaufwand: ca. 2 Min pro Artikel

Option C - API-Integration (Phase 2):

- REST-API zum CMS
 - Automatisches Befüllen bei Freigabe
 - Spätere Implementierung
-

4. Aufwandsreduzierung - konkrete Maßnahmen

4.1 Maßnahme 1: Fokus auf einen Titel

Aspekt	Vollständige Lösung	MVP
Anzahl Titel	3 Titel parallel	1 Titel als Pilot
Komplexität	Unterschiedliche Anforderungen	Fokussierte Anforderungen
Zeitaufwand Anforderungsanalyse	6 Wochen	2 Wochen

Einsparung: 4 Wochen

4.2 Maßnahme 2: Kein vollständiges Archiv-Training

Aspekt	Vollständige Lösung	MVP
Daten-Umfang	5-10 Jahre	1 Monat (~1.000 PM)
Datenaufbereitung	6-8 Wochen + Service	2-3 Wochen (intern)
Arbeitsaufwand	Hoch	Mittel

Einsparung: 4-5 Wochen

Begründung:

- Fine-Tuning auf vortrainierten Modellen braucht nicht gigantische Datenmengen
- 500-1.000 gelabelte Beispiele reichen für gutes Klassifikations-Modell
- Viel besser als 50.000 schlecht gelabelte Artikel

4.3 Maßnahme 3: Vortrainierte Modelle nutzen (statt Custom-Training)

Aspekt	Vollständige Lösung	MVP
Ansatz	Custom-Modelle trainieren	GPT-4 / Claude 3 APIs
Infrastruktur	Eigener ML-Stack, GPUs	Cloud-APIs (No-Ops)
Entwicklungszeit	10-12 Wochen	1-2 Wochen
Kosten	30-40k€	2.000-3.000€/Monat

Einsparung: 8-10 Wochen Entwicklungszeit

Trade-off: Höhere laufende API-Kosten, aber Time-to-Market viel schneller

4.4 Maßnahme 4: Halbautomatischer Prozess statt Vollautomation

Aspekt	Vollständige Lösung	MVP
Integration	Automatische CMS-API	Redakteur-Dashboard + Export

Entwicklung	Komplexe API-Integration	Einfaches Web-Interface
Entwicklungszeit	8-10 Wochen	2-3 Wochen

Einsparung: 6-8 Wochen Entwicklung

Realität: Redakteur braucht nur 10 Minuten für manuellen Import - vollkommen akzeptabel für MVP

4.5 Maßnahme 5: Verzicht auf Mehrsprachigkeit & Komplexfeatures

Feature	Status	Grund
Mehrsprachigkeit	Phase 2	Komplexität, Trainingsaufwand
Entity Linking	Phase 2	Nicht essentiell
A/B-Testing	Phase 2	Basis-Format reicht
Advanced NLP	Phase 2	Fokus auf Kern-Funktionalität

Einsparung: 2-4 Wochen Entwicklung

4.6 Zusammenfassung: Aufwandsreduktion

Phase	Vollständig	MVP	Einsparung
Anforderungen	6 Wo	2 Wo	4 Wo
Datenaufbereitung	6-8 Wo	2-3 Wo	4-5 Wo
KI-Entwicklung	10-12 Wo	1-2 Wo	8-10 Wo
Systemintegration	8-10 Wo	2-3 Wo	6-8 Wo
Pilotbetrieb	8-10 Wo	4-6 Wo	2-4 Wo
TOTAL	50-56 Wo	16-20 Wo	30-36 Wo

Table 2: Zeitliche Aufwandsreduktion: 60-70%

5. Projektplan - MVP Umsetzung

5.1 Phase 1: Initialisierung & Setup (2-3 Wochen)

Ziele:

- Anforderungen klar definiert
- Team zusammengestellt
- Technische Grundlagen etabliert
- Projekt-Kickoff erfolgreich

Aktivitäten:

Aktivität	Dauer	Verantwortung
Kick-off & Stakeholder-Alignment	2 Tage	Projektleiter, GF, Chefredakteur
Anforderungsdokumentation	3 Tage	Projektleiter, Chefredakteur
Technische Architektur definieren	2 Tage	Technik-Lead
Cloud-Accounts einrichten	1 Tag	DevOps/IT
Tool-Auswahl (GPT-4 vs Claude)	1 Tag	Data Scientist
Datenzugriff & Archiv-Analyse	3 Tage	Data Engineer
Rollen & Kommunikations-Plan	1 Tag	Change Manager
Summe	13 PT	

Table 3: Phase 1: Aktivitäten und Aufwand

Deliverables:

- ☒ Anforderungsdokument (5-10 Seiten)
- ☒ Technisches MVP-Konzept
- ☒ Tool-Auswahl und Lizenzen
- ☒ Cloud-Accounts aktiv
- ☒ Detaillierter Projektplan
- ☒ Risikoregister initialisiert

5.2 Phase 2: Datenaufbereitung & Labeling (2-3 Wochen)

Ziele:

- Trainingsdaten bereit
- Klassifikations-Modell trainiert
- Archiv-Qualität validiert

Aktivitäten:

Aktivität	Dauer	Verantwortung
Archiv-Dump exportieren	2 Tage	Data Engineer
Datenbereinigung	2 Tage	Data Engineer
Struktur normalisieren	1 Tag	Data Engineer
Labeling-Interface erstellen	1 Tag	Frontend Dev
Redakteure labeln	5 Tage	Redakteure (intern)
Klassifikations-Modell trainieren	2 Tage	Data Scientist

Performance evaluieren	1 Tag	Data Scientist
Summe	15 PT	

Table 4: Phase 2: Aktivitäten und Aufwand

Redakteur-Engagement: ca. 5 Personentage (verteilt über 2 Wochen à 30 Min/Tag)

Deliverables:

- ☒ ~800 gelabelte Pressemitteilungen
- ☒ Trainiertes Klassifikations-Modell
- ☒ Performance-Bericht (Accuracy >85%)
- ☒ Test-Datensatz

5.3 Phase 3: KI-Pipeline & Dashboard (3-4 Wochen)

Ziele:

- Automatische Relevanzprüfung läuft
- Textgenerierung via APIs funktioniert
- Redakteur-Dashboard live
- Feedback-Loop etabliert

Aktivitäten:

Aktivität	Dauer	Verantwortung
Datenbank / File-Storage setup	2 Tage	DevOps, Backend
API-Integration (GPT-4 / Claude)	3 Tage	Backend Dev
Klassifikations-Modell deployieren	2 Tage	DevOps
Workflow-Orchestrierung	3 Tage	Backend Dev
Dashboard entwickeln	5 Tage	Frontend Dev
Testing & Fehlerbehebung	2 Tage	QA, Backend
Dokumentation & Runbook	1 Tag	Backend Dev
Summe	18 PT	

Table 5: Phase 3: Aktivitäten und Aufwand

Deliverables:

- ☒ Laufender Workflow: PM → Filter → TextGen → Dashboard
- ☒ Redakteur-Dashboard (Web-basiert)
- ☒ Tägliche Verarbeitung von 40-80 PM

- ☒ API-Dokumentation
- ☒ Betriebs-Runbook

5.4 Phase 4: Pilotbetrieb & Optimierung (4-6 Wochen)

Ziele:

- Live-Betrieb mit echten Redakteuren
- Feedback sammeln & Modelle optimieren
- KPIs tracked und gemessen
- Erfolgsbeispiele dokumentiert

Aktivitäten:

Aktivität	Dauer	Verantwortung
Schulung der Redakteure	1 Tag	Projektleiter, Technik
Go-Live auf Produktionssystem	1 Tag	DevOps
Pilot-Betrieb	20 Tage	Alle
Prompt-Tuning	3 Tage	Data Scientist
Modell-Nachtraining	2 Tage	Data Scientist
Relevanz-Schwelle anpassen	1 Tag	Data Scientist
Dashboard-UX verbessern	2 Tage	Frontend Dev
KPI-Tracking etablieren	2 Tage	Analytics
Best Practices dokumentieren	3 Tage	Projektleiter
Summe	35 PT	

Table 6: Phase 4: Aktivitäten und Aufwand

Pilot-Dauer: 4-6 Wochen mit echtem Redakteur-Feedback

Output:

- ☒ Optimiertes System basierend auf echtem Feedback
- ☒ KPI-Baseline gemessen
- ☒ Best Practices dokumentiert
- ☒ Hochwertige Erfolgsbeispiele

5.5 Phase 5: Dokumentation & Deployment-Readiness (1-2 Wochen)

Ziele:

- System produktionsreif
- Dokumentation vollständig

- Support-Prozesse etabliert
- Go/No-Go Entscheidung

Aktivitäten:

Aktivität	Dauer	Verantwortung
Sicherheits-Audit	2 Tage	Security, Legal
Compliance-Dokumentation (DSGVO, AI Act)	2 Tage	Legal, Datenschutz
Benutzer-Dokumentation	2 Tage	Projektleiter
Schulungsmaterialien	2 Tage	Training
Support-Prozesse definieren	1 Tag	Operations
Disaster Recovery testen	1 Tag	DevOps
Final Sign-Off	1 Tag	Steuerkreis
Summe	11 PT	

Table 7: Phase 5: Aktivitäten und Aufwand

Deliverables:

- ☒ System-Dokumentation
- ☒ Benutzer-Handbuch
- ☒ Schulungsmaterialien
- ☒ Support-Runbook
- ☒ Compliance-Report
- ☒ Go-Live Freigabe

5.6 Gesamtprojektplan - Timeline

Projekt-Duration: 16-20 Wochen (ca. 4-5 Monate)

Woche	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Status
1-3	■					Initialisierung
3-5		■				Datenaufbereitung
5-9			■			KI-Pipeline
9-15				■		Pilot-Betrieb
15-17					■	Deployment
20						Go-Live

6. Budget und Kostenschätzung

6.1 Externe Kosten (Zielbudget: 20.000€)


Position	EUR	Begründung
Cloud-Infrastruktur	2.000	Azure/AWS/GCP minimal (4 Mo à 500€)
OpenAI GPT-4 / Claude API	3.500	ca. 80 Texte/Tag × 0,15€ × 120 Tage
Vector DB (Pinecone o.ä.)	500	Archiv-Snippets, minimal tier
Development Tools	1.500	GitHub, Jira, Collaboration
Software-Lizenzen	1.000	IDE, Analytics Tools
External Consulting	5.000	Specialized NLP/ML Support (5 PT)
Annotation/Labeling Service	3.000	Falls interne Kapazität nicht reicht
Schulungen	2.000	Online-Kurse, externe Trainer
Contingency (10%)	1.000	Unvorhergesehene Kosten
TOTAL EXTERNE KOSTEN	20.000€	 max. 20k Budget

Table 8: Externe Kosten - Detaillierung

6.2 Interne Personenkosten

Rolle	PT	Ø-Satz (€)	Gesamt (€)
Projektleiter	40	200	8.000
Chefredakteur	30	200	6.000
Redakteure (Labeling, Pilot)	20	180	3.600
IT/DevOps (intern)	20	180	3600
TOTAL INTERN	110 PT		31.200€

Table 9: Interne Personenkosten - Schätzung

6.3 Gesamtinvestition

Komponente	EUR
Externe Kosten	20.000
Interne Personenkosten	31.200
TOTAL INVESTITION JAHR 1	~51.200

Finanzierungsoptionen:

- Externe Kosten: 20.000€ aus dediziertem Budget
- Interne Kosten: aus bestehendem Personalbudget (projektgebundene Zeit)

6.4 Laufende Kosten ab Monat 5 (Regelbetrieb)

Position	EUR/Monat
Cloud-Infrastruktur	400
API-Kosten (GPT-4, stabilisiert)	800
Vector DB	50
Support & Wartung (intern PT)	1.000
Monatliche Kosten Regelbetrieb	2.250
Jährliche Kosten ab Jahr 2	27.000

7. KPIs und Erfolgsmessung

7.1 Operationale KPIs (täglich / wöchentlich gemessen)

KPI	Baseline	Target nach Pilot	Messmethode
Verarbeitete PM pro Tag	40-80	80-100% Durchsatz	Dashboard
Relevant vorgefiltert	0%	40-50%	ML-Ausgabe
Zeitaufwand pro PM	35 Min	12-15 Min	Zeitmessung
Verwertungsquote	10-20%	22-30% (+50%)	Anteil publiziert
KI-Akzeptanzquote	0%	>70%	Dashboard-Logs
Fehlerquote	n/a	<5%	Stichprobenprüfung

Table 10: Operationale KPIs - Messung und Targets

7.2 Geschäfts-KPIs (monatlich gemessen)

KPI	Erwarteter Wert	Nutzen
Zusätzliche Content-Stunden/Monat	+40-60 Std	Fokus auf Unique Content
Redaktionskosten pro Artikel	-30%	Effizienzgewinn
Time-to-Publish	-40%	First-Publisher-Vorteil
Mitarbeiterzufriedenheit	+20-30%	Weniger repetitive Arbeit

Table 11: Geschäfts-KPIs - Erfolgsmessung

7.3 Qualitäts-KPIs (wöchentlich / Stichprobe)

KPI	Target
Relevanz-Accuracy (echtes Redakteur-Feedback)	>80%
Faktische Korrektheit der Texte	>95%
Lesbarkeit/Journalistische Qualität	4.0+/5.0
Duplicate Content Detection	<1%

Table 12: Qualitäts-KPIs - Qualitätsmessung

8. Risikoanalyse

8.1 Identifizierte Risiken & Mitigation

Risiko	W'keit	Impact	Mitigation
KI-Modell: False Positives	Mittel	Hoch	Schwelle anpassen, Nachtraining, Redakteur-Kontrolle
Redakteure lehnen ab	Mittel	Hoch	Change Mgmt, Champions, Quick Wins
API-Kosten eskalieren	Gering	Mittel	Weekly Monitoring, Fallback auf local Models
Tech. Probleme Go-Live	Gering	Hoch	Testing, Fallback (manuell), Support-Backup
DSGVO/Compliance-Issues	Gering	Sehr hoch	DSFA früh, Legal involved, Datenschutz
Archivdaten unzureichend	Gering	Mittel	Synthetische Daten, Extended Labeling, historische Erweiterung
Feedback unpraktisch	Mittel	Mittel	Feature-Priorisierung, Phase-2-Roadmap
Redakteur-Burnout	Gering	Mittel	Ausreichend Ressourcen, Rotation

Table 13: Risikoanalyse - Identifizierung und Behandlung

8.2 Risikobehandlung - Priorisierung

Höchste Priorität:

1. **Redakteur-Akzeptanz:** Change Management intensiv, Early Adopter als Champions, Quick Wins demonstrieren
2. **Compliance:** Datenschutz-Folgenabschätzung (DSFA) in Phase 1, Legal von Anfang an involved

3. **Technische Stabilität:** Ausgiebiges Testing vor Go-Live, Fallback-Prozess, Support-Backup

Mittlere Priorität:

- API-Kostenmonitoring wöchentlich
 - Feature-Priorisierung für Feedback
 - Modell-Nachtraining bei False Positives
-

9. Implementierungs-Roadmap

9.1 MVP Phase (16-20 Wochen) - Nächste 4-5 Monate

Fokus:

- ☒ 1 Titel mit automatischer PM-Verarbeitung
- ☒ Messbare Zeiteinsparung und höhere Verwertungsquote
- ☒ Redakteur-Akzeptanz etabliert
- ☒ Proof of Concept für Skalierung

Erwartete Ergebnisse:

- 30-40% Zeiteinsparung pro Artikel
- +50-100% Verwertungsquote
- +40-60 verfügbare Redaktionsstunden/Monat
- Positive Redakteur-Rückmeldung

9.2 Phase 2: Rollout auf weitere Titel (Monate 7-10)

Nächste Schritte nach MVP-Erfolg:

- a. und 3. Titel mit gleicher Lösung
- Titel-spezifisches Fine-Tuning
 - Optimierte Workflows basierend auf Pilot-Learnings
 - Vollständige Dokumentation & Schulung

9.3 Phase 3: Erweiterte Features (Monate 11-15)

Mögliche Erweiterungen:

- Mehrsprachigkeit (Englisch, Französisch)
- Automatische CMS-Integration (API-basiert)
- Advanced Analytics (Topic Trends, Performance)
- Content-Vorschau und Headline-Testing

9.4 Phase 4: Ganzheitliche Lösung (ab Monat 16)

Vollständige Lösung:

- Alle bisherigen Features
 - Multi-Titel-Orchestrierung
 - Internationale Expansion
 - Strategische KI-Funktionen (Trend-Erkennung, Topic-Modellierung)
-

10. Erfolgsfaktoren

10.1 Kritische Faktoren für MVP-Erfolg

1. **Realistische Erwartungen**

- MVP = funktionierendes System mit echtem Nutzen, KEINE Wunderwaffe
- Klare Kommunikation: "Erster Schritt zu größerer Lösung"
- Transparent über Limitierungen sprechen

2. **Redakteur-Engagement**

- Frühzeitige Einbindung, nicht Überraschung
- Angst nehmen: "Mehr Freiraum für kreative Arbeit"
- Early Adopters als interne Champions nutzen

3. **Qualität vor Quantität**

- Lieber 50% weniger PM, aber hohe Qualität
- Redakteur-Review als Qualitätsgate
- Weniger ist mehr für MVP

4. **Daten-Qualität**

- 800 gut gelabelte Beispiele sind besser als 10.000 schlecht gelabelte
- Investment in Labeling zahlt sich aus
- Garbage in = Garbage out

5. **Iterative Verbesserung**

- Nicht perfekt, sondern besser werden
- Wöchentliche Review & Anpassung
- Redakteur-Feedback einarbeiten

6. **Kommunikation & Change Management**

- Regelmäßige Updates an Stakeholder
 - Success Stories teilen (intern & extern)
 - Transparent über Probleme sprechen
-

11. Nächste Schritte

11.1 Entscheidungspunkt

Geschäftsführung/Projektleiter entscheiden:

- ☐ **Ja, MVP-Ansatz:** Projektfreigabe erteilen, Budget genehmigen, Team bilden
- ☐ **Nein:** Bitte Begründung notieren, alternative Lösung besprechen
- ☐ **Vielleicht:** Pilot-Gespräche mit Redaktion durchführen

11.2 Bei Ja - Sofortmaßnahmen (Woche 1)

1. Management-Decision & Budget-Freigabe

- Geschäftsführung gibt Projekt frei
- 20.000€ Budget reserviert

2. Projektteam bilden

- Projektleiter: [Name, verfügbar ab Woche 1]
- Technisch Lead: [Name, verfügbar ab Woche 1]
- Data Scientist: [Name oder externe/Teil-Zeit]
- Redakteur (Pilot): [Name aus betreffendem Titel]

3. Kick-Off Workshop (Woche 1, 2 Tage)

- Stakeholder: GF, Chefredakteur, IT-Leitung
- Agenda: Anforderungen klären, Erwartungen setzen, Risiken diskutieren
- Output: Anforderungsdokument & Projektplan signiert

4. Lizenzen & Accounts beschaffen (Woche 1-2)

- OpenAI GPT-4 oder Claude-Accounts
- Azure/AWS/GCP Project setup
- Development-Umgebung vorbereiten

5. Datenzugriff & Archiv-Dump (Woche 1-2)

- Zugriff auf PM-Archiv
- Export der letzten 6 Wochen (ca. 2.000-3.000 PM)
- Datenbank-Setup

11.3 Empfohlener Projekt-Ablauf

Zeitpunkt	Aktivität	Owner
Diese Woche	Dokument präsentieren, Feedback sammeln	GF, PL
Woche 1	Management-Decision treffen	GF
Woche 2	Kick-Off Workshop durchführen	PL
Woche 3-20	MVP-Umsetzung (5 Phasen)	Projektteam
Woche 20	First Live-Resultat, KPIs messbar	Projektteam
Monat 5	Rollout-Entscheidung für weitere Titel	GF, PL

12. Kontakt & weitere Informationen

Projektverantwortlicher:

Frank Eberle
AI Consultant & DEKRA AI Officer (in Zertifizierung)

Kontakt:

E-Mail: feberle@feberle.de
Telefon: +49-176-36349575
Website: <https://www.feberle.de>

Adresse:

Frank Eberle Consulting
Bürgermeister-Haidacher-Straße 10
D-82140 Olching

Stand des Dokuments:

Version 1.0 | November 2025

Anhang: Technische Details

A1. Technology Stack - MVP

Backend:

- Python 3.10+ (Primär-Sprache)
- FastAPI oder Flask für REST APIs
- PostgreSQL für relationale Daten, SQLite als Alternative
- Redis für Caching und Session-Management

AI/ML:

- HuggingFace Transformers (BERT-small für Klassifikation, Fine-Tuning)
- OpenAI GPT-4 API oder Anthropic Claude 3 (Textgenerierung)
- LangChain für LLM-Integration und Prompt-Verwaltung
- scikit-learn für ML-Utilities

Frontend:

- React oder Vue.js (einfaches Dashboard)
- D3.js für KPI-Visualisierung
- Bootstrap/Tailwind CSS für Styling

Infrastruktur:

- Azure Container Instances oder AWS Lambda (serverless)
- Azure Blob Storage / AWS S3 für Dokumente

- GitHub für Code-Repository und Versionskontrolle

Optionale No-Code Alternativen:

- Zapier für Workflow-Automatisierung
- [Make.com](#) für Integration
- n8n für selbst-gehostete Automation

A2. Datenfluss-Diagramm

