Abschlussbericht
Open Source Chatbot EMF

Proof of Concept für Projekt IMPLEMENT

Luca Rolshoven, Lara Burkhalter, Matthias Stürmer

V1.0, 14.01.2025

Berner Fachhochschule
Wirtschaft
Institut Public Sector Transformation

Zusammenfassung und Vorteile dieses PoCs

Der vorliegende Bericht erläutert im Detail, wie die Chatbot-Lösung basierend auf Open Source Software und frei verfügbaren KI-Modelle realisiert wurde. Damit wird Transparenz geschaffen, welche technologischen Elemente (Software und KI-Modelle) im Einsatz stehen und wie die Architektur des Proof-of-Concept (PoC) konkret aussieht. Auch werden die verschiedenen Funktionalitäten des Chatbots wie Folgefragen und Datenquellen erläutert. Und zuletzt wird auch das Vorgehen der Evaluation des vorliegenden PoC beschrieben und auf die Effektivität der Suchergebnisse sowie die Qualität der generierten Antworten eingegangen. Die manuelle Beurteilung der Antwortqualität wurde durch eine Anwältin vorgenommen.

Im Wesentlichen wurde für diesen Chatbot ein selbständig entwickelter «Retrieval-Augmented Generation» (RAG) Dienst realisiert.[1] Damit werden die Stärken von modernen «Large Language Models» (LLM) genutzt und gleichzeitig die Gefahr von falschen Aussagen (Halluzinationen) verringert. Auch können grosse Mengen an Textdaten (bspw. in PDFs) verarbeitet und für die Beantwortung von Fragestellungen verwendet werden.

Durch den Open Source Ansatz bietet dieser PoC folgende Vorteile:

1. **Sicherstellung des Datenschutzes:** Sämtliche Daten können auf eigenen Servern der Stadt Bern verarbeitet und gespeichert werden, sodass künftig auch sensible Dokumente für die Beantwortung von Anfragen verwendet werden können. Zwar wurden die verwendeten Sprachmodelle (siehe Abschnitt 4.2) mit öffentlichen Daten trainiert, aber sie können für den Betrieb in ein abgesichertes Rechenzentrum der Stadt Bern übernommen werden und gewährleisten so den autonomen, vollständig Datenschutz-konformen Betrieb des Chatbots. Für das Vektorisieren der Daten und den regulären Betrieb verlassen damit keine Daten die Server der Stadt Bern.

2. **Transparenz und Nachvollziehbarkeit:** Durch die selbständige Entwicklung basierend auf «State-of-the-Art» Open Source Software Frameworks und Open KI-Modellen sowie «Good Practices» aus der Forschung sind die Architektur und die Funktionsweise dieses Chatbots genau nachvollziehbar. Und durch die Verfügbarkeit von automatisierten Tests (siehe Abschnitt 7.2) kann die Qualität der Ergebnisse laufend transparent überprüft werden.

3. **Verbesserungen laufend möglich:** Wenn in Zukunft weitere fachliche Dokumente berücksichtigt werden sollen (bspw. neue Regulierungen und Vorgaben) oder neue Versionen der KI-Modelle verfügbar sind, können diese Verbesserungen am Chatbot laufend realisiert werden. Zusätzliche Dateien können einfach vektorisiert und so für die künftige Abfrage integriert werden. Und neue Sprachmodelle können rasch in die bestehende Chatbot-Lösung aufgenommen werden um bessere Suchresultate zu bieten. Auch erlauben es zahlreiche Einstellungsmöglichkeiten (sogenannte «Hyperparameter») bei Bedarf eine Feinjustierung der Ergebnisse.

4. **Niedrige Betriebskosten:** Die Betriebskosten des Chatbots sind niedrig und einfach kalkulierbar. Entweder kann über die Anschaffung der entsprechenden Informatik-Infrastruktur (Server, Grafikkarten) eine einmalige Investition getätigt werden, sodass der Chatbot vollständig autonom im eigenen Rechenzentrum betrieben werden kann. Oder über bestehende KI-Services (siehe Abschnitt 5.3) können günstig und einfach die Daten extern verarbeitet werden, ohne dass eine wesentliche Herstellerabhängigkeit eingegangen wird.

[1] Siehe Erläuterungen auf http://bfh.ch/ipst/public-sector-ai

Inhaltsverzeichnis

1 Einführung

Im Rahmen des Projekts IMPLEMENT hat ein Team des Instituts Public Sector Transformation (IPST) des Departements BFH Wirtschaft die Implementierung eines Proof of Concepts (PoCs) für einen Open Source Chatbot übernommen. In diesem Kontext ist unter «Open Source» die Verwendung von frei zugänglichen Sprachmodellen zu verstehen, anstelle von proprietären Sprachmodellen wie beispielsweise die GPT-Modelle der Firma OpenAI.

Dieser Bericht beschreibt das Endresultat dieses PoCs und spezifiziert, was für eine Architektur verwendet wurde, welche Features der Chatbot beinhaltet und wie seitens BFH der Chatbot evaluiert wurde.

2 Funktionsweise

Die Bedienung des Chatbots ist möglichst einfach gehalten. Als erstes müssen sich Benutzer:innen einloggen. Sobald die Benutzer:innen eingeloggt sind, bleiben sie für einige Tage eingeloggt. Der Chatbot begrüsst die Benutzer:innen mit der Nachricht «Ich beantworte gerne Ihre Fragen! Worum geht es?». Ganz unten ist eine Nachrichtenleiste ersichtlich, in der Benutzer:innen Fragen stellen können. Mit einem Klick auf das Symbol rechts von der Leiste oder durch das Betätigen der Enter-Taste wird die Nachricht an den Chatbot gesendet. Dieser löst eine Suche aus und verwendet bis zu 7 Textpassagen während der Generierung der Antwort. Diese Textpassagen sind diejenigen, die von dem Chatbot als «am relevantesten» eingestuft wurden. Es kann dennoch sein, dass einige dieser sieben Textpassagen irrelevant sind – in diesem Fall zitiert der Chatbot die Quellen in der Regel nicht. Dies ist dann unter dem Dropdown «Quellen» ersichtlich, siehe Abschnitt 6.3.

Benutzer:innen können Folgefragen stellen, wenn etwas noch nicht ganz klar ist. Dies ist so gestaltet, dass es in Form einer normalen Konversation mit dem Chatbot erfolgt. Wie diese Funktion genau funktioniert, ist in Abschnitt 6.2 beschrieben.

3 Verwendete Dokumente

Separat zu diesem Abschlussbericht wird eine Liste mit den berücksichtigten Dokumenten geliefert. Grundsätzlich werden diejenigen Dateien verwendet, die auf dem SharePoint der Stadt Bern zur Verfügung gestellt wurden. Allerdings haben wir einige Dokumente, die online (extern) verfügbar waren, noch einmal neu heruntergeladen, da einige Dokumente nur Screenshots beinhalteten oder da gewisse Informationen fehlten (z.B. einige Artikel zwischen den einzelnen Dateien «Bundesgesetz XY.pdf»). Ausserdem haben wir einige Dokumente aus dem Internet heruntergeladen, die nicht bereits in den Dateien auf dem SharePoint zur Verfügung gestellt waren, jedoch durchaus von Interesse sein könnten. Zusätzlich haben wir die Dateinamen geändert, so dass diese im User Interface leichter verständlich sind. Die berücksichtigten Dokumente und die Änderungen, die an den externen Dokumenten zur Verbesserung des Chatbots vorgenommen wurden, sind in den beiden Arbeitsblättern der beigelegten Excel-Datei «Übersicht Dokumente.xlsx» ersichtlich.

Insgesamt umfasst die Wissensdatenbank des Chatbots 296 Dokumente, die 3932 Seiten bzw. 1'289'103 Wörter umfassen. Die wurden gemäss nachfolgenden Erläuterungen in 3522 Chunks (Textpassagen) unterteilt.

4 Verwendete Technologien

4.1 Software

Die Software, welche für die Implementierung des Chatbots verwendet wurde, ist ebenfalls unter einer Open Source Lizenz veröffentlicht. Für das Frontend, welches die Interaktion mit dem Chatbot ermöglicht, wurde das User Interface (UI) Framework Streamlit[2] verwendet. Das Backend, welches die Logik abbildet, wurde mit Hilfe des KI-Frameworks Haystack[3] implementiert.

4.2 Sprachmodelle

Der Chatbot verwendet drei verschiedene Sprachmodelle: Das erste Sprachmodell wandelt die Frage der Benutzer:innen in ein Embedding um. Ein Embedding ist ein Vektor, der semantische Informationen über den Text abspeichert und somit die Bedeutung des Texts repräsentiert. Das zweite Modell dient dazu, die gefundenen Textpassagen der beiden Suchmethoden (lexikografisch und semantisch) zu kombinieren und neu zu bewerten, was in einer einheitlichen und verfeinerten Rangliste der Textpassagen resultiert. Das dritte Sprachmodell ist dasjenige, welches den Benutzer:innen eine ausformulierte Antwort zurückliefert. Im Folgenden werden wir diese Modelle als **Embedder**, **Reranker** und **Large Language Model (LLM)** bezeichnen.

Für den Embedder haben wir ursprünglich das Modell gte-multilingual-base[4] von Alibaba-NLP verwendet, da dies in bisherigen Projekten zu guten Resultaten geführt hat. Im Dezember 2024 wurde allerdings ein neues Embedding-Modell namens Arctic Embed 2.0[5] der Firma Snowflake veröffentlicht, das laut den Experimenten der Autoren[6] bessere Resultate liefert. Aus diesem Grund haben wir uns für dieses Modell als Embedder entschieden. Als Reranker verwenden wir das Modell BAAI/bge-reranker-v2-m3[7]. Für das LLM, das die Antwort generiert, verwenden wir das Modell Llama-3.3-70B-Instruct[8], das in unseren Tests am besten abschnitt. Alle Modelle können relativ einfach ausgewechselt werden, sobald eine neue Version erscheint oder alternative Modelle bessere Ergebnisse aufzeigen.

5 Architektur

5.1 Ablauf einer Chatbot-Anfrage

Eine vereinfachte Darstellung der Architektur des Chatbots ist in Abbildung 1 zu sehen. Benutzer:innen interagieren mit dem Chatbot via dem Streamlit UI, in dem sie dem Chatbot eine Frage stellen. Diese Frage wird an das Haystack-Backend gesendet, das Textpassagen aus den gespeicherten Dokumenten sucht, die zu der Frage passen. Dies geschieht einerseits mittels einer lexikografischen (wortgetreuen) Suche, die anhand von einzelnen Wörtern und deren Informationsgehalt ähnliche Textpassagen findet. Andererseits wird eine semantische Suche angewendet, die auf Ähnlichkeiten von Vektor-Embeddings basiert. Die so identifizierten Textpassagen werden in einem nächsten Schritt noch einmal von einem spezialisierten Sprachmodell, dem Reranker, neu bewertet. Die sieben Textpassagen mit der höchsten Relevanz werden zusammen mit der Frage an ein LLM weitergegeben, das schliesslich die Antwort generiert. Diese Antwort wird dann an das UI weitergeleitet und den Benutzer:innen angezeigt.

[2] Siehe https://streamlit.io
[3] Siehe https://haystack.deepset.ai
[4] Siehe https://huggingface.co/Alibaba-NLP/gte-multilingual-base
[5] Siehe diesen Blogbeitrag von Snowflake
[6] Siehe https://arxiv.org/abs/2412.04506v2, insbesondere Tabelle 1
[7] Siehe https://huggingface.co/BAAI/bge-reranker-v2-m3
[8] Siehe https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

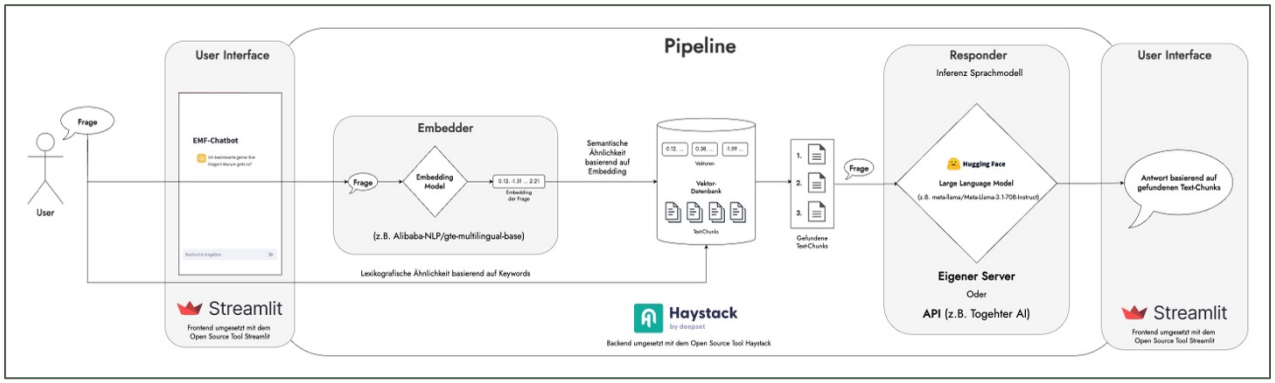

Abbildung 1: Grundsätzliche Architektur des OSS Chatbots im Rahmen des PoCs. Benutzer:innen interagieren mit dem Chatbot via dem Streamlit User-Interface. Die Fragen werden an das Haystack-Backend gesendet, welches diese mittels einem LLM beantwortet.

5.2 Möglichkeit für Follow-up Fragen

Sollten die Benutzer:innen eine weitere Frage zu demselben Thema haben, so können im nächsten Schritt Follow-Up Fragen gestellt werden. Der Chatbot merkt selbstständig, ob es sich bei der nächsten Frage um eine Follow-Up Frage handelt oder ob es sich um eine neue Frage handelt. Im Falle einer Follow-Up-Frage schreibt der Chatbot die Frage so um, dass diese ohne weiteren Kontext, also ohne die vorherige Konversation zu lesen, verstanden werden kann. Mit dieser neuen Frage wird dann der gleiche Suchprozess angestossen wie zuvor, und es wird erneut eine Antwort generiert. Die umformulierte Frage kann angezeigt werden, wenn die Zwischenschritte aufgerufen werden. Dieses Feature ist im folgenden Abschnitt «Funktionsweise» erläutert.

5.3 Technische Implementierung

Eine technischere Illustration des Prozesses ist in Abbildung 2 ersichtlich. Alle Modelle in dieser Architektur sind frei verfügbar als Open Models und können entweder in der Cloud oder On-Premise auf einem eigenen Server ausgeführt werden. Eine weitere Variante ist, das KI-Modell über eine Programmierschnittstelle (Application Programming Interface, API) wie beispielsweise Together[9] zu nutzen.

[9] Siehe https://www.together.ai/

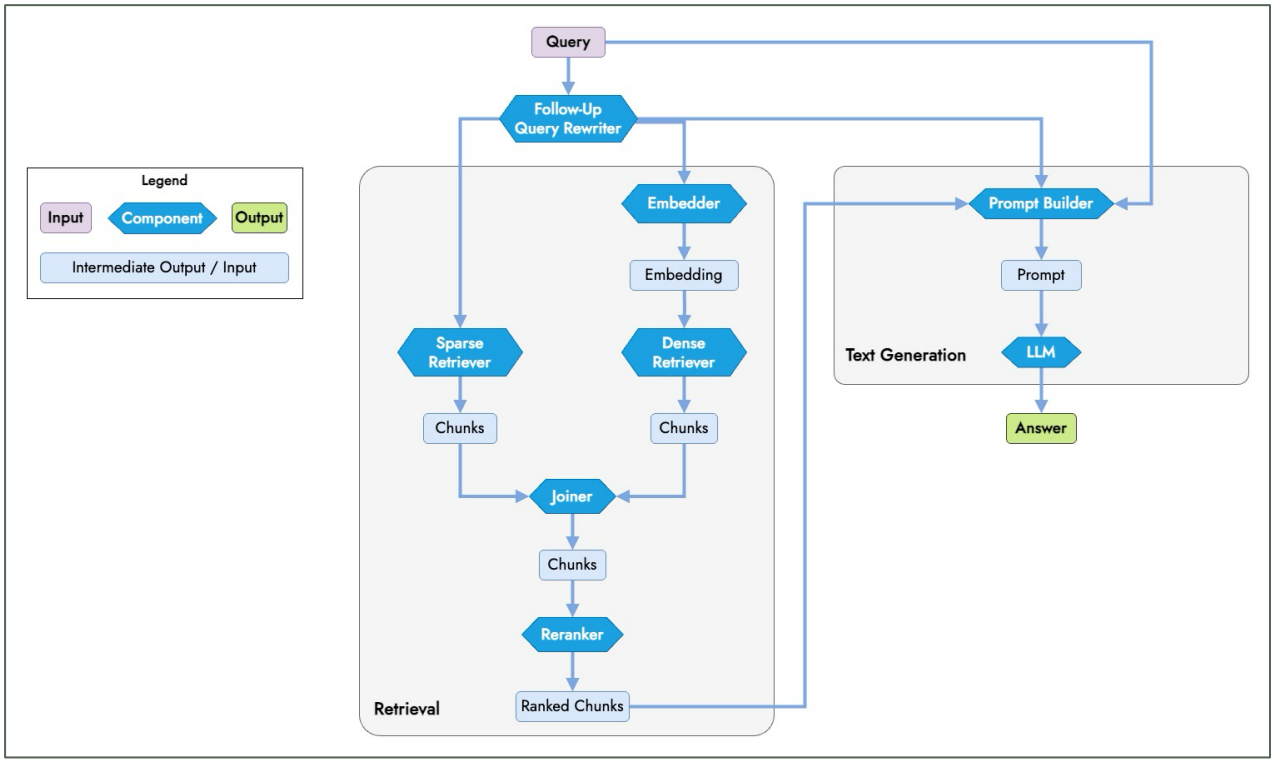

Abbildung 2: Ablauf der Verarbeitung einer Chatbot-Anfrage

Abbildung 2 zeigt den Prozess, der durchlaufen wird, wenn Benutzer:innen eine Frage an den Chatbot stellen. Die beiden Begriffe *sparse* und *dense* bedeuten in diesem Kontext, dass die entsprechende Suche lexikografischer oder semantischer Natur ist. Die beiden Suchresultat werden kombiniert und dann durch den Reranker noch einmal neu bewertet. Die relevantesten Textpassagen werden anschliessend zusammen mit der Frage an das LLM weitergegeben, das daraus eine Antwort generiert.

6 Features

Im Folgenden werden einzelne Features des Chatbots beschrieben, die zur Verbesserung der Benutzerfreundlichkeit und der Qualität der Antworten implementiert wurden.

6.1 Contextual Retrieval

Bei Contextual Retrieval[10] handelt es sich um eine Technik, welche die Suche von relevanten Textpassagen bei sogenannten «Retrieval-Augmented Generation» (RAG) verbessert. Da Textpassagen immer nur einen Teil des gesamten Dokuments abbilden, kann es passieren, dass wichtige Informationen, welche zum Verständnis der Textpassage benötigt werden, nicht in der gleichen Textpassage enthalten sind. Dies kann dazu führen, dass gewisse Textpassagen nicht gefunden werden oder dass die Textpassagen aufgrund fehlender Information durch das LLM falsch interpretiert werden, was eine inkorrekte Antwort zur Folge haben kann.

Um diesem Problem entgegenzuwirken, können mittels eines LLMs Zusatzinformationen zu jeder Textpassage generiert werden. Dabei erhält das LLM die Aufgabe, jede Textpassage im gesamten Dokument zu situieren. Das LLM verwendet einerseits die entsprechende Textpassage und nutzt andererseits das gesamte Dokument. Damit kann das LLM eine Beschreibung liefern, wie dieser spezifische Textausschnitt im gesamten Dokument eingebettet ist. Diese Beschreibung kann dann in Form von Metainformation zusammen mit der Textpassage abgespeichert werden. Davon profitiert einerseits die Suche

[10] Siehe Blogbeitrag von Anthropic

nach relevanten Textpassagen und andererseits auch die Antwortformulierung durch das LLM, das auf die Frage der Benutzer:innen anhand der gefundenen Dokumente antworten muss.

Dieses Verfahren wird nur bei der Generierung der Vektor-Datenbank angewendet. Im Betrieb des Chatbots sind diese Informationen dann bereits verfügbar. Eine Schwierigkeit bei der Implementierung dieses Verfahrens ist die maximale Länge des Textes, den das LLM auf einmal verarbeiten kann. Im Falle des Modells, welches wir verwenden, ist diese Maximallänge zwar relativ gross, jedoch nicht gross genug, um sehr grosse Dokumente (100-300 Seiten) auf einmal zu verarbeiten. Aus diesem Grund haben wir das Verfahren so angepasst, dass es auch mit grösseren Dokumenten funktioniert. Dabei wird eine Textpassage anhand der 3 vorherigen und den 3 nachfolgenden Textpassagen im Dokument situiert. Zusätzlich werden bei der Situierung eine Beschreibung des Dokuments (Summary) und bestimmte Restriktionen (Restrictions) zur Hilfe gezogen. Diese wurden ebenfalls durch ein LLM generiert, das lediglich Zugriff auf die ersten 10 Textpassagen des Dokuments hat.

Ein Beispiel einer solchen Situierung ist im Anhang abgebildet.

6.2 Follow-Up Questions

Nachdem Benutzer:innen eine Frage gestellt und eine Antwort erhalten haben, können sie im nächsten Schritt Bezug auf die vorherige Frage nehmen. Beispielsweise könnte die erste Frage lauten: «Wann bekommen Professoren ihre Niederlassungsbewilligung?», woraufhin der Chatbot beispielsweise antwortet: «Ordentliche und ausserordentliche Professorinnen und Professoren, die an einer Universität, an einer Eidgenössischen Technischen Hochschule oder am Institut Universitaire des Hautes Etudes Internationales (IUHEI) unterrichten, erhalten die Niederlassungsbewilligung sofort.»

Eine Folgefrage könnte lauten: «Was heisst sofort?»

In diesem Fall wandelt der Chatbot die Frage in eine Folgefrage um, die ohne zusätzlichen Kontext verständlich ist. In diesem Beispiel wurde die Frage umgewandelt in die Frage «Was bedeutet "sofort" im Zusammenhang mit der Erteilung einer Niederlassungsbewilligung an bestimmte Personen?». Dies geschieht automatisch im Falle einer Folgefrage und ist auf den ersten Schritt nicht ersichtlich, um Benutzer:innen nicht zu verwirren. Die Umformulierung kann jedoch eingesehen werden, indem zuerst die Quellen mit einem Klick auf «Quellen» ausgeklappt werden und dann der Button «Zwischenschritte inspizieren» betätigt wird. Im Falle einer Umformulierung wird einerseits die ursprüngliche Frage, und andererseits die Neuformulierung angezeigt. Der Chatbot erkennt automatisch, ob sich eine Frage auf eine vorherige Frage oder Antwort bezieht, oder ob es sich um eine neue Frage handelt.

6.3 Quellen anzeigen

Der Chatbot gibt bei den Antworten jeweils die Quellen (Textpassagen aus verschiedenen Dokumenten) an, die er für die Antwort verwendet hat. Diese können mit einem Klick auf das Dropdown-Element «Quellen», das in Abbildung 3 abgebildet ist, geöffnet werden. Im Folgenden wird beschrieben, wie diese Quellen besser interpretiert werden können.

Abbildung 3: Dropdown, welches das Einsehen von Quellen ermöglicht, auf denen die Antwort des Chatbots basiert.

6.3.1 Zitierte und nicht zitierte Quellen

Nachdem das Dropdown «Quellen» geöffnet wurde, sind die Quellen ersichtlich, die durch den Chatbot gefunden wurden. Wie dies aussieht, ist in Abbildung 4 ersichtlich. Dabei fällt auf, dass nicht alle Quellen zitiert wurden («Gefundene Dokumente, die nicht zitiert wurden»). Dies hängt damit zusammen, dass nicht immer alle gefundenen Textpassagen relevant sind. Der Chatbot entscheidet eigenständig, welche der gefundenen Textpassagen für die Antwort berücksichtigt werden sollen und welche nicht. Auch die Zitierung der einzelnen Quellen im Antworttext anhand der eckigen Klammern und der Quellennummer wird durch den Chatbot bestimmt. Bei Unsicherheiten kann es Sinn machen, sich auch die nicht zitierten Textpassagen genauer anzuschauen.

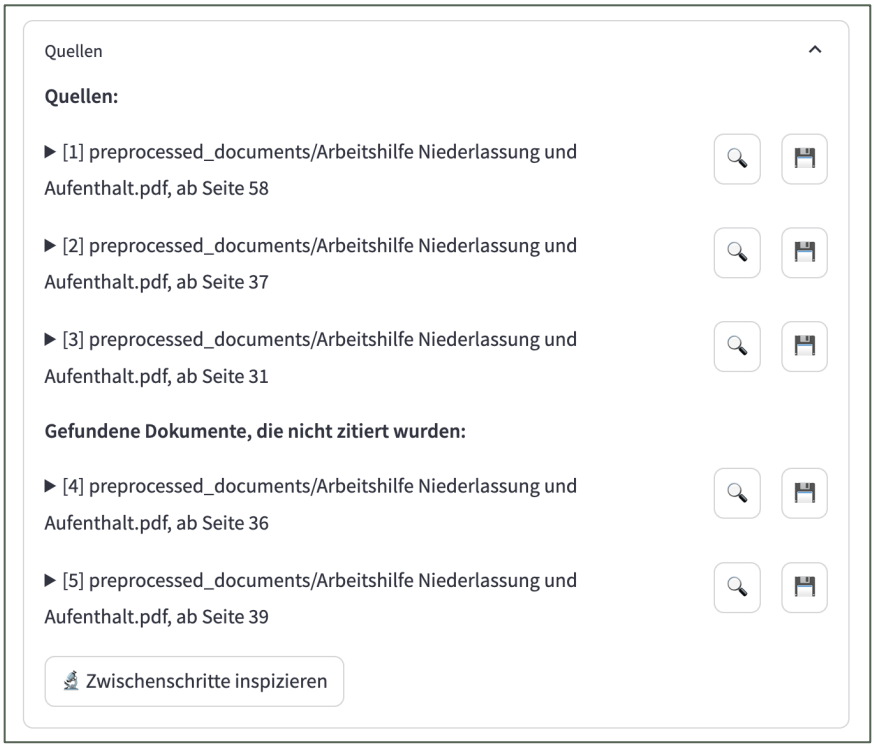

Abbildung 4: Gefundene Textpassagen (Quellen) auf eine spezifische Frage.

6.3.2 Text einer Quelle auslesen

Um den Text einer Textpassage auszulesen, muss auf den Namen der Quelle geklickt werden. Dies wird auch visuell durch das dreieckige Symbol angedeutet, siehe Abbildung 4. Sobald dies geschieht, wird der Inhalt der Quelle ausgeklappt und der Text ist ersichtlich. Ein erneutes Klicken auf den Namen der Quelle klappt den Text wieder ein.

6.3.3 Dokument anzeigen

Im Falle von PDF-Dateien können die Textpassagen mit Klick auf die Lupe auch direkt im Browser angezeigt werden. Es öffnet sich ein Fenster, in dem die PDF-Datei geladen wird, was allerdings einen Moment dauern kann. Die entsprechende Seite, auf der die Textpassage beginnt, wird automatisch angezeigt.

6.3.4 Dokument herunterladen

Nebst der Option, eine Textpassage anzuzeigen, kann diese auch heruntergeladen werden, indem auf das Disketten-Symbol neben der Quelle geklickt wird. Dies veranlasst einen Download der Datei, um diese auf dem eigenen Laptop oder Smartphone genauer anzuschauen.

6.3.5 Zwischenschritte inspizieren

Manchmal kann es nützlich sein, den ganzen Prozess des Chatbots nachzuverfolgen. Für diesen Fall wurde eine Funktion implementiert, die wir «Zwischenschritte inspizieren» nennen. Die Funktion ist ganz unten im ausgeklappten Dropdown «Quellen» zu finden, siehe Abbildung 4. Sobald auf diesen Button geklickt wird, öffnet sich ein Fenster, das detaillierte Informationen zu den Zwischenschritten enthält. An erster Stelle wird die ursprüngliche Frage angezeigt. Falls es sich um eine Follow-Up-Frage handelt, wird auch die umformulierte Frage dargestellt. Diese beiden Informationen sind in Abbildung 5 und Abbildung 6 abgebildet.

Abbildung 5: Ansicht der Zwischenschritte im Falle einer neuen Frage.

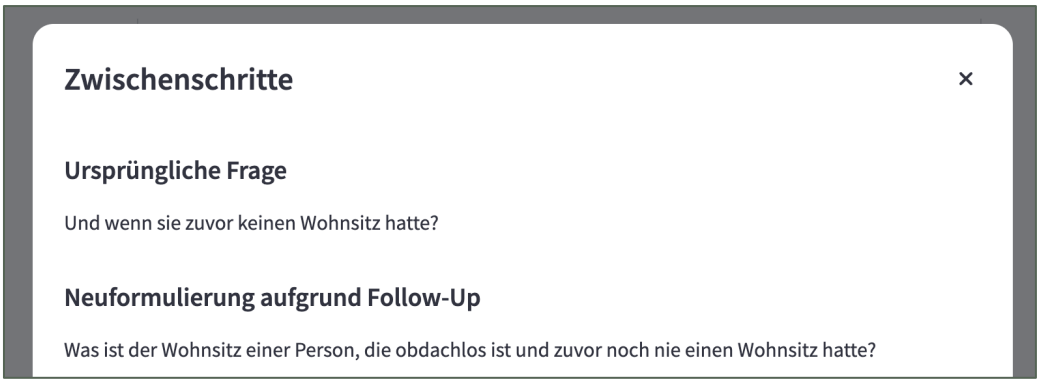

Abbildung 6: Ansicht der Zwischenschritte im Falle einer Follow-Up-Frage.

Ausserdem ist in diesem Fenster ersichtlich, welche Textpassagen ursprünglich alle gefunden und zum Teil wieder verworfen wurde. Der Abschnitt «Gefundene Textausschnitte mittels BM25» bezieht sich auf Textpassagen, die anhand von Stichwörtern (lexikografisch) mittels dem BM25 Algorithmus[11] gefunden wurden. Dies sind initial deutlich mehr als die sieben Quellen, die der Chatbot am Ende zur Generierung der Antwort zur Verfügung hat, was gewollt ist. Der Grund dafür ist, dass der Reranker danach eine lange Liste von Textpassagen erhält und davon eine neue Rangliste generiert, welche die Relevanz der Quellen widerspiegelt. Erst in dieser verfeinerten Rangliste werden dann die relevantesten Textausschnitte für den Chatbot ausgewählt.

Nach den Textpassagen, die mittels des BM25 Algorithmus gefunden wurden, stehen die Textpassagen, die mittels einer semantischen Suche basierend auf Embeddings gefunden wurden. Diese sind unter dem Abschnitt «Gefundene Textausschnitte mittels Vektor-Embedding» ersichtlich. Zuletzt sind diejenigen Textpassagen ersichtlich, die durch den Reranker am relevantesten eingestuft wurden. Diese sind unter dem Abschnitt «Top-7 Dokumente nach Re-Ranking» ersichtlich.

Jede dieser Quellen ist aufklappbar um weitere Informationen anzuzeigen. Bereits im Namen der Quelle ist ein Score angegeben, der beschreibt, wie ähnlich die Textpassage zur Frage ist, die an den Chatbot gestellt wurde. Da der BM25-Algorithmus und die semantische Suche Scores mit unterschiedlichen Skalen generieren, können diese nicht direkt miteinander verglichen werden. Nur innerhalb eines einzelnen Abschnitts sind die Scores miteinander vergleichbar. Der Reranker ermöglicht es schliesslich, die gefundenen Textpassagen in eine einzelne Rangliste zu kombinieren und mit einem einheitlichen Score zu bewerten.

Abbildung 7 zeigt, wie eine ausgeklappte Quelle im Menü «Zwischenschritte inspizieren» aussieht. Die Quelle enthält Informationen über die Textpassage in Form eines JavaScript Object Notation (JSON) Elements. Dieses beinhaltet folgende Inhalte:

- **chunk_id**: Identifikationsnummer (ID) der Textpassage
- **document_id:** ID des Dokuments, aus dem die Textpassage stammt
- **file_path**: der Pfad zur Datei innerhalb des Chatbots
- **page**: Die Seite, auf der die Textpassage innerhalb des Dokuments beginnt. Manchmal stimmt diese Seitenzahl nicht mit den Seitenzahlen überein, welche in dem Dokument in der Kopf- oder Fusszeile angezeigt wird. Das hat damit zu tun, dass teilweise gewisse für den Chatbot irrelevante Seiten entfernt wurden. Deshalb ist die absolute Seitenzahl innerhalb des PDFs gemeint und nicht die Seitenzahl in der Kopf- oder Fusszeile.
- **split_id**: Die ID der Textpassage innerhalb des Dokuments. Die Nummerierung beginnt bei 0, das heisst die ID 23 bedeutet beispielsweise, dass es sich um die vierundzwanzigste Textpassage im Dokument handelt.
- **score**: Der Score, der dieser Textpassage initial zugeteilt wurde (entweder durch den BM25-Algorithmus oder die semantische Suche).
- **reranker_score**: Der Score, der dieser Textpassage durch den Reranker zugewiesen wurde (zwischen 0 und 1).
- **contextual_metadata**: Enthält Informationen, die via dem Verfahren Contextual Retrieval (siehe Abschnitt 6.1) generiert wurden.
- **content**: Der Inhalt der Textpassage.

Alle diese Informationen können einfach kopiert werden, indem das blaue Icon neben dem entsprechenden Eintrag angeklickt wird. Teilweise ist dieses Symbol erst sichtbar, wenn der Mauszeiger über der entsprechenden Stelle ist.

[11] Für mehr Informationen siehe den Artikel Okapi BM25 auf Wikipedia

Abbildung 7: Die Details einer Textpassage, wenn der Button «Zwischenschritte inspizieren» angeklickt wurde.

7 Evaluation seitens BFH

Wir haben den Chatbot seitens BFH sowohl qualitativ manuell als auch quantitativ (soweit möglich) automatisiert evaluiert. Die qualitative Evaluation wurde dabei von Lara Burkhalter[12] durchgeführt und wird in Abschnitt 7.1 von ihr beschrieben. Sie ist Anwältin und arbeitet am IPST als wissenschaftliche Mitarbeiterin. Die quantitative Analyse basiert grösstenteils auf den Fragen, die im Rahmen der qualitativen Evaluation erstellt wurden. Im Weiteren wird das Vorgehen und die Resultate der beiden Evaluationen beschrieben.

7.1 Qualitative Evaluation

Bei der qualitativen Evaluation wurden die Antworten und Quellenangaben des Chatbots auf ihre Richtigkeit überprüft. Dabei wurden dem Chatbot bestimmte Fragen gestellt. Diese Fragen wurden auf der Grundlage der zur Verfügung gestellten Dokumente formuliert. Grundsätzlich wurde eine Frage auf Basis eines bestimmten Dokuments (sog. Informationsgrundlage) erarbeitet. Gestützt auf derselben Informationsgrundlage wurde die erwünschte Antwort auf diese formulierte Frage erstellt. Anschliessend wurde dieselbe Frage dem Chatbot gestellt. Die durch den Chatbot generierten Antworten wurden protokolliert und mit den erwünschten Antworten verglichen. Sodann wurden die zitierten Dokumente ebenfalls festgehalten und mit der Informationsgrundlage (Dokument, auf dem die Frage basierte) verglichen. Dabei wurde berücksichtigt, ob eine Antwort auf der Grundlage weiterer Dokumente gegeben werden konnte. In diesem Fall wurden die weiteren zitierten Quellen auf ihren Inhalt und ihre Richtigkeit in Bezug auf die Frage überprüft.

Insgesamt ist der Chatbot sowohl bei den Antworten als auch bei den Quellenangaben sehr zufriedenstellend. Vor allem die genaue Zitierung der verwendeten Dokumente zeichnet den Chatbot besonders aus und hebt ihn von weiteren KI-Sprachmodellen ab. Bemerkenswert ist auch, dass der Chatbot den Zusammenhang zwischen den Dokumenten erkennt und eine Antwort auf der Grundlage mehrerer unterschiedlicher Dokumente richtig formulieren kann.

Was bei der Nutzung des Chatbots berücksichtigt werden muss, ist die Tatsache, dass es sich um ein KI-Sprachmodell handelt und dementsprechend auch unvollständige Resultate liefern kann. Das bedeutet, dass Fragen grundsätzlich korrekt beantwortet werden, aber gewisse Teile der Inhalte fehlen können. Dies geschieht, indem der Chatbot manchmal nur bestimmte Abschnitte der relevanten Textpassagen in seine Antwort einfliessen lässt. Durch die integrierte Quellenangabe und durch die Funktion «PDF-Viewer» kann eine Antwort jedoch jederzeit auf ihre Vollständigkeit überprüft werden.

7.2 Automatisierte Evaluation

Für die automatisierte Evaluation wurden einerseits die Fragen, Antworten und Informationsgrundlagen von der qualitativen Evaluation, welche in Abschnitt 7.1 beschrieben wird, verwendet. Andererseits wurden Inhalte aus dem Dokument «Fragen-Antworten-Implement»[13], das uns zur Verfügung gestellt wurde und 9 Frage-Antwort-Paare mit den jeweiligen Quellenangaben verwendet.

Das Dokument «Fragen-Antworten-Referenz-Katalog EM IMPLEMENT»[14] wurde nicht verwendet, da beispielsweise bei der ersten Frage ein Dokument[15] benötigt wird, welches wir nicht in der Dokumentensammlung fanden und deshalb davon ausgegangen werden musste, dass dies evtl. bei weiteren Fragen aus diesem Dokument der Fall ist. An dieser Stelle sei erwähnt, dass der Chatbot nur so gut sein kann wie die vorhandene Dokumentengrundlage es erlauben kann. Eine faire Evaluation berücksichtigt also nur solche Fragen, die auch aufgrund der vorliegenden Dokumente beantwortet werden kann. Eine Liste aller Dokumente, die der Chatbot «kennt», ist in der beiliegenden Excel-Tabelle enthalten.

[12] Siehe https://www.bfh.ch/de/lara-burkhalter
[13] Siehe folgendes SharePoint-Dokument: Fragen-Antworten-Implement
[14] Siehe folgendes SharePoint-Dokument: Fragen-Antworten-Referenz-Katalog EM IMPLEMENT
[15] Das benötigte Dokument haben wir schlussendlich gefunden, es ist aber nicht Teil der Dokumentensammlung und somit dem Chatbot nicht zugänglich.

7.2.1 Evaluation der Suchergebnisse (Retrieval)

In dieser Evaluation geht es darum zu prüfen, wie gut der Chatbot dabei ist, die richtigen Dokumente zu finden, auf denen die Antwort basieren soll. Je nachdem, ob die Information in denen durch den Chatbot als am relevantesten eingestuften Textstellen vorkommt oder nicht, schneidet der Chatbot besser oder schlechter ab. Auch die Reihenfolge der Relevanz spielt eine Rolle.

Als Grundlage für diese Evaluation dienten die zuvor beschriebenen Dokumente, was zu einem Testdatensatz mit 49 Beispielen führte. Diese Beispiele enthielten jeweils eine Frage und eine Liste von Quellen, welche zur Beantwortung der Frage berücksichtigt werden sollten. Die Gefahr bei dieser Evaluation ist, dass es noch weitere Dokumente gibt, welche die Frage auch beantworten würden, jedoch nicht in den Testdaten aufgelistet sind. Das führt zu einer tiefen Bewertung des Chatbots, obwohl dieser die Frage vielleicht sogar richtig beantwortet – nur aufgrund eines anderen Dokuments.

Eine weitere Einschränkung dieser Evaluation ist, dass wir jeweils die Seitenangaben für die entsprechende Information im Dokument haben, wir aber nicht genau wissen in welcher Textpassage (Chunk) sich die Information befindet. Die Evaluation ist so implementiert, dass sie auf folgender Heuristik basiert: Eine gefundene Textpassage wird als korrekt interpretiert, wenn sie die «richtige» Seite beinhaltet. Das bedeutet, dass die Textpassage entweder auf dieser Seite beginnt, endet, oder dass der ganze Inhalt der entsprechenden Seite inmitten einer Textpassage vorkommt.

Dennoch kann diese Evaluation einen Anhaltspunkt geben, wie gut der Chatbot diejenigen Dokumente findet, die ein Mensch aufsuchen würde. Die Evaluation der Suchergebnisse umfasst folgende Metriken: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG) und Recall. Im Folgenden ist eine kurze Beschreibung dieser Metriken erläutert:

MAP berechnet für jeden Rang in der Rangliste der als am relevantesten eingestufte Dokumente wie viele der Dokumente wirklich relevant sind. Am Ende werden diese Werte gemittelt und es resultiert ein einzelner Wert, der aussagt, wie gut die Suche die richtigen Dokumente findet und diese in ihrer Wichtigkeit einordnet. Je höher ein relevantes Dokument in der Liste der gefundenen Dokumente, desto höher der MAP Score.

MRR sagt etwas darüber aus, wie weit oben das erste relevante Dokument in der Liste der gefundenen Dokumente steht. Je höher der Wert, desto höher das erste relevante Dokument in der Rangliste der gefundenen Dokumente. Dieser Wert wird über alle Suchanfragen gemittelt.

nDCG fokussiert sich auf die Reihenfolge der gefundenen Dokumente. Je höher die relevanten Dokumente in dieser Liste stehen, desto besser der Score.

Recall beschreibt, wie viele der Dokumente, die in Wirklichkeit relevant sind, in den finalen Suchergebnissen (nach dem Re-Ranking) vorkommen.

7.2.2 Einstellungsmöglichkeiten über «Hyperparameter»

Da es eine Vielzahl an Parametern gibt, die bei Systemen mit RAG optimieren werden können, haben wir uns nur auf einen kleinen Teil dieser Parameter beschränkt. Insbesondere wollten wir wissen, wie viele Dokumente die Suche mit BM25 zurückgeben soll (Top K Sparse) und wie viele Dokumente der Reranker am Ende dem Chatbot zur Verfügung stellen soll (Top K Reranker). Weitere Einstellungen (sogenannte «Hyperparameter»), die wir für diese Experimente fixiert haben, sind folgende Vorgaben:

Parameter für NLTKDocumentSplitter:
- Split by: word
- Split length: 400
- Split overlap: 50
- Split threshold: 15
- Respect Sentence Boundary: True
- Language: de

Parameter für Retrieval
- Top K Dense: 30
- Embedding Model: Snowflake/snowflake-arctic-embed-l-v2.0
- Reranker Model: BAAI/bge-reranker-v2-m3

Die genaue Beschreibung des NLTKDocumentSplitters[16] ist in der Dokumentation von Haystack erläutert. Diese Einstellungen führen zu Chunks (Textpassagen) mit 400 Wörtern, wobei es zwischen den einzelnen Chunks eine Überlappung in einem Rahmen von 50 Wörtern gibt. Satzgrenzen sollen dabei, wenn möglich, berücksichtigt werden. Für das semantische (dense) Retrieval haben wir den Top-K Wert, der für die Anzahl der gefundenen Dokumente steht, auf 30 festgelegt. Somit werden die 30 relevantesten Textstellen verarbeitet. Das Embedding-Modell und das Reranker-Modell gelten ebenfalls als Hyperparameter.

7.2.3 Ergebnisse der Evaluation

Die Resultate der Tests verschiedenen Einstellungen von «Top K» Werten sind in Tabelle 1 zu sehen. Es fällt auf, dass die Unterschiede nicht sehr gross sind. Die besten Werte pro Metrik sind hervorgehoben. Da nDCG und Recall bei dem Setting mit den Top-K-Werten 20 (Sparse) und 7 (Reranker) am besten sind, entschieden wir uns für diese Konfiguration.

Tabelle 1: Ergebnisse der Evaluation verschiedener Top-K-Werte für die lexikografische Suche (Sparse) und den Reranker.

Top K (Sparse)	Top K (Reranker)	MRR	MAP	nDCG	Recall
10	4	0.5561	**0.5493**	0.4348	0.6939
10	5	0.5602	0.5412	0.4464	0.7143
10	6	**0.5636**	0.5378	0.4588	**0.7347**
10	7	**0.5636**	0.5378	0.4588	**0.7347**
20	4	0.5493	0.5425	0.4331	0.6735
20	5	0.5575	0.5384	0.4526	0.7143
20	6	0.5609	0.5350	0.4604	**0.7347**
20	7	0.5609	0.5354	**0.4646**	**0.7347**
30	4	0.5493	0.5425	0.4331	0.6735
30	5	0.5575	0.5446	0.4489	0.7143
30	6	0.5575	0.5310	0.4567	0.7143
30	7	0.5604	0.5342	0.4641	**0.7347**

7.2.4 Evaluation des LLM für die Antwort-Generierung

[16] Siehe https://docs.haystack.deepset.ai/docs/nltkdocumentsplitter

In dieser Evaluation wollten wir uns neben den gefundenen Dokumenten auch die ausformulierten Antworten anschauen, um das richtige LLM für die Generierung der Antworttexte auszuwählen. Dazu haben wir uns für die Metriken «Answer Relevancy»[17] und «Faithfulness»[18] entschieden und diese mit dem Framework DeepEval, bzw. mit der Integration von DeepEval in Haystack, evaluiert. Beide Metriken werden anhand eines LLMs generiert. Bei der Answer Relevancy muss das LLM entscheiden, wie relevant die Antwort auf die gestellte Frage ist. Bei der Faithfulness entscheidet das LLM, ob die gegebene Antwort den Fakten entspricht, die in den gefundenen Textpassagen erwähnt werden. Hier wird also geprüft, ob das LLM Informationen verfälscht oder erfindet. Bei beiden dieser Metriken muss das LLM jeweils eine Begründung angeben. Das LLM, das wir als «Judge» für diese beiden Metriken verwendet haben, ist «Qwen2.5-72B-Instruct».

Die beiden Metriken wurden anhand von 49 Frage-Antwort-Paaren evaluiert. Tabelle 1 zeigt die Ergebnisse dieser Evaluation. Qwen2.5-72B-Instruct hat die höchste Faithfulness, jedoch eine etwas geringere «Answer Relevancy»[19]. Bei DeepSeek-V3 konnten wir leider nur die Answer Relevancy evaluieren, da die Evaluation der Faithfulness in DeepEval aufgrund eines Formatierungsfehlers scheiterte. In Bezug auf die Answer Relevancy hat DeepSeek-V3 in unseren Tests jedoch nicht so gut abgeschnitten wie die anderen LLMs. Wir konnten eine deutliche Verbesserung der Answer Relevancy zwischen Llama-3.1-70B-Instruct und Llama-3.3-70B-Instruct feststellen. Da Llama-3.3 in beiden Metriken gut abschneidet, haben wir uns dazu entschieden, dieses Modell für die Generierung der Antworten einzusetzen. In Zukunft könnten weitere Tests mit mehr Beispielen und weiteren Metriken die Analyse zusätzlich verfeinern, sollte das Projekt mit dem Ziel eines produktiven Einsatzes weiterverfolgt werden.

Tabelle 2: Auswertung der generierten Antworten von verschiedenen LLMs.

LLM	Metrik	Mittelwert	Standardabweichung
Qwen2.5-72B-Instruct	Answer Relevancy	0.900251	0.225611
	Faithfulness	0.943631	0.098946
DeepSeek-V3	Answer Relevancy	0.892910	0.224147
Llama-3.3-70B-Instruct	Answer Relevancy	0.920099	0.202174
	Faithfulness	0.922451	0.113597
Llama-3.1-70B-Instruct	Answer Relevancy	0.859147	0.265897
	Faithfulness	0.929628	0.111001

[17] Siehe https://docs.confident-ai.com/docs/metrics-answer-relevancy
[18] Siehe https://docs.confident-ai.com/docs/metrics-faithfulness
[19] In diesem Fall hat sich Qwen selbst evaluiert, da sowohl bei der Antwort-Generierung als auch bei der Evaluation das gleiche LLM verwendet wurde.

8 Mögliche Weiterentwicklungen und Verbesserungspotenziale

Im Folgenden einige Ideen, wie der Chatbot weiterentwickeln könnte, sollte eine produktive Version eines solchen Chatbots geplant werden.

1. Es sollte eine zentrale Liste aller Dokumente geben, welche diese einerseits in verschiedene Kategorien einstuft (Gesetze, Rundschreiben, FAQs, etc.) und andererseits dazu dienen soll, diese periodisch automatisch in die Datenbank einzupflegen, sollte eine neue Version eines Dokuments verfügbar sein.
2. Der Chatbot könnte in Form eines «Agentic Workflows» abgebildet werden, bei dem das LLM Zugriff auf gewisse Tools hat und autonom entscheidet, welches Tool für diese Frage am meisten Sinn macht. Ein Tool wäre dann das in diesem PoC implementierte System. Es sind aber noch weitere Tools denkbar wie beispielsweise eine einfache Abfrage der FAQs oder evtl. eine Websuche, die sich auf spezifische Webseiten beschränkt.

Am wichtigsten ist jedoch die Vollständigkeit der Daten. Alte Dokumente, die nicht mehr gültig sind, sollten regelmässig aus der Datenbank gelöscht werden. Neue Dokumente müssen jeweils zeitnah zur Datenbank hinzugefügt werden, ansonsten sind die Informationen des Chatbots nicht mehr aktuell.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

Beispiel Contextual Retrieval:

Beschreibung des gesamten Dokuments anhand der ersten 10 Textpassagen (Summary):
Das Dokument beschreibt die Einführung und Umsetzung der Lohnmeldung für entsandte Dienstleistungserbringer aus dem EU/EFTA-Raum in der Schweiz. Es erläutert die Änderungen in verschiedenen Verordnungen und die technischen Anpassungen im Online-Meldeverfahren und in ZEMIS, um die Lohnmeldung zu ermöglichen.

Beschreibung der Einschränkungen, die für das Verständnis des Dokuments wichtig sind, anhand der ersten 10 Textpassagen (Restrictions):
Das Dokument bezieht sich auf die Schweiz und den EU/EFTA-Raum, insbesondere auf die Ausländerbehörden der Kantone und des Fürstentums Liechtenstein sowie auf die Arbeitsmarktbehörden der Kantone. Es gilt für entsandte Dienstleistungserbringer, die in der Schweiz tätig sind, und trat am 1. Mai 2013 in Kraft.

Situierung der Textpassage im Dokument anhand der vorherigen 3 und darauffolgenden 3 Textpassagen (Context):
Im Rahmen des Abkommens mit der Europäischen Union über die Personenfreizügigkeit hat die Schweiz flankierende Massnahmen zum freien Personenverkehr eingeführt, um die Arbeitsbedingungen von entsandten Arbeitnehmern zu verbessern. Eine dieser Massnahmen ist die Einführung einer Lohnmeldung für ausländische Dienstleistungserbringer aus dem EU/EFTA-Raum, die in der Schweiz tätig sind. Diese Lohnmeldung soll sicherstellen, dass die entsandten Arbeitnehmer den in der Schweiz geltenden Mindestlöhnen entsprechen. Im Folgenden werden die Änderungen in verschiedenen Verordnungen und die technischen Anpassungen im Online-Meldeverfahren und in ZEMIS erläutert, um die Lohnmeldung zu ermöglichen.

Inhalt der Textpassage:
Schweizerische Eidgenossenschaft Eidgenössisches Justiz- und Polizeidepartement EJPD
Confdration suisse Bundesamt fur Migration BFM
Confederazione Svizzera
Confederaziun svizra
Eidgenössisches Departement für Wirtschaft, Bildung und Forschung WBF
Staatssekretariat für Wirtschaft SECO
Rundschreiben
An : Ausländerbehörden der Kantone und des Fürsten
tums Liechtenstein sowie der Städte Bern, Biel und
Thun
Arbeitsmarktbehörden der Kantone
Ort, Datum : Bern-Wabern, den 29. April 2013
ReferenzlAktenzeichen : FS 2013-03-22/7

Abkommen mit der Europäischen Union über die Personenfreizügigkeit: Einführung und Umsetzung der Lohnmeldung für entsandte Dienstleistungserbringer
Sehr geehrte Damen und Herren
Der Bundesrat hat anlässlich seiner Sitzung vom 16. April 2013 drei Verordnungsänderun gen verabschiedet. Diese stehen im Zusammenhang mit der in der Sommersession 2012 durch die eidgenössischen Räte verabschiedeten Revision der Entsendegesetzgebung zur Verbesserung der flankierenden Massnahmen zum freien Personenverkehr.
Das Parlament hat in der Schlussabstimmung der letztjährigen Sommersession vom 15. Juni 2012 u.a. beschlossen eine Lohnmeldung für ausländische Dienstleistungserbringer aus dem EU/EFTA-Raum in Art. 6 Abs. 1 Bst. a. des Bundesgesetzes über die flankierenden Massnahmen bei entsandten Arbeitnehmerinnen und Arbeitnehmern und über die Kontrolle der in Normalarbeitsverträgen vorgesehenen Mindestlöhne (EntsG)1 einzuführen. Art. 6 Abs. 1 Bst. a. EntsG tritt per 1. Mai 2013 in Kraft. Somit ist ab diesem Zeitpunkt eine gesetzliche Grundlage zur Erhebung der Lohnmeldung vorhanden.
Das vorliegende Rundschreiben des Bundesamtes für Migration (BEM) und des Staatssekre tariats für Wirtschaft (SECO), soll Sie über die sich daraus ergebenden Änderungen infor mieren.
1 SR 823.20
Bundesamt für Migration BFM
Quellenweg 6, 3003 Bern-Wabern
Tel. 41 31 32511 11, Fax. 41 31 3255843
www.bfm.admin.ch
Erläuterungen zu den drei Verordnungsänderungen
Oben erwähnte Anpassung von Art. 6 Abs. 1 Bst. a. EntsG macht Änderungen bzw. Präzisie rungen in folgenden drei Verordnungen nötig:
- Art. 6 Abs. 4 Bst. a. der Verordnung über die in die Schweiz entsandten Arbeitnehmerin nen und Arbeitnehmer (EntsV)2:
Art. 6 Abs. 4 Bst. abs EntsV verpflichtet den ausländischen Arbeitgeber im Rahmen einer Meldung (via Online-Meldeverfahren) von entsandten Arbeitnehmenden den effektiv für die gemeldete Dienstleistung in der Schweiz entrichteten Bruttostundenlohn für die entspre chend in der Schweiz ausgeübte Tätigkeit zu melden.
Der Arbeitgeber muss den in der Schweiz für die ausgeübte Tätigkeit sowie die berufliche Qualifikation des entsandten Arbeitnehmenden geltenden Bruttostundenlohn melden. Die Lohnmeldung gilt für Entsandte unabhängig von der Branche in der sie tätig sind.
- Art. 9 Abs. 1 bis der Verordnung über die schrittweise Einführung des freien Personenver kehrs (VEP)3:
Die Revision von Art.

Versionskontrolle

Version	Datum	Beschreibung	Autor
0.1	07.01.2025	Dokument erstellt	Luca Rolshoven
0.2	09.01.2025	Ergänzung des Abschnitts «Qualitative Evaluation»	Lara Burkhalter
0.3	10.01.2025	Fertigstellung des Berichtsentwurfs	Luca Rolshoven
1.0	13.01.2025	Kleinere Anpassungen und Finalisierung	Luca Rolshoven, Matthias Stürmer