

Memorandum

An: Benutzerinnen und Benutzer von Red Ink
Von: David Rosenthal, VISCHER AG
Datum: 16. Oktober 2025 (wird laufend nachgeführt)
Betreift: Anleitung zu Red Ink – Generation 2

I.	ÜBERBLICK	2
II.	BENUTZUNG	4
A.	Grundsätzliche Handhabung	4
B.	Grundfunktionen von Red Ink in Word	7
C.	Freestyle-Funktion	18
D.	KI-Texte im eigenen Schreibstil: MyStyle	32
E.	Suche nach Themen: Context Search	34
F.	Zugriff auf weitere Datenquellen und Dienste: Special Services	37
G.	Transcriptor	41
H.	Document Check	47
I.	Klauseldatenbank	52
J.	Podcasts und Audiobooks erstellen	55
K.	Integrierte Anonymisierungs-Funktion	59
L.	WebAgent	64
M.	Word-Helpers – praktische Alltags-Helfer (fast) ohne KI	67
N.	Integrierter Chatbot "Inky"	70
O.	Weitere Tipps zur Handhabung von Red Ink in Word	72
P.	Funktionen von Red Ink in Excel	74
Q.	CSV Analyzer	81
R.	Funktionen von Red Ink in Outlook	83
S.	Separater Chatbot "Inky"	87
T.	Browser-Erweiterung	89
U.	Verwendung von Red Ink durch andere Programme	91
III.	INSTALLATION	94
A.	Für Ungeduldige: Die Installation mit einem Klick	94
B.	Die Installation im Detail	95

C.	Vorbereitung: API-Zugang	97
D.	Schritt 1: Installer bzw. Installationspaket herunterladen	99
E.	Schritt 2: Installer ausführen.....	100
F.	Schritt 3: Erste Konfiguration mittels Assistent	101
G.	Schritt 4: Helper installieren (Optional, kann später erfolgen).....	103
H.	Schritt 5: Add-ins benutzen.....	104
I.	Schritt 6: Bei Bedarf weitere Anpassungen vornehmen	105
J.	Installation der Browser-Erweiterung	105
IV.	KONFIGURATION (FÜR FORTGESCHRITTENE)	106
A.	Konfigurationsdatei "redink.ini"	106
B.	Prompt-Bibliothek	131
C.	Weitere alternative Sprachmodelle	132
D.	OAuth2.0 (z.B. Google Vertex API).....	133
E.	Konfiguration erweiterter API-Calls.....	134
F.	Sicherheitsfunktionen	136
V.	HÄUFIGE FRAGEN (FAQ)	139
VI.	RELEASE NOTES	153
VII.	ROADMAP	160
ANHANG 1: IDEEN ZUM KENNENLERNEN VON RED INK		161
ANHANG 2: PROGRAMMIEREN VON RESPONSE-TEMPLATES		164
ANHANG 3: WEB AGENT SCRIPTING		168

I. ÜBERBLICK

- 1 Red Ink ist ein von uns ursprünglich für unsere internen Bedürfnisse als Wirtschaftskanzlei selbst entwickeltes **KI-Tool**, welches in Form von Office-Add-ins direkt in **Word**, **Excel** und in **Outlook** integriert ist und es dort erlaubt, verschiedenste KI-Funktionen mit den eigenen Texten, Arbeitsblättern und Mails auszuführen. Prompts können direkt eingegeben werden, es ist aber ebenso möglich, einen selektierten Text, Zellen in einem Arbeitsblatt oder E-Mail-Ketten von der KI übersetzen, überarbeiten, zusammenfassen, kommentieren und durchsuchen zu lassen. Sogar ein Chatbot steht zur Verfügung, das Tool kann transkribieren, aus Texten



- Audiobooks herstellen und aus dem Browser kann auch darauf zugegriffen werden.
- 2 Ein besonderes Merkmal des Tools ist, dass jeder Betrieb anders als etwa bei "ChatGPT" oder "Copilot" selbst festlegen kann, welches Sprachmodell von welchem Hersteller benutzt werden soll. Es werden sowohl gängige Sprachmodelle etwa von OpenAI, Microsoft und Google in der Cloud unterstützt wie auch auf eigenen Servern betriebene Open-Source-Modelle. So kann **kontrolliert werden, was mit den Daten geschieht** und ob die Daten das eigene Haus verlassen; auch sonst kann das Tools sehr detailliert auf die eigenen Bedürfnisse hin konfiguriert werden. Wir haben keinen Zugriff auf die Daten anderer Unternehmen. Die Software ist zudem quelloffen und nutzt neben den Tools von Microsoft nur quoöffne Bibliotheken. Es ist also völlig transparent, was das Add-in mit den Daten tut. Jeder Betrieb hat die volle Kontrolle über seine Inhalte und kann trotzdem allen Mitarbeitenden erlauben, einen "intelligenten" Agenten im Alltag als Helfer hier und dort einzusetzen.
 - 3 Die Add-ins sind als COM-Add-in für Word, Excel und (das klassische, nicht das neue) Outlook programmiert (in VB.net) und existieren daher nur für **Windows**. Die installationsbereiten Add-ins sind zur Sicherheit digital von VISCHER signiert (zu den weiteren Sicherheitsfunktionen siehe unten). Das gilt auch für die beiden (optionalen) Hilfs-Add-ins für Word und Excel, die in VBA programmiert sind (d.h. in der Makrosprache von Microsoft Office) und gewisse Dinge ermöglichen, die sonst nicht gehen (z.B. dass auch aus eigenen Excels auf die KI-Schnittstellen von Red Ink zugegriffen werden kann). Zwei weitere optionale Erweiterungen für den Edge- und Chrome-Browser sind in Javascript programmiert.
 - 4 Die Add-ins sind derzeit im Beta-Test und dürfen während diesem kostenlos genutzt werden. Auch danach sollen sie **für die persönliche Nutzung kostenlos** sein, dies entsprechend den noch zu publizierenden Lizenzbedingungen. Für die kommerzielle Nutzung sind je nach Fall ebenfalls kostenlose Lizenzen vorgesehen und solche für grössere Installationen gegen eine moderate Gebühr.
 - 5 Der Quellcode ist auf GitHub abrufbar (<https://github.com/VISCHER-LIL/redink>). Die Website <https://vischer.com/redink> enthält weitere Infos und einen Download-Link sowie ein **Demo-Video**.
 - 6 Trotz allem ist die Leistung von Red Ink **nur so gut und schnell, wie es auch das jeweils verwendete Sprachmodell ist**, denn dort werden die Ergebnisse produziert. Dabei haben wir durchaus erhebliche Unterschiede zwischen den Modellen festgestellt. Es gibt auch einige systembedingte Hürden, etwa, dass Sprachmodelle nicht darauf ausgelegt sind, Texte mit Formatierungen zu verarbeiten und Word und Outlook es in dieser Hinsicht auch nicht einfach machen, Umgehungslösungen zu finden. Wir haben verschiedenste Tricks und Methoden im-

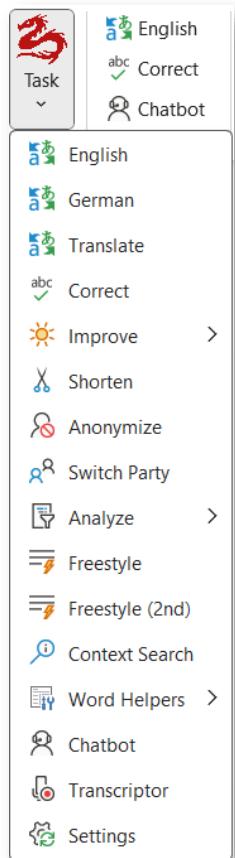
plementiert, um damit so gut wie möglich umzugehen – und wir hören auch, dass Red Ink das offenbar besser tut als einige andere, auch bekannte Tools.

- 7 Wir empfehlen daher, dass jeder und jede selbst herausfindet, wo und wie das Tool ihm und ihr am besten hilft. Am Ende hat es in einem Anhang einige **Ideen zum Ausprobieren** und das Demovideo hilft auch. Selbstverständlich gilt auch hier, dass die Ergebnisse der KI jeweils auf ihre Richtigkeit zu prüfen sind. Mängel und natürlich auch **Feature-Vorschläge** sind dem Autor david.rosenthal@vischer.com direkt zu melden.
- 8 Nachfolgend sind die Funktionen (Rz. 11 ff.), die Installation (Rz. 252 ff.), die Konfiguration (Rz. 309 ff.) und weitere Aspekte wie z.B. die Sicherheitsfunktionen beschrieben. Diese Anleitung gibt es auf Deutsch und Englisch. Einen garantierten Support bieten wir noch keinen an – wir sind nach wie vor eine Anwaltskanzlei und der Autor von Red Ink leitet hauptberuflich das Data & Privacy Team von VISCHER. Wir beraten gerne auch in Fragen des KI-Rechts, oder wenn ein Vertrag mit einem der Anbieter von Sprachmodellen datenschutz- und wie in unserem Fall sogar berufsgeheimniskonform abgeschlossen werden soll.
- 9 Wer, wie die meisten, kein Sprachmodell auf einem eigenen Server betreibt, muss ein solches für den Einsatz von Red Ink abonnieren, denn das Tool muss auf eine entsprechende sog. API konfiguriert werden. Bekannte Anbieter sind OpenAI, Microsoft und Google; wir nutzen für unsere Zwecke vor allem Google. Das ist zwar kostenpflichtig, aber die Kosten sind nach unserer Erfahrung sehr tief – viel tiefer, als wenn für jeden Mitarbeiter ein Abo für manche der bekannten Dienste gelöst wird. Zu erwähnen ist auch, dass die Nutzung von Red Ink im Betrieb kein Login benötigt und jedenfalls von Red Ink selbst nicht aufgezeichnet wird.
- 10 Die vorliegende Anleitung betrifft ausschliesslich **Generation 2** von Red Ink. Es sind dies die Versionen ab 31. Dezember 2024.

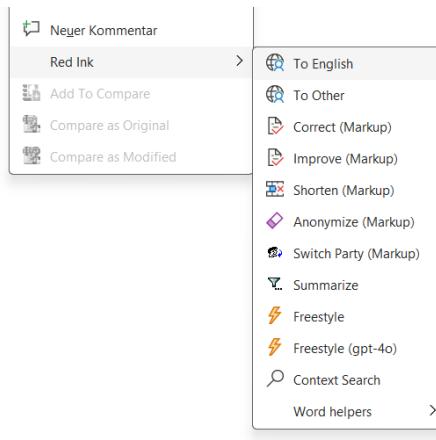
II. BENUTZUNG

A. Grundsätzliche Handhabung

- 11 Ein **Demovideo**, wie Red Ink benutzt werden kann, ist auf der Website <https://vischer.com/redink> abrufbar.
- 12 Im Kern funktionieren die Add-ins so, dass Text oder Zellen selektiert werden und dann die KI gebeten werden kann, damit etwas zu tun. Was genau, hängt davon ab, ob das Add-in in Word, Excel oder Outlook benutzt wird. Red Ink steht über zwei Kacheln zur Verfügung. Sie erscheinen in Word, Excel und Outlook in der Hauptanzeige und bei Outlook zusätzlich immer dann, wenn eine Mail zum Schreiben geöffnet ist (d.h. beim Erstellen, Antworten oder Weiterleiten oder beim Öffnen eines Entwurfs):

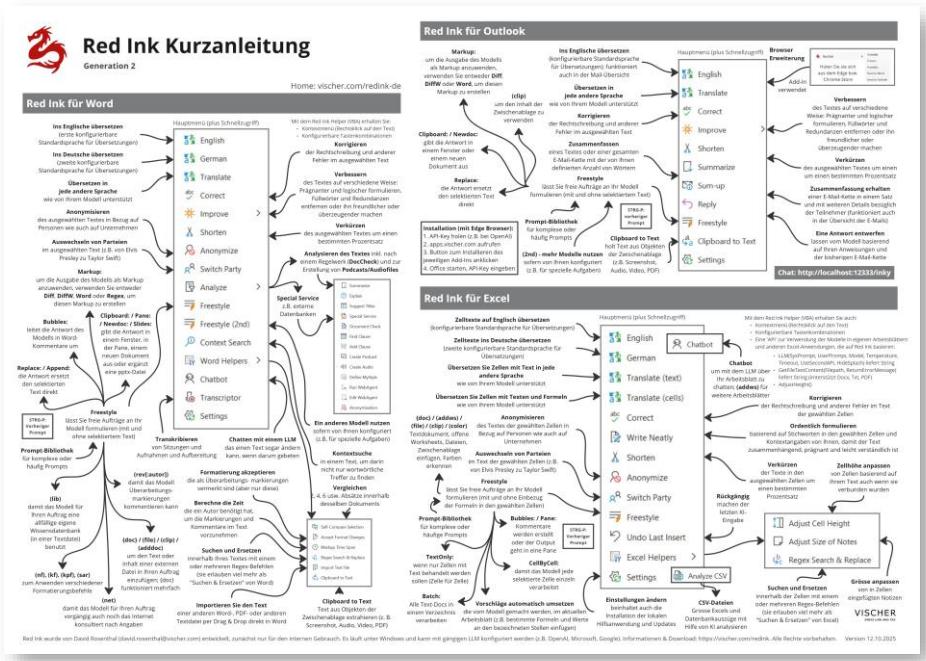


- 13 Eine der Kacheln dient mit den (je Office-Anwendung) häufigsten drei Funktionen für den Schnellzugriff (die auf dem Knopf angegebene Sprache für den Schnellzugriff auf die Übersetzungsfunktion kann geändert werden). Die andere gibt den Zugang zu den Funktionen, sobald das Logo angeklickt wird (bleibt die Maus darauf, wird die aktuelle Version und das aktuell konfigurierte Sprachmodell angezeigt). Die Kacheln können in den jeweiligen Anwendungen umpositioniert werden (soweit der Systemadministrator das zulässt).
- 14 In Word und Excel können die Funktionen zusätzlich über das Kontextmenü und bestimmte Tastenkombinationen ausgewählt werden. Das Kontextmenü erscheint, wenn nach selektiertem Text oder selektierten Zellen die rechte Maustaste geklickt wird:



- 15 Die Tastenkombinationen können in den Konfigurationseinstellungen programmiert werden. Die Kontextmenüs und Tastenkonfigurationen erfordern allerdings systembedingt, dass bei der Installation zwei Hilfsprogramme (VBA-Add-ins) installiert werden (siehe Rz. 291 ff. unten). Red Ink läuft aber auch ohne sie, aber wenn sie fehlen, erscheint das Kontextmenü einfach nicht. Sind Tastenkombinationen definiert, werden sie angezeigt, wenn mit der Maus über den Menüeintrag gefahren wird.
- 16 Je nach Bedarf kann aber auch ohne Auswahl eines Textes direkt auf die KI zugegriffen werden, z.B. um einen vom Text unabhängigen Prompt einzugeben oder gleich eine ganze E-Mail zusammenfassen zu lassen. Auf diese Weise muss beim Schreiben nicht mehr in einen separaten Chat gewechselt werden, und das Ergebnis kann direkt weiterverwendet werden.
- 17 Die Ausgabe erfolgt normalerweise im aktuellen Dokument, Arbeitsblatt oder in der aktuellen Mail. Gewisse Funktionen (Freestyle) lassen eine Ausgabe auch in einem separaten Fenster und in die Zwischenablage zu. Im Add-in für Word gibt es auch einen integrierten Chatbot mit eigenem Fenster.
- 18 Alle drei Add-ins bieten eine Reihe von Funktionen für vordefinierte Aufgaben (z.B. Übersetzen, Korrigieren), können aber auch über die sog. Freestyle-Funktion auch mit eigenen Instruktionen beauftragt werden. Die Freestyle-Funktion verfügt vor allem in Word über zahlreiche Optionen (wie z.B. das Kommentieren von Markups oder Einbeziehen externer Dokumente), die manche andere Tools nicht bieten. Ferner kann in Red Ink eine eigene Prompt-Bibliothek benutzt werden. Alle von Red Ink verwendeten Prompts können übrigens über die Konfigurationsdatei geändert und den eigenen Bedürfnissen angepasst werden.
- 19 Für Chromium-basierte Web-Browser (z.B. Edge, Chrome) gibt es ferner eine Extension, mit welcher auch aus dem Browser überall dort, wo Texte bearbeitet oder selektiert werden können, diese direkt an das Add-in in Outlook gesendet und dort verarbeitet werden kann (z.B. für eine Übersetzung oder Korrektur).

- 20 Red Ink selbst ist nur auf Englisch verfügbar, aber kann Texte in allen Sprachen verarbeiten, die das jeweilige Sprachmodell unterstützt. Die eingebauten Prompts sind auch auf Englisch verfasst, können aber geändert werden (normalerweise ist das nicht nötig).
- 21 Wer nicht die ganze Anleitung lesen will, dem hilft vielleicht auch diese Kurzübersicht (sie ist zum Ausdrucken im Installationspaket enthalten und unter <https://vischerlnk.com/redink-ka> abrufbar):

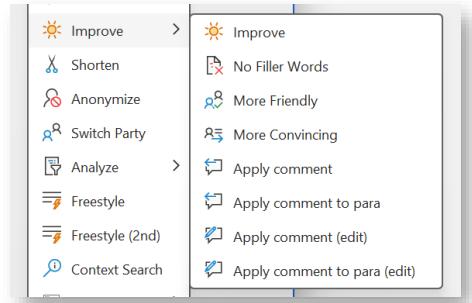


B. Grundfunktionen von Red Ink in Word

- 22 Die vordefinierten KI-Funktionen sind:

- To English, To German, To Other:** Der selektierte Text wird übersetzt und mit der Übersetzung ersetzt. To Other kann für eine Übersetzung in zahlreiche weitere Sprachen benutzt werden. Sie muss bloss (auf Englisch) angegeben werden, z.B. "French" für Französisch und nicht "Französisch". Die beiden Sprachen "English" und "German" sind in den Menüs vorkonfiguriert, können aber geändert werden. Zwei Hinweise: Die KI ist angewiesen, nach Satzzeichen keine doppelten Leerschläge zu machen (wie dies in bestimmten Sprachen früher noch üblich war). Auch ist sie angewiesen, bei Übersetzungen, bei denen der formelle und informelle Stil verschieden sind, diesen Stil zu wahren und im Zweifel den formellen Stil zu verwenden. Das englische "you" wird auf dieser Basis auf "Sie" übersetzt und nicht auf "Du", es sei denn, der Text enthält Hinweise auf eine informelle Sprache wie z.B. Anreden oder Grüsse mit Vornamen.

- **Correct:** Der selektierte Text wird sprachlich korrigiert, d.h. nicht nur Schreibfehler, sondern auch andere Fehler, wie beispielsweise nicht passende Wörter oder falsche Satzzeichen.
- **Improve:** Hier wird der Text auch inhaltlich lektoriert und verbessert, aber ohne, dass neue Informationen hinzukommen. Es werden beispielsweise Anpassungen zur besseren Verständlichkeit vorgeschlagen. Nebst Improve bietet dasselbe Menü auch noch die Varianten **No**



Filler Words zum Entfernen von Füllwörtern und Redundanzen, **More Friendly**, um den selektierten Text freundlicher zu formulieren, und **More Convincing**, um ihn überzeugender zu gestalten. Schliesslich kann, falls eine MyStyle Prompt-Datei konfiguriert worden ist, auch die Funktion **Apply MyStyle** angewählt werden (dazu Rz. 48 ff.).



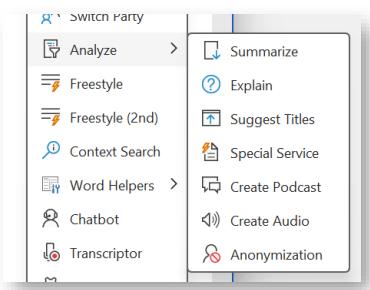
Eine besondere Funktion ist **Apply**

Comment in seinen Ausprägungen. Sie kann den Inhalt eines Word-Kommentars mit KI-Hilfe im Dokument umsetzen, also z.B. eine Korrektur vornehmen oder eine im Kommentar umschriebene Ergänzung umsetzen. Hierzu muss die Kommentar-Blase angewählt werden. Alle vier Varianten setzen den ganzen Kommentar (oder, falls angewählt, den selektierten Text darin) um und zwar entweder auf die vom Kommentar markierte Stelle oder den ganzen Absatz (bzw. die ganzen Absätze), in welchem sich der Kommentar befindet ("to para"). Die Variante "(edit)" erlaubt es dem Benutzer, den Prompt zum Einfügen des vorab zu bearbeiten. Diese Funktion kann mit dem Präfix "Bubbles:" von Freestyle kombiniert werden, sodass die KI Ihr Dokument kommentieren und die Kommentare dann mit dieser Funktion implementieren kann.

- **Shorten:** Mit dieser Funktion kann ein Text gekürzt werden, möglichst ohne dass Informationen verloren gehen, oder nur jene, die als weniger wichtig erachtet werden. Der Benutzer kann angeben, um wieviel Prozent der Text gekürzt werden soll; die KI hält sich allerdings unter Umständen nicht wirklich strikt an solche Längenangaben.
- **Anonymize:** Der selektierte Text wird in Bezug auf natürliche Personen, aber auch Unternehmen anonymisiert. An den betreffenden Stellen wird ein "[redacted]" eingeführt. Red Ink wird die Regex-Markup-Methode vorschlagen (siehe unten), die für grös-

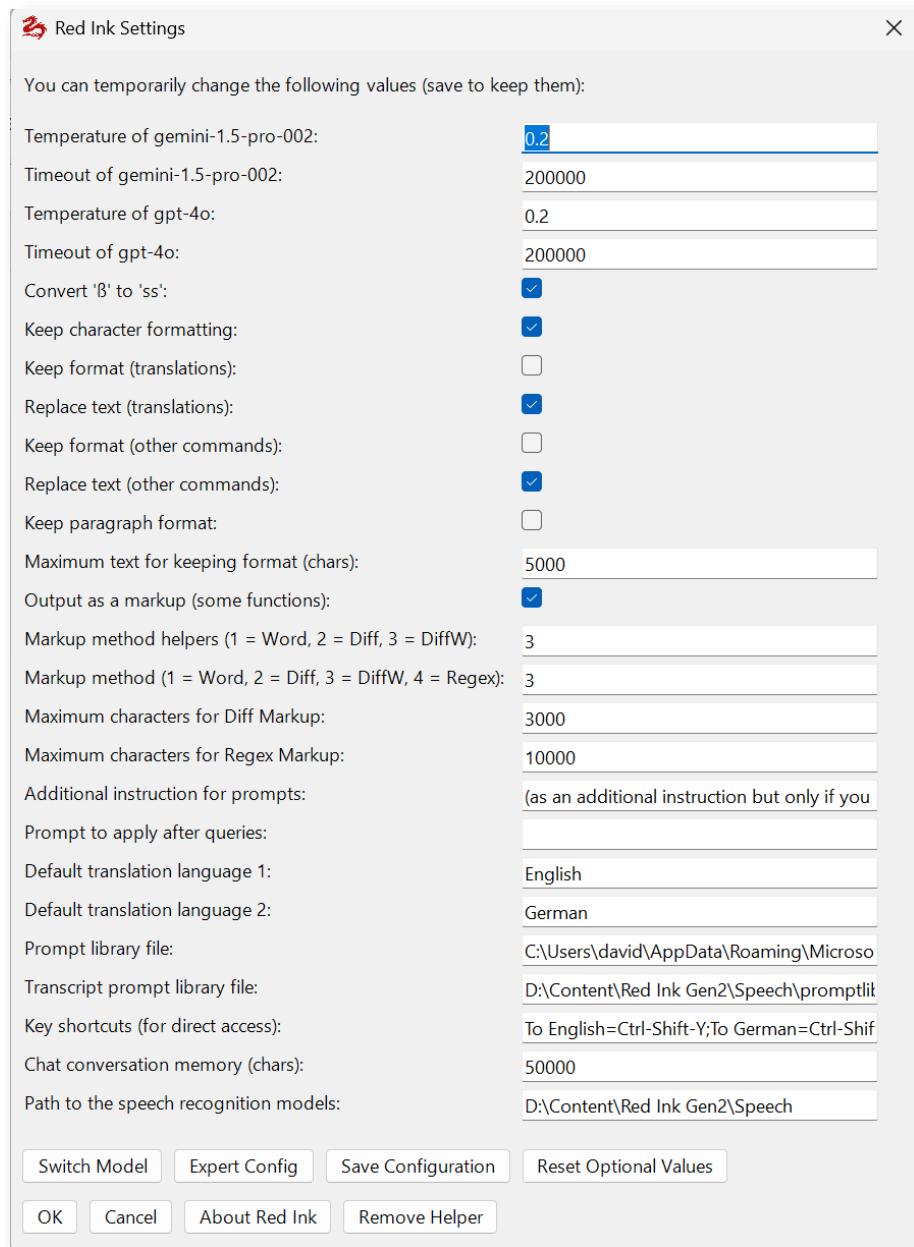
sere Texte besser geeignet sein könnte, falls eine andere gewählt ist.

- **Switch Parties:** Mit dieser Funktion können in Verträgen, Rechtsschriften und anderen Texten Verweise auf bestimmte Personen "intelligent" ersetzt werden. Aus dem "dem Provider" kann zum Beispiel "die Anbieterin" gemacht werden, wobei jeweils die gesamte Formulierung berücksichtigt und angepasst wird. Dies ist mit einem normalen "Suchen-und-Ersetzen" nicht möglich. Hier kann es sinnvoll sein, die Markup-Methode "Regex" einzusetzen. Red Ink wird dies auch vorschlagen, sofern dies nicht schon so aktiviert ist.
- **Analyze:** Hier sind verschiedene Kommandos zusammengefasst, welche den selektierten Text auf die eine oder andere Art analysieren und verarbeiten. **Summarize** fasst den Text zusammen. Dies erlaubt einen raschen Überblick. Die Zusammenfassung wird am Ende des Textes eingefügt. Es kann angegeben werden, wieviele Wörter die Zusammenfassung haben soll. Wird **Explain** gewählt, liefert das Tool ebenfalls eine kurze Zusammenfassung, aber geht spezifischer auf die Dinge ein, die gemäss dem Text zu tun sind, auf die Argumente und Logik des Autors oder der Autorin, und er liefert Erläuterungen zu den fachspezifischen Ausführungen, die das Modell selbst kennt (dem Modell wird von Red Ink ferner das aktuelle Datum als Referenz mitgegeben). **Suggest Titles** schlägt für den selektierten Text jeweils Titel vor, und zwar jeweils drei verschiedene Titel für unterschiedliche Anwendungsfälle (Memo, Blog, informeller Text, humorvoller Text, tiefgründiger Text). Bei diesen beiden letzten Funktionen wird der Text separat angezeigt, nicht eingefügt (kann aber bearbeitet und so in die Zwischenablage übernommen werden). Über das Menü können auch **Podcasts** und **Audioaufnahmen** von Texten generiert werden. Mehr dazu in Rz. 131 ff. unten. Ebenso kann auf **Special Service** zugegriffen werden, falls solche konfiguriert sind. Mehr dazu in Rz. 68 ff. unten. Schliesslich kann mit **Anonymization** die eingebaute Anonymisierungsfunktion getestet werden, die für das transparente Anonymisieren von Texten benutzt werden kann, die an die Sprachmodelle übermittelt werden (siehe dazu Rz. 149 ff.).



- 23 Wenn Sie eine von Red Ink vorgenommene Einfügung oder Änderung **rückgängig machen** möchten, können Sie die Rückgängig- bzw. Undo-Funktion von Word verwenden, müssen diese aber mehrfach ausüben, da Red Ink Text in mehreren Schritten ersetzt und einfügt.

- 24 Wie diese Funktionen mit dem Text umgehen, d.h. ob sie ihn ersetzen oder ob der Output angehängt wird, ob ein Markup (Vergleichsversion) erstellt wird und mit welcher Methode und ob und wie versucht werden soll, die bestehenden Formatierungen zu erhalten, kann über die Konfigurationsdatei oder die **Settings**-Funktion gesteuert werden. Sie ist über das Kachelmenü abrufbar und kann benutzt werden, um die Änderungen temporär vorzunehmen oder für künftige Sitzungen in der Konfigurationsdatei zu speichern (eine Erklärung jeder Option wird angezeigt, wenn mit der Maus über den Text gefahren wird):



- 25 Im Einzelnen:

- **Keep character formatting** sagt dem Add-in, dass er der KI die gebräuchlichsten Wort-Formatierungen wie fett, kursiv und unterstrichen der KI in einem ihr bekannten Format (Markdown)

übergeben soll, damit auch der Output entsprechend formatiert ist. Dies funktioniert allerdings nicht immer, und in Kombination mit Markups ist diese Funktion nur bei DiffW aktiv. In der Standardeinstellung ist diese Funktion eingeschaltet. Diese Funktion ist zudem auf Anzahl Zeichen für den Diff-Markup beschränkt, weil sonst das Programm zu lange braucht. Achtung: Die Formatierungen können trotzdem verschwinden, nämlich wenn sie von der aktuellen Absatzformatierung überschrieben werden, die Red Ink ebenfalls zu erhalten versucht. Achtung: Die Funktion kann bei Dokumenten, die Tabellen enthalten, zu längeren Wartezeiten führen.

- **Keep format** sagt dem Add-in, dass es der KI nicht nur den selektierten Text übermitteln soll, sondern darin (vorübergehend in HTML) hinterlegt auch die grundlegenden Formatierungen (wie z.B. Fettdruck oder eine Liste). Wenn als der Text "Wir haben viel Spass" übersetzt wird, wird die KI "We are having *a lot of fun*" liefern, wenn sie den Instruktionen folgt. Zu beachten ist allerdings, dass diese Funktionalität mit vielen zusätzlichen Daten verbunden ist und sich daher nicht für grosse Texte eignet (ggf. warnt das Add-in), weil die KI damit überfordert sein kann und es viel Zeit braucht. Um den Aufwand (und damit die Wartezeit) in Grenzen zu halten, werden nicht sämtliche Formatierungen erhalten. In der Funktion Freestyle kann Keep format über das eingefügte Kürzel "(kf)" aktiviert werden.
- **Keep paragraph format** geht etwas weniger weit als Keep format, kann aber schon helfen, die Formatierungen des bestehenden Textes aufrecht zu erhalten. Auch hier werden in den Text Formatierungen vorübergehend hinterlegt, aber nur jeweils die Absatzformatierungen, also viel weniger. Darum geht das auch schneller als die vorherige Option. Das ist vor allem bei jenen Texten oftmals genügend, in denen viel mit Formatvorlagen gearbeitet wird. Wird dies nicht gewählt, versucht das Add-in sich jedenfalls bei nicht sehr langen Texten die Absatzformatierungen trotzdem zu merken, aber das ist weniger zuverlässig, weil der Output möglicherweise nicht dieselben Absätze hat (schon eine zusätzliche Absatzmarkierung bringt das Konzept durcheinander). Ist "Keep format" gewählt, hat dies Vorrang.
- Nebst der Speicherung von Absatzformatierungen wird das Add-in für Word auch versuchen, **Fussnoten**, **Endnoten** und **dynamische Felder** (z.B. Querverweise, Datumsfelder) im Text zu speichern, der an die KI geht, und sie nachträglich wieder einzufügen (z.B. als übersetzte Fussnote). Gewisse weitere Angaben wie z.B. Tabellen, Bilder oder auch Kommentare gehen allerdings verloren. Sie werden nicht "zwischengespeichert" und müssen daher vorher gesichert werden. Das geschieht auch dann, wenn "Keep paragraph format" nicht gewählt ist, dies geschieht jedoch

nur, wenn der bestehende Text ersetzt wird, aber nicht, wenn er angehängt wird (kann in Freestyle mit "(sar)" übersteuert werden).

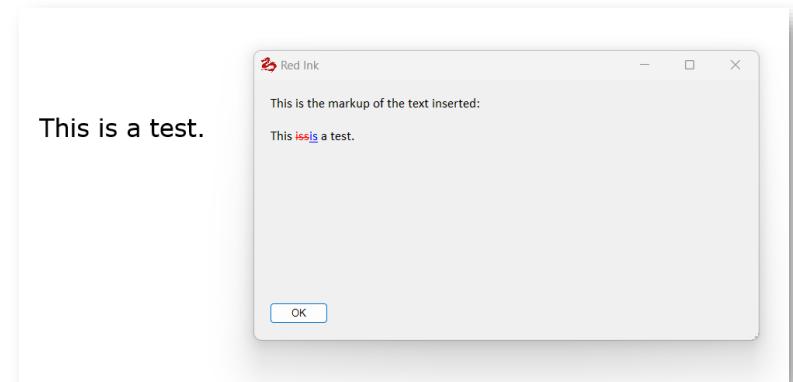
In der Funktion Freestyle kann "Keep paragraph format" über das eingefügte Kürzel "(kpf)" aktiviert werden.

Wir empfehlen, es zunächst auch ohne "Keep paragraph format" zu versuchen. In den meisten Fällen genügt das.

- Mit **Maximum text for keeping format** kann ein Wert (Zeichenzahl, z.B. 10'000) gesetzt werden, ab welchem das Add-in die Befehle zum Merken des Formats nicht mehr berücksichtigt, um zu lange Wartezeiten zu vermeiden. Wird der Wert auf 0 gesetzt, dann ist diese Sicherheitsfunktion deaktiviert. Diese Sicherheitsfunktion bietet insofern auch Komfort, vor allem beim Einsatz von Freestyle (siehe Rz. 33 ff. unten), da Red Ink erfahrungsgemäss bei grösseren Texten nicht für direkte Anpassungen am Text eingesetzt wird, sondern für andere Funktionen, die weniger Zeit brauchen. Da ist es sinnvoll, dass das Add-in die zeitraubenden Funktionen zur Hinterlegung und Verarbeitung der Formatierungsangaben in diesen Fällen automatisch ausschaltet, insbesondere dann, wenn ein Text nur befragt wird oder als Ausgangsbasis für eine Anfrage dient, bei der es nicht auf seine Original-Formatierung ankommt. In Freestyle kann hierfür der Trigger "(noformat)" oder "(nf)" benutzt werden. Wer Formatierungsfunktionen vorübergehend oder ganz ausschalten will, kann den Wert auch sonst auf 1 setzen.
- "**Replace text**" sagt dem Add-in, dass es den Output der KI, also z.B. den übersetzten Text, an Stelle des selektierten Textes einfügen soll. Das kann für Übersetzungen und anderen Funktionen getrennt konfiguriert werden. Bei Übersetzungen wird meist Replace verwendet, bei Korrekturen und anderen Funktionen ist es eine Geschmacksfrage (wir setzen Replace auch für Korrekturen ein).
- "**Do markup**" sagt dem Add-in, dass es bei Funktionen, wo das Sinn macht (z.B. bei Korrekturen, nicht aber Übersetzungen), einen Markup erstellen soll zum selektieren Text. Falls diese Funktion aktiviert ist, ist dies im Kontextmenü ersichtlich (wie angezeigt in Rz. 14 oben). Die Erstellung von Markups ist technisch innerhalb von Office nicht ganz trivial. Wir stellen darum vier verschiedene **Markup-Methoden** zur Verfügung (anzugeben ist jeweils der Wert 1, 2, 3 oder 4):
 - Ist **Word** (Wert 1) gewählt, dann wird die Word-interne Vergleichsfunktion verwendet. Diese funktioniert so, dass der selektierte Text und der neue Text des Add-in in zwei temporäre Dokumente kopiert wird und Word dann davon ein drittes temporäres Dokument erstellt. Dessen Inhalt

wird dann mit den Markups in das Hauptdokument eingefügt. Das geht alles automatisch, aber es ist am Bildschirm zu sehen, was verwirren kann. Dies lässt sich technisch nicht unterdrücken und es kann beim Einsatz von Fremd-Add-ins zu Störungen führen. Das Dokumentenverwaltungssystem "iManage" (welches wir einsetzen) hält sich z.B. nicht die Office-Vorgaben und blockiert die automatische Schliessung der temporären Dokumente und fragt den Benutzer, ob die Dateien gespeichert werden sollen (was nicht der Fall ist); leider konnten wir den Hersteller bisher nicht dazu bewegen, dieses falsche Verhalten seines Add-ins zu korrigieren.

- Wird **Diff** (Wert 2) gewählt, wird der Markup mit einem einfachen Diff-Algorithmus erstellt, der die Texte Wort für Wort vergleicht. Dies weist nicht die Probleme der Word-Vergleichsfunktion auf, ist aber weniger zuverlässig und ist bei langen Texten zu langsam. Darum kann eine maximale Zeichenzahl konfiguriert werden, ab welcher das Add-in fragt, ob die Methode wirklich angewandt werden soll. Zusätzlich kann die Ausgabe, falls sie zu lange dauert, durch Drücken der Taste "**Esc**" abgebrochen werden.
- Wird **DiffW** (Wert 3) gewählt, wird der Markup mit demselben Diff-Algorithmus erstellt, der die Texte Wort für Wort vergleicht. Im Unterschied zu Diff (Wert 2) wird die Vergleichsversion in einem Fenster (W für "Windows") angezeigt, was sehr viel schneller geht als das normale Diff. Das Fenster bleibt so lange geöffnet, bis es mit der OK-Taste geschlossen wird. Es kann also gleichzeitig am neuen Text gearbeitet werden. Dies ist die Standardeinstellung.



- Selbst erfunden haben wir schliesslich die **Regex**-Technik (Wert 4), die so funktioniert, dass der geänderte Text zuerst von der KI mit dem Originaltext verglichen wird. Sie verfasst dann eine Beschreibung aller Änderungen (ggf. mit etwas Kontext), die dann durch eine Suchen-Ersetzen-Funktion umgesetzt wird (ursprünglich wurde hierfür eine

Methode benutzt, die unter dem Namen Regex für "Regular Expressions" bekannt ist). Wie gut dies funktioniert, hängt stark vom verwendeten Sprachmodell ab. Auch hier kann eine maximale Zeichenzahl konfiguriert werden, die aber typischerweise viel höher ist als bei der Diff-Methode. Diese Methode eignet sich für punktuelle Anpassungen in Texten und kommt auch mit etwas grösseren Texten zurecht. Sie setzt allerdings ein leistungsfähiges Modell voraus und braucht mehr Zeit, weil nach jeder Abfrage für die Vorbereitung des Compare eine zweite Abfrage erfolgt.

- Für Benutzer in der Schweiz und andere, welche mit dem "**scharfen S**" ("β") nichts anfangen können, wie es manche Sprachmodelle bei deutschen Texten liefern, kann das Add-in so konfiguriert werden, dass es dieses automatisch mit einem Doppel-S ersetzt.

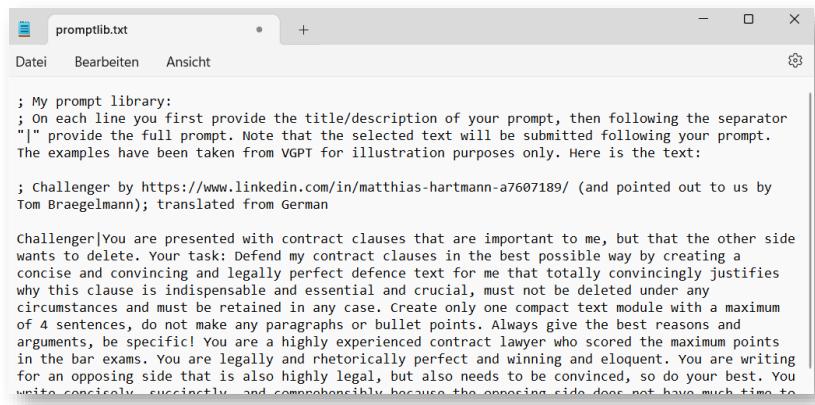
Wir empfehlen insbesondere bei Übersetzungen den Einsatz von Replace text, bei den anderen Funktionen aber ebenfalls. Für den Anfang empfehlen wir weiter, auf Keep format zu verzichten und ebenfalls auf Keep paragraph format um zu prüfen, ob die fix aktivierte Funktionen zum Erhalt von Absatzformatierungen genügen. Keep character formatting würden wir einsetzen. Genügt dies nicht, kann Keep paragraph format hinzugewählt werden. Bei langen Texten empfehlen wird ein schrittweises Vorgehen (d.h. immer nur einige Absätze bearbeiten lassen); hierzu kann bei Freestyle auch der "(iterate)"-Trigger benutzt werden. Alternativ ist auch die Kommentar-Funktion von Freestyle ("Bubbles:") benutzt werden, um grössere Texte zu bearbeiten (siehe Rz. 33 ff. unten).

26 Weitere Konfigurationen, die über das Settings-Menü vorgenommen werden können sind insbesondere:

- Die **Temperatur** gibt an, wie kreativ das Sprachmodell sein soll, wenn es Antworten liefert. Für Aufgaben wie Übersetzungen oder Korrekturen empfehlen wir einen tiefen Wert (z.B. 0.2), für freiere Aufgaben wie das Finden von Argumenten einen höheren (z.B. 0.8). Nicht alle Modelle unterstützen die Angabe einer Temperatur. Sollen andere Parameter konfiguriert werden, ist dies über die Konfigurationsdatei des Add-in möglich.
- **Timeout** bestimmt, wie lange auf eine Antwort des Sprachmodells gewartet werden soll. Gerade komplexe Aufgaben benötigen mitunter einige Zeit, und diese kann dem Modell hiermit gegeben werden. Die Angabe ist in Millisekunden. Kommt es in der normalen Verwendung zu einem Timeout-Fehler kann versucht werden, den Wert hochzusetzen.
- Die Add-ins können auf **mehrere Sprachmodelle** gleichzeitig fix konfiguriert werden, ein primäres, ein sekundäres und ggf. noch beliebig viele weitere Modell (sie können aber nicht über Settings

konfiguriert werden, dazu Rz. 331 ff.). Hier sind die Werte für beide angegeben. Mit Switch Model können die beiden (falls definiert) gewechselt werden. Ansonsten kann auf das sekundäre Modell über Freestyle direkt zugegriffen werden. Dies kann z.B. benutzt werden, um Modelle mit besserer Problemlösungsfähigkeit, die aber langsamer sind, griffbereit zu hinterlegen, auf sie aber nur bei Bedarf zuzugreifen.

- Es können auch **zwei Zusatzprompts** konfiguriert werden. Der erste wird bei jeder vordefinierten Funktion (z.B. Übersetzen oder Kürzen) dem Sprachmodell zusätzlich mitgegeben, ausser beim Transcriptor und Chat. Das kann benutzt werden, um bestimmte sprachliche Probleme, die regelmässig auftauchen, zu lösen, z.B. wenn sich das Sprachmodell nicht an die Sprache hält oder bestimmte Begriffe jeweils anders geschrieben werden sollten. Wird der zweite Zusatzprompt auch ausgefüllt, wird das Ergebnis der Abfrage mit diesem Prompt jeweils separat nachbearbeitet. Das kann wirksamer sein, braucht aber viel mehr Zeit. Wir empfehlen, den zweiten Zusatzprompt nur ausnahmsweise zu nutzen, und auch den ersten nur dann, wenn sich der Bedarf zeigen sollte.
- Es können zwei **Standardsprachen** angegeben werden, für die im Menü eigene Kurzwahl-Schaltflächen erscheinen. Standardmäßig sind dies Englisch und Deutsch. Die Sprache ist in Englisch anzugeben.
- Es kann der Pfad und der Name einer eigenen **Prompt-Bibliothek** angegeben werden. Sie steht in Freestyle zur Verfügung (siehe Rz. 33 ff. unten). Es muss sich um eine Text-Datei handeln, und jeder Prompt ist in einem bestimmten Format (eine Kurzbezeichnung bzw. Titel, dann ohne Leerschlag "|" und dann der Prompt) auf jeweils einer eigenen Zeile zu erfassen. Leerzeilen stören nicht, Kommentare mit vorangestelltem ";" werden ignoriert. Eine Muster-Prompt-Bibliothek wird im Installationspaket bereitgestellt (in der aktuellsten Version jeweils auf <https://apps.vischer.com>). Sie kann in den Add-ins geändert werden (siehe zur Darstellung in Rz. 36 unten, weitere Details in Rz. 324 ff. unten).



```

promptlib.txt

; My prompt library:
; On each line you first provide the title/description of your prompt, then following the separator
;" provide the full prompt. Note that the selected text will be submitted following your prompt.
The examples have been taken from VGP for illustration purposes only. Here is the text:

; Challenger by https://www.linkedin.com/in/matthias-hartmann-a7607189/ (and pointed out to us by
Tom Braegelmann); translated from German

Challenger|You are presented with contract clauses that are important to me, but that the other side
wants to delete. Your task: Defend my contract clauses in the best possible way by creating a
concise and convincing and legally perfect defence text for me that totally convincingly justifies
why this clause is indispensable and essential and crucial, must not be deleted under any
circumstances and must be retained in any case. Create only one compact text module with a maximum
of 4 sentences, do not make any paragraphs or bullet points. Always give the best reasons and
arguments, be specific! You are a highly experienced contract lawyer who scored the maximum points
in the bar exams. You are legally and rhetorically perfect and winning and eloquent. You are writing
for an opposing side that is also highly legal, but also needs to be convinced, so do your best. You
write concisely, succinctly, and comprehensibly because the opposing side does not have much time to

```

- Es können die **Tastenkombinationen** definiert werden, mit denen sich die Funktionen in Word aufrufen lassen. Hierzu ist der Menüpunkt anzugeben (genau so, wie es im Kontextmenü steht, z.B. "Correct"), dann ein "=" und dann ohne Leerschlag den Tastengriff (z.B. "Ctrl-Alt-C"). Mehrere Shortcuts sind durch ";" zu trennen. Sie gelten gleichzeitig auch für das Excel Add-in. Auf den Zusatz "(Markup)" kommt es allerdings nicht an. Gewisse Tastenkombinationen sind allerdings schon besetzt und funktionieren daher nicht. Die Funktion setzt auch voraus, dass die Helper-Zusatzdatei in Word installiert ist (mit VBA-Code, was in gewissen Umgebungen allerdings gesperrt ist, siehe unten, Rz. 291 ff. unten) und das Kontextmenü aktiviert ist (was sich auch konfigurieren lässt). Die Tastenkombinationen können auch in Word direkt bearbeitet werden; sie bleiben dort gespeichert. Technisch funktionieren sie so, dass ein Makro in der zusätzlich installierten Datei, die bei jedem Start von Word geladen wird, aufgerufen wird, die wiederum den Code im Add-in aufruft. Das lässt sich leider anders nicht realisieren, da Shortcuts eine aus Sicht von Microsoft "veralte" Technik sind und in der modernen Oberfläche der Office-Produkte nicht vollständig unterstützt werden. Da sie jedoch sehr nützlich sein können, unterstützen wir sie auf diese Weise.
- Der Parameter **Chat conversation memory** gibt an, wie viele Zeichen des bisherigen Dialogs sich der Chatbot merken soll. Der jeweilige in Word bearbeitete Text wird in diesem Speicher jedoch nicht abgelegt, ausser der Chatbot zitiert solche Teile in seinem Dialog (wozu er angehalten ist, falls er sich etwas merken soll, z.B. wenn zwischen verschiedenen Dokumenten hin- und hergeschalten wird). Der Standardwert ist hier 50'000. Ist dieser Wert zu hoch, wird das Sprachmodell einen Teil der übermittelten Inhalte, insbesondere auch des Dokuments, des Benutzers nicht verarbeiten können.
- Es kann ferner angegeben werden, wo die für die **Spracherkennung** erforderlichen lokalen Modelle (und im Falle von Whisper

- die zusätzlichen Runtime-Bibliotheken) gespeichert sind. Mehr dazu in Rz. 83 ff. unten.
- 27 Über **Expert Config** können weitere Konfigurationswerte angeschaut oder geändert werden. Wir empfehlen dies allerdings nicht. Sind spezifische Konfigurationen erforderlich, geht dies einfacher und zuverlässiger über die manuelle Bearbeitung der Konfigurationsdatei (dies ist möglich über das Anklicken von "**Edit .ini Files**"; es kann dann gewählt werden, ob die Hauptkonfigurationsdatei oder auch allfällige Modell- oder Special-Services-Konfigurationsdateien bearbeitet werden und sie werden in einem einfachen Editor angezeigt; die dort allenfalls vorgenommenen und gespeicherten Änderungen werden aber erst aktiv, wenn sie wieder geladen werden). Die Konfigurationsdatei kann allerdings aus den Add-ins heraus aktualisiert werden. Ist die Verschlüsselung des API-Keys bzw. Private Keys aktiviert, wird dort nur der verschlüsselte Schlüssel angezeigt. Wird die Experten-Konfiguration mit OK beendet, wird die (lokale) Konfigurationsdatei aktualisiert oder neu geschrieben.
- 28 Die geänderte Konfiguration kann mit **Save Configuration** in eine (lokale) Konfigurationsdatei gespeichert bzw. ergänzt werden. Sonst geht die geänderte Konfiguration beim Verlassen von Word verloren.
- 29 **Reset Optional Values** dient schliesslich dazu, das jeweilige Add-in auf die Standard-Werte zurückzusetzen, wobei die für den minimalen Betrieb benötigten Werte (z.B. API-Key für die API) nicht zurückgesetzt werden. Die Funktion kann also gefahrlos benutzt werden. Je nach lokaler Konfiguration erscheint statt Reset Optional Values eine alternative Funktion, um auf eine zentrale Konfiguration "zurückzuschalten", also die eigenen Anpassungen zu verwerfen und wieder auf jene Werte zurückzugehen, welche die eigene Organisation standardmäßig bereitstellt.
- 30 Die Knöpfe **Install Helper** oder **Remove Helper** dienen dazu, das zusätzliche Helper-Programm zu installieren oder wieder zu entfernen; sie ermöglichen in Word und Excel das Kontextmenü und die Tastenkombinationen sowie in Excel das API. Ist der Helper installiert, kann er entfernt werden, ist er es nicht, erscheint der Knopf zum Installieren. Wird Install Helper geklickt, wird die aktuellste Helper-Datei von der Website <https://apps.vischer.com> heruntergeladen und im Verzeichnis abgelegt, in welchem Word (bzw. Excel) die VBA-Add-ins abgelegt und beim Starten automatisch lädt. Das Installieren oder Laufenlassen solcher Add-ins kann von Sicherheitsfunktionen im jeweiligen Betrieb blockiert werden; die Funktion sollte nur benutzt werden, wenn die internen Vorgaben den Einsatz solcher VBA-Add-ins erlauben. Alternativ ist auch die manuelle Installation möglich (dazu Rz. 291 ff. unten). Soll der Helper entfernt werden, versucht Red Ink ihn zu deaktivieren und die Datei zu löschen; das gelingt aber nicht immer; in diesem Fall muss das jeweilige Programm zuerst beendet und die angegebene Datei manuell gelöscht werden.

- 31 Im Settings-Menü können ferner noch Knöpfe zum Prüfen auf **Updates** erscheinen. Ob und wie sie funktionieren, hängt davon ab, ob die Anwendung via Internet (d.h. <https://apps.vischer.com>) oder von einer lokalen Quelle installiert worden ist.
- 32 Wird Settings mit OK beendet, dauert es 1-2 Sekunden, bis Red Ink sich neu konfiguriert hat.

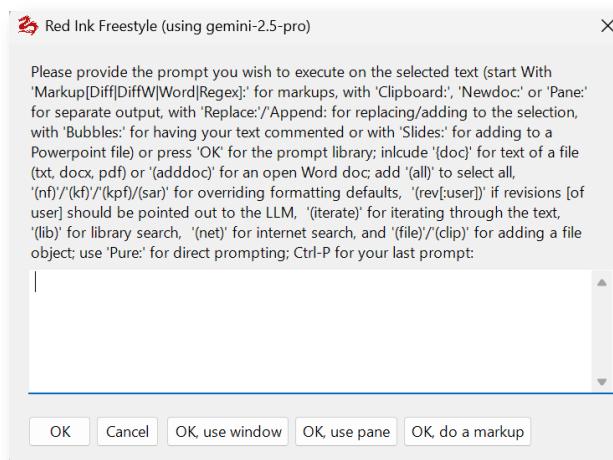
C. Freestyle-Funktion

- 33 Die Freestyle-Funktion erlaubt das **freie Promoten** mit einem Sprachmodell und bietet zahlreiche weitere Funktionen, die dabei helfen können und beim Studieren, Erfassen und Bearbeiten von Dokumenten hilfreich sein können. Sie ist daher ein sehr mächtiges und vielseitiges Instrument, das eine gewisse Übung erfordert, um es auszureißen. Es können damit **zum Beispiel:**
- Texte mit Kommentaren versehen werden, z.B. wo etwas verbessert werden könnte;
 - Texte befragt werden, z.B. wo sich bestimmte Inhalte befinden, die sich mit klassischen Suchwerkzeugen nicht finden lassen, oder welche Regelungen ein Vertrag zu einem bestimmten Thema enthält;
 - Texte umgestaltet werden, wo die vordefinierten Funktionen nicht ausreichen, z.B. nach bestimmten stilistischen Vorgaben (z.B. einen Text freundlicher oder bestimmter machen oder einen Text geschlechtsneutral umformulieren);
 - Texte mit bestimmten Inhalten ergänzt werden, z.B. ein Vertrag um eine bestimmte Klausel ergänzt werden, die in Stichworten vorgegeben wird ("Verfasse mir eine Klausel zur Vertragslaufzeit mit Mindestlaufzeit und monatlicher Kündigung und berücksichtige die bestehenden Regelungen." – wobei diesfalls der bestehende Vertrag selektiert sein muss, sonst greift das Add-in nicht darauf zu) oder aber nach Vorlage aus einer Klauselbibliothek;
 - Von einer anderen Person vorgenommene Markups von der KI zusammengefasst und beurteilt werden, um einen rascheren Überblick zu erhalten;
 - Extrakte aus einem Text erstellt werden, die in einer anderen Anwendung weiterverwendet werden können;
 - Informationen aus anderen Dokumenten von der KI in den eigenen Text situativ eingefügt werden oder ein neuer Text basierend auf Informationen aus einem anderen Dokument (auch PDF) erstellt werden lassen;
 - Informationen aus einem Text extrahiert und in einer besonderen Form dargestellt werden, z.B. für eine Tabelle mit zeitlichen Entwicklungen;

- Ideen zu einem Text von der KI formuliert werden, z.B. wie sich eine Vertragsklausel in einer Verhandlung besser verteidigen lässt;
- Ein Text, z.B. eine Rechtsschrift, von der KI kritisch begutachtet werden;
- Von der KI zwei Texte inhaltlich miteinander verglichen werden.

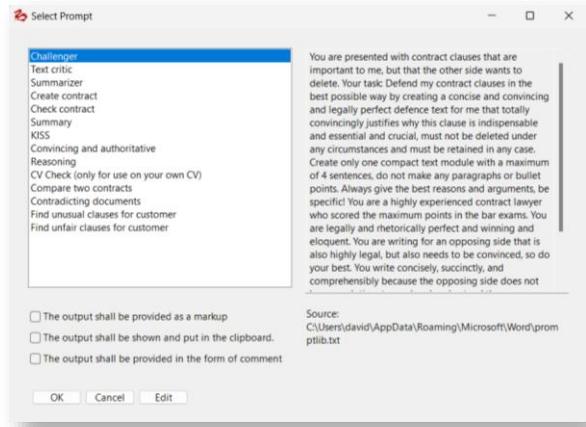
Wie gut diese Beispiele funktionieren, hängen nach unserer Erfahrung von den Fähigkeiten des Sprachmodells, der Grösse und den Formatierungen des Textes ab – und natürlich vom Prompt, der entweder eingegeben oder aus der Prompt-Bibliothek abgerufen wird.

- 34 Im Grundsatz kann mit der Funktion dem Sprachmodell ein beliebiger Befehl gegeben werden, **mit und ohne selektiertem Text**, wie dies auch in KI-Chatprogrammen wie "ChatGPT" oder "Copilot" möglich ist. Es gibt aber zwei wichtige Unterschiede: Freestyle merkt sich die bisherige Interaktion bewusst nicht, d.h. er nimmt den Text, wie er gerade ist und den aktuellen Befehl. Was vorher diskutiert worden ist, beeinflusst die Ausführung nicht. Der zweite wichtige Unterschied ist, dass die Resultate der KI direkt in Word weiterverarbeitet und auf den bestehenden Text angewandt werden können. Es muss nicht herumkopiert werden. Wer einen Chat benötigt, für den bietet das Tool einen solchen über einen anderen Befehl.
- 35 Wird Freestyle aufgerufen, kann in einem Fenster ein bei Bedarf auch **mehrzeiliger Prompt** erfasst werden. Der Prompt wird dann zusammen mit dem Text, auf den er angewandt wird, dem Sprachmodell übermittelt (mit einigen Begleitinstruktionen im Hintergrund, die allerdings auch einsehbar und änderbar sind).

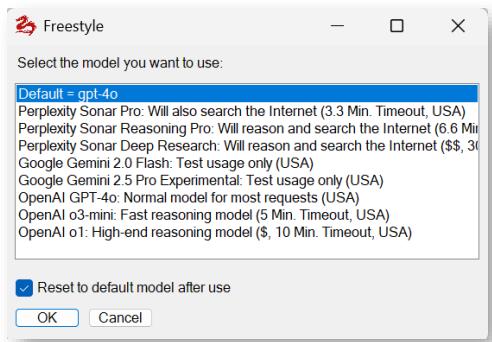


- 36 Red Ink merkt sich (auch wenn Word zwischenzeitlich geschlossen wurde) den letzten Prompt, der in Freestyle eingegeben wurde. Er kann dann mittels **Ctrl-P** eingefügt werden. Wer den Prompt unverändert nochmals ausführen will mit der aktuellen Selektion, kann auch **Freestyle (redo)** wählen.

- 37 Neben dem "OK"-Knopf bietet das Fenster auch die Möglichkeit, "OK"-Knöpfe mit Zusatzfunktionen zu benutzen. In diesen Fällen wird der dazu erforderliche Prefix (wie sogleich beschrieben) dem Prompt automatisch vorangestellt.
- 38 Wird kein Prompt eingegeben und nicht abgebrochen, erscheint – falls vorhanden – die **Prompt-Bibliothek**, aus welcher der gewünschte Prompt abgerufen werden kann. Sie eignet sich für komplexe Prompts, die immer wieder gebraucht werden. Wir haben einige Muster, auch von Dritten, darin hinterlegt. Die Prompt-Bibliothek lässt sich in Red Ink auch bearbeiten. Wir empfehlen daher, dass jeweils eine lokale, eigene Kopie der Prompt-Bibliothek benutzt wird (auch wenn es möglich wäre, diese zentral gespeichert zu halten). Die Prompt-Bibliothek wird bei jedem Aufruf neu geladen. Sie kann getrennt oder gemeinsam mit Excel und Outlook genutzt werden. Sie kann in einem Unternehmen auch zentral gespeichert werden, was aber die Gefahr birgt, dass ein unvorsichtiger Benutzer sie ungewollt verändert. Die Prompts können auch mit Platzhaltern für Benutzerparameter versehen werden. Weitere Details in Rz. 324 ff. unten.



- 39 Freestyle steht in Word sowohl für das **primäre Sprachmodell** zur Verfügung als auch für das **sekundäre**, falls ein solches konfiguriert worden ist. Sind zudem **weitere Modelle** konfiguriert (siehe Parameter "AlternateModelPath" und Rz. 331), kann vorgängig ausgewählt werden, welches dieser Modelle verwendet werden soll (wird die Checkbox abgewählt, dann bleibt das neu gewählte Modell bis zum Neustart von Word als sekundäres Sprachmodell aktiv und kann z.B. im Chatbot genutzt werden):



Auf das sekundäre und die weiteren Modelle wird über **Freestyle (2nd)** zugegriffen.

- 40 Über Prefixe im Prompt können diverse Ausgabeformate angesteuert werden (sie werden im Text des Prompt-Fensters zur Erinnerung erwähnt):

Please provide the prompt you wish to execute (with 'Clipboard:' for separate output or with 'Bubbles:' for having your text commented) or press 'OK' for the prompt library; include '{doc}' for text of a file (txt, docx, pdf); add '(rev[:user])' if revisions [of user] should be pointed out to the LLM, '(nf)'/'(kf)'/'(kpf)' for overriding formatting defaults, '(lib)' for library search, and '(net)' for internet search; use 'Pure:' for direct prompting; Ctrl-P for your last prompt:

- Freestyle kann gebeten werden, das Resultat als Markup zum selektierten Text auszugeben. So wird rascher ersichtlich, was sich geändert hat. Hierzu ist dem Prompt die Textfolge "**Markup:**" voranzustellen. Alternativ kann "**MarkupWord:**", "**MarkupDiff:**", "**MarkupDiffW:**" und "**MarkupRegex:**" verwendet werden, um eine bestimmte Markup-Methode anzuwenden (siehe Rz. 25 oben); sonst wird der für die anderen Funktionen eingestellte Standard verwendet.
- Wird "**Replace:**" verwendet, dann wird der selektierte Text durch den Output des Sprachmodells einfach (ohne Markup) ersetzt (z.B. "Replace: Formuliere mir diesen Satz etwas schmeichelhafter."). Wird hingegen "**Append:**" oder "**Add:**" benutzt, geschieht genau das umgekehrte: Der Output des Sprachmodells wird selbst bei anderer Standardeinstellung nach dem selektierten Text eingefügt. Feld-, Referenz- und Formatierungserhaltung sind standardmäßig deaktiviert (kann in Freestyle mit "(sar)" aufgehoben werden).
- Wer den Output der KI nicht im Dokument haben will, stellt bei seinem Befehl das Wort "**Clipboard:**" (oder "**Clip:**") voran. Der Output wird am Ende in einer Box angezeigt und kann dort bearbeitet werden. Der fertige Text (oder der Original-Text) kann dann in die Zwischenablage kopiert werden (ohne Formatierung-)

gen). Clipboard und Markup können nicht kombiniert werden, und Formatierungen werden in der Variante Clipboard auch nicht unterstützt (mit Ausnahme, dass in der Box Stellen, die von der KI als fett markiert sind, auch als solche angezeigt werden). Allerdings ist es möglich, den Text, den die KI geliefert hat (d.h. ohne Bearbeitungen des Benutzers), mitsamt den Formatierungen in Word einzufügen. Dafür ist ein eigener Button vorgesehen. Clipboard ist sehr praktisch, wenn eine Antwort von der KI gewünscht wird, die aber nicht im Text weiterverarbeitet werden soll. Die Weiterverarbeitung ist aber trotzdem möglich (über die Zwischenablage, die automatisch bedient wird).

- Statt in ein Fenster kann der Output auch in ein neues Word-Dokument eingefügt werden. In diesem Fall ist der Prefix "**Newdoc:**" zu verwenden.
- Wer den Output so angezeigt haben möchte, dass er am Dokument weiterarbeiten kann, sollte das Word "**Pane:**" seinem Prompt voranstellen (oder aber bei der Ausgabe via "Clipboard:" den Knopf "**Transfer to Pane**" klicken). Der Output wird dann in einen Fensterbereich umgelegt, der rechts vom Dokument aufgeht, d.h. in eine sog. *Pane*. Diese kann vergrössert und verkleinert oder sogar losgelöst werden. Das Ergebnis kann in der Pane selbst bearbeitet werden. Die Pane hat Buttons um den selektierten Text in die Zwischenablage einzufügen oder intelligent mit dem markierten Text im aktiven Dokument zusammengeführt werden ("**Merge Selection**"). Wird Merge Selection gewählt, dann geht ein Fenster auf und es wird ein Prompt angezeigt, der geändert werden kann, und der für das Zusammenführen sorgt. Wer die Original-Antwort der KI mit Formatierung in sein Dokument (der ein neues Dokument) einfügen will, kann das ebenfalls wählen (die Pane wird danach geschlossen) oder die Pane kann auch einfach so geschlossen werden. Die Pane merkt sich übrigens die Breite und wird bei der nächsten Verwendung mit derselben Breite starten.
- Es ist möglich, die Antwort der KI nicht in Form eines Textes im Dokument oder als dessen Markup auszugeben, sondern unter Verwendung der Kommentarfunktion ("Bubbles" für "Blasen"). In diesem Fall ist dem Befehl das Wort "**Bubbles:**" voranzustellen (z.B. "Bubbles: Gib mir alle Sätze an, die ich verbessern könnte, und sage wie."). Das Add-in wird dann den selektierten Text der KI übergeben und die Antwort in entsprechende Word-Kommentare der entsprechenden Textstellen darstellen. Dies hat den Vorteil, dass die Kommentare am Ende einfach wieder gelöscht werden können. Kann das Add-in eine Antwort der KI nicht zuordnen oder sonst nicht auswerten, gibt es sie am Ende des selektierten Textes aus. Die Kommentare sind an den Initialen "RI:" erkennbar; für die Kommentare wird bewusst der aktuelle Benutzer

zername benutzt, so dass der von der KI verfasste Kommentar bei Bedarf gleich als eigener Kommentar genutzt werden kann. Die Zuverlässigkeit dieser Funktion hängt von der Leistungsfähigkeit des Sprachmodells ab; wenn dieses die Anweisungen nicht korrekt befolgt, funktioniert auch diese Funktion nicht gut. Die Kommentarfunktion setzt ferner voraus, dass die von der KI angegebenen Textstellen im Dokument auch gefunden werden, was nicht immer ausnahmslos funktioniert, weil nicht sichtbare Zeichen oder Formatierungen im Dokument einen Treffer verhindern können; auch Markups können den Mechanismus stören. Dabei kann es trotz Gegenmassnahmen im Add-in auch dazu kommen, dass die kommentierten Stellen verschoben sind; sie sind also jeweils zu kontrollieren. Ist eine Zuordnung nicht möglich, zeigt das Add-in nach der Kommentierung in einem separaten Fenster an, was der KI im falschen Format geliefert wurde oder dem bestehenden Text nicht zugeordnet werden konnte; falls dies nicht abgebrochen wird, wird dieser Text am Ende des selektierten Textes in einem eigenen Kommentar eingefügt. Bubbles kann sowohl auf Textteile oder den gesamten Text angewandt werden (d.h. ohne vorher zu wählen).

- Red Ink kann auch eine bestehende Powerpoint-Datei um weitere Folien ergänzen mit dem Inhalt der KI. So kann die KI zum Beispiel gebeten werden, ein Memorandum oder ein Vertrag auf einigen Folien vorzustellen. Voraussetzung ist eine bereits bestehende Powerpoint-Datei im Format .pptx, in welcher sich auch jene Folienvorlagen bzw. eine Folie befindet, die das gewünschte Design aufweist, damit sich Red Ink bzw. die KI daran orientieren kann. Die Angaben zur bestehenden Präsentation (auch deren bisheriger Text-Inhalt) werden der KI übergeben, so dass auch darauf Bezug genommen werden kann in der Instruktion. Der Instruktion ist das Prefix "**Slides:**" voranzustellen (z.B. "Slides: Ergänze die bestehende Präsentation ab Folie 3 mit weiteren Ausführungen zum Vertrag und berücksichtige dabei das, was schon in der Präsentation steht."). Red Ink wird dann nach der Powerpoint-Datei fragen und sie anpassen (dazu muss diese geschlossen sein, sonst kommt es zu einem Fehler); wird der Vorgang abgebrochen, fragte Red Ink ob eine neue Präsentation erstellt werden soll und tut dies auf Wunsch auch (so kann ohne bestehende Vorlage eine Präsentation erstellt werden). Allenfalls ist es nötig, noch gewisse Anpassungen vorzunehmen oder das Folienlayout zurückzusetzen, falls die KI die Formatierungen nicht richtig getroffen hat. Wenn entsprechend instruiert, wird die KI auch grafische Elemente und Icons einbauen ("Füge in der Präsentation wo passend auch illustrierende Icons hinzu."). Es können jedoch keine Bilder mittels "Slides:" erzeugen. Auch bestehende Folien werden nicht angepasst (der Inhalt der eingefügten Folien wird aber darauf abgestimmt, wenn verlangt). Solche wei-

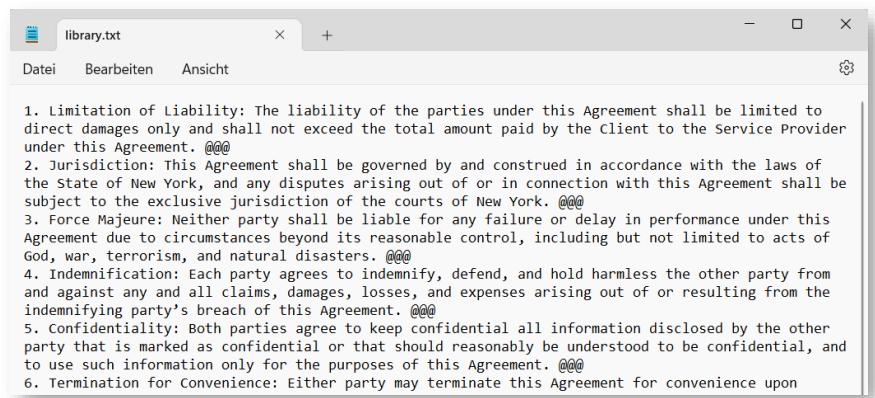
teren Anpassungen müssen manuell erfolgen. Achtung: Es wird die bestehende pptx-Datei angepasst. Ggf. ist also vorher eine Sicherheitskopie zu erstellen. Nach der Generierung, kann über "Create Audio" (Rz. 134) eine gesprochene Version erstellt werden, d.h. die Sprechernotizen können in einen Audioclip umgewandelt werden, der automatisch in die Präsentation eingefügt wird.

- Mit "**Pure:**" kann Red Ink schliesslich veranlasst werden, dass die eingegebene Instruktion ohne zusätzliche Instruktionen von Seiten Red Ink an das Sprachmodell übergeben wird (mit Ausnahme der Instruktion, dass bestehende Formatierungen erhalten bleiben, falls die entsprechende Option gewählt worden ist). Dies kann benutzt werden, um direkte Prompts an die KI zu übergeben (funktional als System-Prompt, wo danach unterschieden wird). Normalerweise wird das aber nicht benötigt. Es werden diesfalls keine weiteren Trigger-Codes ausgeführt. Ein markierter Text wird jedoch als User-Prompt übergeben (einbettet in einen <TEXTTOPROCESS>-Tag).

41 Im Prompt selbst können weitere funktionale Trigger-Codes enthalten sein, die eine Funktion auslösen:

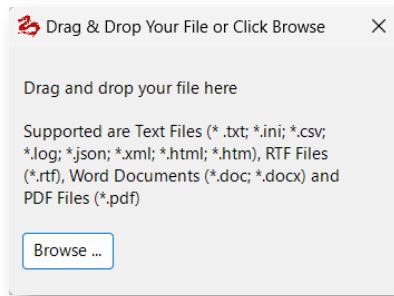
- In gewissen Fällen kann es sinnvoll sein, für die Ergänzung eines Textes oder Beantwortung einer Frage noch **Informationen aus dem Internet** abzurufen. Ist eine Suchmaschine konfiguriert, kann das Add-in in Word vor dem Ausführen des Befehls gebeten werden, eine Internet-Suche mit den für die Ausführung des Befehls fehlenden Informationen durchzuführen und die Informationen aus den ersten Treffern ebenfalls für den Befehl zu verwenden. Hierzu muss dem Befehl ein "**(net)**" angehängt werden. Ob diese Funktionalität zur Verfügung steht, wird jeweils im Hilfetext the Prompt-Box angezeigt, wenn der Freestyle-Befehl abgefragt wird (dies lässt sich konfigurieren, auch die Suchmaschine). Wird sie benutzt, wird Red Ink bei entsprechender Konfiguration (Parameter "ISearch_Approve", siehe unten) anzeigen, mit welchen Suchbefehlen die Suche durchgeführt werden soll und eine Bestätigung verlangen. Damit kann sichergestellt werden, dass keine vertraulichen Angaben an die Suchmaschine weitergegeben werden. Diese Funktion funktioniert nur dort, wo die Suchtreffer die Informationen unter der gefundenen Adresse tatsächlich angezeigt werden; bei komplexeren Websites ist dies nicht unbedingt der Fall, weil die gefundene Adresse ihrerseits auf zahlreiche weitere, verschachtelte Unterseiten verweist. Das Add-in kann konfiguriert werden, wie tief es in eine Website einsteigt und für wie lange.
- Das Add-in kann gebeten werden, zur Ausführung eines Befehls statt auf sein eigenes Wissen primär auf eine passende Information aus einer Bibliothek zurückzugreifen. Dies wird in der Fach-

sprache als **Retrieval Augmented Generation**, kurz **RAG**, bezeichnet. Ein Anwendungsbeispiel ist eine Text-Datenbank mit Vertragsklauseln sein. Hierzu muss dem Befehl ein "(lib)" hinzugefügt werden (kann nicht mit "(net)" kombiniert werden), vorausgesetzt, die Bibliothekssuche ist konfiguriert worden und die Bibliothek ist abrufbar. In diesem Fall wird Red Ink zuerst die Bibliothek entsprechend den konfigurierten Vorgaben und dem eingegebenen Freestyle-Befehl absuchen und dann den Befehl mit dem gefundenen Inhalt auf den selektierten Text anwenden. Wie er dies tut, muss mittels dem entsprechenden Prompt gesagt werden. Wird nicht Markup verwendet, wird der Output an den selektierten Text angehängt. Die Bibliothek ist eine einfache Datei im Format TXT oder Word (Word ist etwas langsamer) mit den entsprechenden Einträgen, getrennt durch ein Zeichen, das im Prompt zu berücksichtigen ist (z.B. "@@@"). Die Funktion "(lib)" steht sowohl mit dem primären Sprachmodell als auch einem allfälligen sekundären Modell zur Verfügung. Die Modellwahl hat aber nur auf den zweiten Befehl Auswirkung, d.h. die Anwendung des aus der Bibliothek extrahierten Inhalts, nicht die Extraktion des Begriffs aus der Bibliothek. Die Prompts können individuell konfiguriert werden. Damit kann auch auf eine etwaige spezielle Struktur oder andere Trennzeichen der Bibliotheksdatei eingegangen werden:



- Wird an einer Stelle im Prompt der Platzhalter "{doc}" eingefügt, wird Freestyle dort den Text eines **externen Text-Dokuments** einfügen. Das kann sinnvoll sein, um beispielsweise den Inhalt dieses Dokuments auf einen bestehenden Text anzuwenden oder mit diesem von der KI vergleichen zu lassen, wie das mit einer sonstigen Markupfunktion nicht möglich wäre. Es werden reine Text-Formate unterstützt (wie ".txt", ".html" oder ".csv"), Word-Dokumente (".docx" und ".doc") sowie PDF und (in Word) auch Powerpoint-Dateien ("pptx"). Im Falle von PDFs wird zuerst versucht, der im PDF eincodierte Text auszulesen. Wo Text hingegen als Bild abgelegt ist, d.h. das PDF nicht nach Text durchsuchbar ist, geht dies nicht. Falls das primäre Modell zur Verarbeitung von

Dateiobjekten konfiguriert worden ist, kann das Add-in in solchen Fällen versuchen, eine Texterkennung durch die KI durchzuführen. Das Add-in wird danach fragen und benötigt dafür auch etwas mehr Zeit. Falls das Modell dies unterstützt und dafür konfiguriert worden ist, können alternativ auch die Trigger "(file)" oder "(clip)" benutzt werden, dazu sogleich. Diese Funktion steht auch als Word-Helper zur Verfügung (Rz. 175 ff. unten).



Wird "{doc}" in den Prompt eingebaut, empfiehlt es sich, ihn mit einem Tag zu versehen, damit das Sprachmodell ihn besser von der Instruktion unterscheiden kann, z.B. mit "Hier ist der externe Text: <TEXT>{doc}</TEXT>".

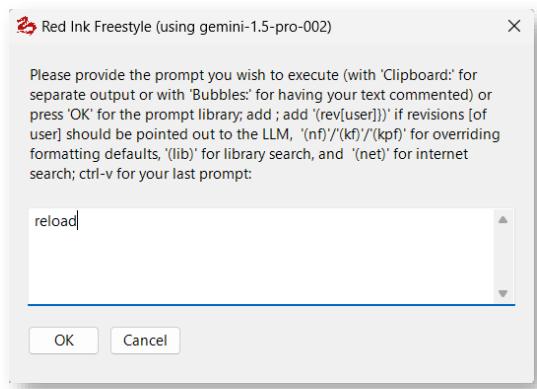
Wird "{doc}" **mehrfach** im selben Prompt verwendet, fragt das Add-in den Benutzer mehrfach nach einer Datei ab. So können mehrere Dokumente im selben Prompt berücksichtigt werden. Hierbei ist die Verwendung von Tags wie soeben beschrieben besonders wichtig, weil die Texte jeweils an der Stelle im Prompt eingefügt werden, wo sich das jeweilige "{doc}" befindet.

- Der Trigger "**(file)**" erlaubt es auch, externe Dateien einzulesen, aber auf eine andere Weise. Hier wird das Add-in beauftragt, den Inhalt der Datei direkt an das Sprachmodell zu übermitteln. Das funktioniert allerdings nur mit den Dateiformaten, die das Sprachmodell auch unterstützt und wenn das Sprachmodell entsprechend konfiguriert worden ist ("APICall_Object"). Moderne Sprachmodelle unterstützen die gängigen Formate von Bildern, Tondateien und Videos sowie PDF-Dokumente. Wird der Trigger gesetzt, geht ein Fenster auf, in welches die betreffende Datei gezogen werden kann. Ein Beispielprompt für diesen Trigger wäre "*Was ist auf diesem Bild zu sehen? (file)*" oder – falls das Modell auch die Bildgenerierung unterstützt und entsprechend konfiguriert worden ist – "*Färbe mir das Objekt auf diesem Bild blau ein und erstelle ein neues Bild davon. (file)*". Andere Anwendungen sind das Transkribieren von Tondateien und Analysieren oder Umwandeln von Inhalten von PDF-Dokumenten. Das Add-in prüft allerdings weder, ob das Format der Datei unterstützt wird noch ob die Datei zu gross ist. Je grösser die Datei, desto länger dauert auch die Antwort des Sprachmodells.

- Dasselbe funktioniert auch mit dem Inhalt der Zwischenablage, wenn statt "(file)" der Trigger "**(clip)**" eingefügt wird. Das ist ganz besonders praktisch, wenn während dem Arbeiten an einem Dokument Text aus einem anderen Dokument, der nicht einfach kopiert werden kann, eingefügt werden soll. Es braucht blass ein Screenshot der betreffenden Stelle gemacht zu werden (Shift-Windows-S) und dann kann Freestyle mit einem Befehl wie "Extrahiere mir den Text aus dem Bild (clip)" die KI angewiesen werden, den Text zu extrahieren und als Antwort zu liefern. Der Einfachheit halber ist ein entsprechender Befehl unter den Word-Hilfslern bereits vorprogrammiert ("**Clipboard to text**") und kann auch mit dem Kurzbefehl "**insertclip**" or "**iclip**" innerhalb von Freestyle ausgeführt werden.
- Der Trigger "**(rev)**" beauftragt das Add-in, sämtliche **Markups im selektierten Text als solche zu codieren**, bevor der Text an das Sprachmodell übergeben wird. Damit ist es möglich, dem Sprachmodell Fragen zu den Markups zu stellen, z.B. "Fasse die am Vertrag gemachten Änderungen hinsichtlich ihrer Bedeutung zusammen.". Wer nur Markups eines bestimmten Autors codiert haben möchte, gibt den Namen nach einem Doppelpunkt genau so an, wie er im Text steht (z.B. "(rev:VISCHER)"). Sie müssen der KI ausreichend klar sagen, was mit den Markierungen (Löschen oder Einfügungen/Ergänzungen) zu tun ist, weil sie diese nur sehen wird, aber standardmäßig nicht weiß, was sie damit machen soll.
- Mit "**(noformat)**" oder "**(nf)**" im Prompt kann das Add-in veranlasst werden, keine Formatierungssangaben zum Originaltext vorübergehend zu hinterlegen oder sich zu merken, was bei längeren Texten signifikant die Verarbeitungsgeschwindigkeit erhöht. Zwar kann wie oben dargelegt, eine Zeichenlimite fix konfiguriert werden, allerdings nicht für Freestyle isoliert (siehe "Maximum text ..." in Rz. 25 oben). Mit "**(keepformat)**" oder "**(kf)**" wird umgekehrt die Keep-Format-Funktion für den aktuellen Prompt aktiviert, mit "**(keepparaformat)**" oder "**(kpf)**" die zweite, in Rz. 25 beschriebene Funktion zum Erhalt von Formatierungen. Wer also bei Freestyle will, dass das Format eines bestehenden Textes erhalten bleibt (so gut das geht), muss das mit diesen Codes im Prompt verlangen; die obigen Konfigurationseinstellungen zum Erhalt von Formatierungen greifen bei Freestyle bewusst nicht. Wird "**(sar)**" (*same as replace*), wird die Spezialfunktion für den Erhalt von dynamischen Feldern, Referenzen und Zeichenformatierungen aktiviert, wie wenn es um das Ersetzen des Textes ginge.
- Mit dem Zusatz "**(mystyle)**" wird die MyStyle-Funktion aktiviert, mit welcher die KI angewiesen werden kann, den eigenen, vorher definierten Schreibstil zu befolgen (siehe Rz. 48 ff.).

- Mit dem Zusatz "**(all)**" kann das Add-in angewiesen werden, den Text des gesamten Dokuments für die Anfrage zu verwenden, auch wenn nicht alles selektiert worden ist.
 - Mit dem Zusatz "**(adddoc)**" wird das Add-in angewiesen, zusätzlich zum selektierten Text noch den gesamten Text des aktuellen oder eines anderen geöffneten Word-Dokuments (oder der gesamte Text aller geöffneter Word-Dokumente) der KI mitzugeben mit der Aufgabenstellung. Dies kann z.B. benutzt werden, um die KI zu bitten einen Textteil zu überarbeiten aber dabei den ganzen Inhalt des Dokuments zu berücksichtigen.
 - Mit dem Zusatz "**(iterate)**" kann das Add-in angewiesen werden, den Befehl nicht in einem Stück, sondern Schritt für Schritt abzuarbeiten. Wird die gewählt, fragt das Add-in nach der Anzahl Absätze, die jeweils pro Schritt abgearbeitet werden. Wird also z.B. der Wert 3 gewählt, wird das Add-in nicht den gesamten selektierten Text mit dem gewünschten Befehl ausführen, sondern zuerst nur die ersten drei Absätze, dann die nächsten drei Absätze. Mit Clipboard und Pane funktioniert dies auch nicht. Bubbles funktioniert jedoch damit. Mit Markups werden nur Diff und Regex als Methode unterstützt (es wird automatisch danach gefragt, wenn sie nicht gewählt sind). Iterate ist vor allem bei grossen Texten sinnvoll, die auf diese Weise in kleinere Blöcke für besseren Output aufgeteilt werden kann. Vordefinierte Befehle wie Translate, Shorten oder Improve bieten Iterate nicht an, aber über Freestyle kann diese mit dem entsprechenden Prompt gewissermassen nachempfunden werden ("Kürze mir jeden Absatz soweit wie möglich (iterate)"). Am zuverlässigsten ist der Wert 1.
- 42 Diese Zusätze funktionieren auch, wenn sie in den Prompts der Prompt-Bibliothek angegeben sind. Dort können die speziellen Ausgabeformate (Prefixe "Clipboard", "Markup" und "Bubbles") auch über eine Checkbox aktiviert werden.
- 43 Falls das verwendete Sprachmodell multimodal ist und die **Generierung von Bildern** unterstützt, speichert Red Ink ein vom LLM geliefertes Bild automatisch in einer (fortlaufend nummerierten) Datei auf dem Desktop und fügt einen Hinweis im Text ein, der den Speicherort des Bildes angibt (vorausgesetzt, Bildcodierung und -format wird von Red Ink unterstützt). Diese Ausgabe kann z.B. auch in die Pane umgeleitet werden.
- 44 Soll der letzte Freestyle-Befehl mit dem zuletzt gewählten Modell (aber mit der aktuellen Selektion) wiederholt werden, dann führt dies **Freestyle (redo)** aus.
- 45 Red Ink **speichert** automatisch jeden Freestyle-Prompt, inklusive Zeitstempel. Die letzten 50 Prompts können mit dem Kurzbefehl "prompt-log" abgerufen werden (Rz. 47).

- 46 Soweit dies für ein Modell mittels des Parameters "**TokenCount**" bzw. "**TokenCount_2**" konfiguriert ist, ist es möglich, die für eine Free-style-Anfrage aufgewendeten Tokens zu protokollieren und mit einem Währungsbetrag multiplizieren zu lassen. Dies kann dazu genutzt werden, um entsprechende Aufwendungen einem Klienten oder einer Kostenstelle weiterzuverrechnen. Die Token-Kosten mögen zwar tief sein, aber wenn durch den Einsatz der KI gewisse Arbeiten schneller erledigt werden können, kann es je nach Fall Sinn machen, die Wertschöpfung zu verrechnen, gewissermassen nach dem Prinzip, dass auch die "Denkleistung" der KI einen Wert hat. Das kann beim Einsatz von Freestyle nützlich sein, z.B. wenn die KI gebeten wird, ein Memorandum zu erstellen oder eine Analyse vorzunehmen und einem Klienten dies verrechnet werden soll. Die Ausgabe erscheint in einer Textdatei namens "**redink-cost.txt**" auf dem Desktop des Benutzers. Weitere Einträge werden jeweils angehängt. Die Kostenauswertung ist auf Freestyle in Word und Excel sowie beim CSV-Analyzer in Excel beschränkt (Achtung: Da die Parameter TokenCount und TokenCount_2 modellspezifisch sind, müssen sie bei jedem alternativen Modell, das verwendet wird und bei dem protokolliert werden soll, ebenfalls definiert werden).
- 47 Über Freestyle können ferner noch eine Reihe von **Kurzbefehlen** ausgeführt werden; einige davon sind nur für Administratoren. Sie werden einfach anstelle des Prompts eingegeben und mit OK bestätigt:



- "**model**" gibt das aktuell verwendete primäre Modell und der aktuelle Timeout-Wert des Modells aus.
- "**terms**" gibt die im Ini-File allfällig vorkonfigurierten Verwendungsbeschränkungen oder Nutzungsbefugnisse aus. Sie werden auch angezeigt, wenn mit der Maus über das Red Ink Logo in der Menüsleiste gefahren wird.
- "**version**" macht Angaben zur aktuellen Version des Add-in; sie erscheinen auch beim Fahren über das Red Ink Logo in der Menüsleiste. Der Befehl "version" zeigt auch, bis wann die aktuelle Kopie von Red Ink lizenziert ist (diese Information lässt sich auch über das Settings abrufen, dort unter "About Red Ink". Dort sind

auch weitere Angaben, z.B. über verwendete Drittbibliotheken abrufbar.

- "**cleanmenu**" entfernt allfällige bestehende Kontextmenüs von Red Ink und baut sie neu auf, falls sie aktiviert sind.
- "**switch**" kann temporär das primäre und sekundäre KI-Modell tauschen; dies ist auch über Settings möglich.
- "**clearlastprompt**" löscht die Zwischenspeicherung des zuletzt ausgeführten Freestyle-Prompts.
- "**reload**" sorgt dafür, dass das Add-in die Konfigurationsdatei neu lädt (weil z.B. etwas inzwischen manuell angepasst worden ist; dies geschieht automatisch auch nach dem Speichern der Konfiguration in Settings).
- "**reset**" setzt die lokale Konfigurationsdatei so zurück, dass nur noch die minimal erforderlichen Einträge vorhanden sind und die anderen Werte auf die Standardwerte gesetzt werden.
- "**settings**" ruft die Funktion zur manuellen Anpassung der Einstellungen auf.
- "**encode**" kann benutzt werden, um API Keys und Private Keys so zu verschlüsseln, dass sie nicht im Klartext in der Konfigurationsdatei abgelegt werden müssen. Es ist dazu der Key im Klartext in Word zu markieren (siehe dazu Rz. 345 ff. unten).
- "**decode**" kann sie entschlüsseln, falls das Schlüsselwort bekannt ist (siehe dazu Rz. 345 ff. unten).
- "**inipath**" erlaubt es, das Verzeichnis für eine zentrale Konfigurationsdatei in die Registry zu schreiben (siehe dazu Rz. 320 ff. unten).
- "**codebasis**" erlaubt es, das Schlüsselwort in die Registry zu schreiben, sollte es nicht im Programmcode fest einprogrammiert sein (siehe dazu Rz. 345 ff. unten).
- "**domain**" zeigt die aktuelle Domäne an, in welcher das Add-in läuft und ob und auf welche Domänen es beschränkt ist, sollte dies als Sicherheitsfunktion einprogrammiert sein (siehe dazu Rz. 345 ff. unten).
- "**definemystyle**" startet den Prozess zur Analyse des Schreibstils und Erstellung eines MyStyle-Prompts.
- "**editmystyle**" öffnet einen Texteditor mit der MyStyle Prompt-Datei.
- "**speech**" startet den Transcriptor (siehe dazu Rz. 83 ff. unten).
- "**voices2**" öffnet das Fenster zur Auswahl von zwei Stimmen für die Verwendung der Google Text-to-Speech-Funktion, "**voices**" jenes für die Auswahl einer Stimme; diese Funktion braucht es

normalerweise nicht, weil sowohl bei der Podcast- als auch der Audiobook-Funktion die Auswahl automatisch geöffnet wird; allenfalls können diese Befehle hilfreich sein, wenn nur Stimmen ausgewählt werden sollen. Mehr dazu in Rz. 131 ff. unten.

- "**createpodcast**" startet die Funktion zur Erstellung von Podcasts (siehe dazu Rz. 131 ff. unten).
- "**read**" startet die Funktion zur Erstellung von Audiobooks, d.h. der selektierte Text wird vorgelesen (siehe dazu Rz. 131 ff. unten).
- "**readlocal**" wird den aktuell markierten Text mit der eingebauten Speech-to-Text-Funktion vorlesen (oder dies abbrechen); es werden keine Daten ans Internet gesandt (anders als bei Words Laut-Vorlesen- und der Google-Text-to-Speech-Funktion), aber die Stimme klingt nicht natürlich.
- "**voiceslocal**" erlaubt das Einstellen der Stimme für "readlocal".
- "**anonymize**" führt die sonst beim Aufruf von Sprachmodellen optional benutzte Anonymisierungsfunktion (im Modus 3 oder 4) ohne Aufruf eines Sprachmodells am selektierten Text aus. Damit kann die Anonymisierung getestet werden. Dies kann auch über den Menüpunkt "Analyze" aufgerufen werden.
- "**generateresponsetemplate**" oder "**generateresponsekey**" dient dem automatisierten Erstellen der Templates, die in der "Special Service"-Konfigurationsdatei für Verarbeitung der vom jeweiligen Service zurückgelieferten JSON-Strings benötigt werden (siehe dazu Rz. 340 ff. und Anhang 2). Um die Funktion zu benutzen, sollte in ein Word-Dokument zuerst ein beispielhafter JSON-String in der typischen Antwort-Struktur des betreffenden Service kopiert werden, und dann eine Prosa-Beschreibung dessen, was das Template können soll bzw. wie die Ausgabe basierend auf den jeweiligen Feldern und Werten aussehen soll. Dies wird selektiert und in Freestyle der Befehl angewählt. Er wird dann dem aktuellen LLM zur Generierung des Templates übergeben, zusammen mit den dazu nötigen Angaben (d.h. dem Programmcode, der die Templates verarbeitet).
- "**insertclipboard**", "**insertclip**", "**iclip**" oder "**clipboard**" führt den Befehl aus, welcher dem Sprachmodell den Inhalt der Zwischenablage übergibt und ihn bittet, den Text daraus zu extrahieren (bei Bildern mit Text oder Ton- oder Videoaufnahmen) oder den Inhalt in Text zu beschreiben (bei Bildern). Die Funktion steht auch als Word Helper zur Verfügung. Das Modell muss für diese Art der Funktion (Verarbeitung von Binär-Objekten) konfiguriert sein.
- "**redinktest**" liest den Text, der in der Datei "redinktest.txt" auf dem Desktop des Benutzers enthalten ist, und zeigt ihn in einem

Fenster an. Es kann verwendet werden, um das Rendern von Markdown-formatiertem Text zu testen. Das Fenster verwendet RTF. Wenn der Inhalt in das Dokument eingefügt wird, wird er separat gerendert.

- "**promptlog**" zeigt die letzten gespeicherten Freestyle-Prompts an, wie sie automatisch zwischengespeichert wurden. Die zwischengespeicherten Prompts können auch bearbeitet werden (z.B. falls einer gelöscht werden soll aus dem Zwischenspeicher, dann ist er aus dem angezeigten Text inkl. Trennzeile zu löschen und der bearbeitete Text mit "OK" zu übernehmen; der Zwischenspeicher wird basierend darauf nachgeführt). Ist die Limite (50) erreicht, wird der jeweils älteste Prompt gelöscht.
- "**webagent**" ruft die WebAgent-Funktion auf. Sie setzt allerdings voraus, dass Ordnerpfade für die betreffenden Skriptdateien konfiguriert sind. Mit "**webagentcreator**" kann die Funktion zum Erstellen und Verändern von Skripten aufgerufen werden.

D. KI-Texte im eigenen Schreibstil: MyStyle

1. Überblick

48 Wer Texte von der KI in seinem ganz persönlichen Schreibstil haben möchte, dem bietet Red Ink die Möglichkeit, diesen zu definieren und in verschiedenen Funktionen einzusetzen, d.h. bei der Verwendung von Freestyle in Word und Outlook, mit der Reply-Funktion von Outlook und in Word und Outlook als Variante der Funktion "Improve". Das funktioniert so, dass in einem ersten Schritt der eigene Stil von der KI analysiert werden muss. Daraus entsteht ein Prompt, der in einer persönlichen Datei gespeichert wird. Soll der Stil dann in einem konkreten Fall angewandt werden, wird dieser Prompt zusätzlich an die KI gesendet, damit sie ihn in ihrer Generierung von Text befolgt. Es können mehrere solcher MyStyle-Prompts hinterlegt werden, und sie können für Word und Outlook unterschiedlich sein. In Word werden sie anhand von Word-Textproben erstellt, in Outlook basierend auf E-Mails.

49 Die Funktion steht nur zur Verfügung, wenn ein Speicherpfad für die MyStyle Prompt Datei konfiguriert worden ist. Dies ist über die Konfigurationsdatei möglich (Rz. 309 ff.) oder über die Funktion "Settings" (dort allerdings nur temporär, falls sie nicht gespeichert wird). Nur wenn sie definiert ist, erscheinen die Menü-Einträge für MyStyle (z.B. im Improve-Menü).

2. MyStyle ermitteln

50 In einem ersten Schritt ist in Word bzw. Outlook "Define MyStyle" aufzurufen; in Word befindet sich der Befehl im Menü "Analyze". Es wird eine Einweisung angezeigt, die über die zu unternehmenden Schritte informiert (es kann mit einem Click auf den passenden Knopf auch die bestehende MyStyle Prompt Datei geöffnet und bearbeitet werden).

51 In Word werden berücksichtigt:

- Der allfällig bereits im aktuellen Word-Dokument selektierte Text;
- Ein vom Benutzer ausgewähltes offenes Word-Dokument oder alle offenen Word-Dokumente;
- Weitere Anweisungen, wie z.B. öffentliche Links, unter denen das Modell weitere Texte mit dem betreffenden Stil abrufen kann, falls das Modell dazu in der Lage ist.

Am besten ist es, wenn alle Word-Dokumente geöffnet werden, die relevante Texte enthalten und kein Text selektiert wird. In der Auswahl ist dann die Option für alle Dokumente zu wählen.

- 52 In Outlook werden berücksichtigt:
- Alle offenen E-Mails mindestens im Lesezugriff (es dürfen E-Mail-Ketten auch mit Mails anderer Personen sein);
 - Weitere Anweisungen, wie z.B. öffentliche Links, unter denen das Modell weitere Texte mit dem betreffenden Stil abrufen kann, falls das Modell dazu in der Lage ist.
- Der Benutzer wird ferner nach seinem Namen gefragt. Anhand dieser Angabe werden von der KI aus allen E-Mails jene für Analyse isoliert, deren Stil ermittelt werden soll. Es hat sich gezeigt, dass fünf bis zehn E-Mails genügen, wenn diese einigermassen konsistent sind.
- 53 Danach muss das Modell gewählt werden, welches die Analyse durchführen soll. Es kann zwischen dem primären und einem der sekundären bzw. alternativen Modell gewählt werden, falls solche konfiguriert sind. Es hat sich gezeigt, dass vorzugsweise ein Reasoning-Modell gewählt wird, je nach Zusatzinstruktion mit Internet-Suche, falls Links abgefragt werden sollen.
- 54 Wenn die KI die Analyse erstellt hat, wird das Ergebnis angezeigt. Sie enthält einen Teil für den Benutzer, wo ihm der Stil erläutert wird. Am Ende ist aber auch ein Prompt enthalten, mit welchem dieser Stil einer KI beauftragt werden kann. Es kommt folgendes Format zur Anwendung:
- [Title = xxxx] wobei xxxx für den Titel steht, unter welchem dieser Prompt bzw. Stil später abrufbar sein wird.
 - [Prompt = yyyy] wobei yyyy für den Prompt steht, der den Stil generieren soll.
- Die Analyse und insbesondere Prompt und Titel können geändert werden (z.B. ein neuer Titel gewählt werden). Wird einer der OK-Knöpfe gedrückt, werden Titel und Prompt als Stil der betreffenden Applikation (Word, Outlook) in der MyStyle Prompt Datei abgelegt (und von der alten wird eine Sicherheitskopie erzeugt).
- 55 Die MyStyle Prompt Datei ist eine normale Text-Datei, die mit jedem Editor bearbeitet werden kann. Wer sie innerhalb von Red Ink bearbeiten will, kann "Define MyStyle" aufrufen und bei der Instruktion auf

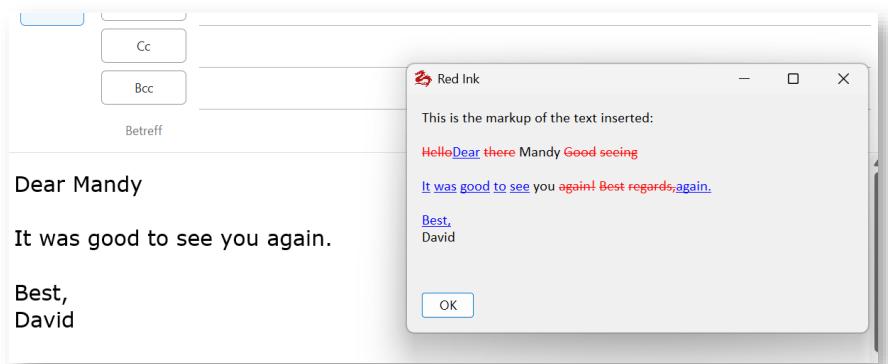
"Edit" klicken. Die Datei ist so aufgebaut, dass jeder Prompt auf einer Zeile steht, beginnend mit "All", "Word" oder "Outlook" für die Anwendung, in welcher er verfügbar sein soll, dann der Titel, dann der Prompt, alles jeweils abgetrennt durch einen senkrechten Strich ("|"). Zeilen, die mit ";" beginnen, werden ignoriert.

- 56 Soll ein Prompt gelöscht werden, dann geht das einfach durch Löschen aus der Textdatei. Er kann dort auch editiert werden.

3. MyStyle anwenden

- 57 Die Nutzung von MyStyle ist in verschiedenen Funktionen möglich, abhängig von der Anwendung (Word, Outlook):

- In Freestyle kann MyStyle zugeschaltet werden, indem dem Prompt der Trigger "(mystyle)" zugefügt wird. Bevor er ausgeführt wird, wird der Benutzer gebeten, aus der Liste der für die Anwendung verfügbaren Stilen einen auszuwählen.
- Im Menü "Improve" steht die Funktion Apply MyStyle zur Verfügung. Sie wird wie die anderen Improve-Funktionen benutzt: Textstelle anwählen, dann Apply MyStyle ausführen, der passende Stil-Prompt auswählen und Red Ink führt die Anpassung aus:



- In Outlook steht MyStyle zusätzlich in der Funktion "Reply" zur Verfügung. Ist dort eine etwaige Instruktion eingegeben, dann folgt die Abfrage des gewünschten Stil-Prompts. Alternativ kann auch "None" ausgewählt werden.

- 58 MyStyle ist für die Anwendung eines persönlichen Stils gedacht. Soll ein Unternehmensweiter Stil umgesetzt werden, eignet sich die Nutzung des Konfigurationsparameters "PreCorrection" (siehe Rz. 309 ff.) besser. Für die Anpassung des "Scharfen S" wiederum gibt es wiederum einen eigenen Parameter. Dafür braucht MyStyle nicht bemüht zu werden.

E. Suche nach Themen: Context Search

- 59 Mit der Funktion Context Search kann das Add-in gebeten werden, im aktuellen Text nach bestimmten Themen zu suchen und diese zu markieren. Anders als die eingebaute Text-Suche von Word findet diese Funktion auch Stellen, die nicht genau den eingegebenen Suchbegrif-

fen entsprechen, aber dasselbe Thema behandeln. Zum Beispiel kann der Kontext-Suchbegriff "Liability" in einem Vertrag auch Textstellen finden, in denen das Wort in genau dieser Form nicht vorkommt, aber etwas, das dieselbe Bedeutung aufweist (z.B. den Satz "... remain fully liable for all obligations ..."). Der Treffer wird vom Add-in angezeigt, indem er selektiert wird:

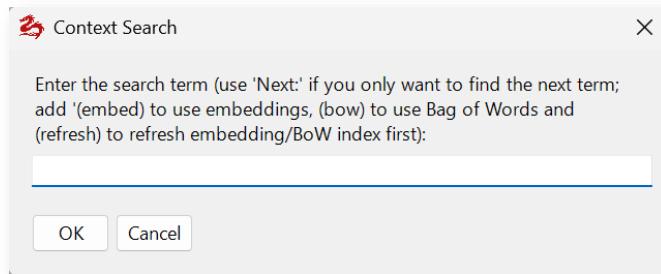
ii. if required under Applicable Privacy Laws, the data protection obligations described in this Addendum are imposed on the Subprocessor (as may be further described in Appendix 3 (Specific Privacy Laws)); and

b. remain fully liable for all obligations subcontracted to, and all acts and omissions of, the Subprocessor.

11.4 Opportunity to Object to Subprocessors.

a. When Google engages any New Subprocessor during the Term, Google will at least 30

- 60 Bei der Eingabe von Kontext-Suchbegriffen muss kein vollständiger Prompt erfasst werden. Es genügt, die Begriffe zum Kontext einzugeben. Es kann allerdings auch eingeschränkt werden ("liability but not audit" findet keine Audit-Klauseln, obwohl diese mit Liability zusammenhängen können).



- 61 Diese Kontext-Suche wird durch das aktuelle primäre Sprachmodell durchgeführt, d.h. bei längeren Texten kann dies etwas dauern. Alternativ bietet Red Ink auch eine Vektor-Suche und eine sog. Bag-of-Words-Suche an:

- Bei der **Vektor-Suche** wird das aktuelle Dokument zuerst "vektoriert", d.h. der Text wird in sog. Chunks aufgeteilt (z.B. Gruppen von zwei Sätzen), die dann im Speicher in einer multidimensionalen Datenraum (der "Embedding Space") an einer virtuellen Koordinate ihrer Bedeutung entsprechend abgelegt werden. Das ermöglicht es, dem Benutzer mit eigenen Worten nach Sätzen zu suchen, die dasselbe aussagen, aber möglicherweise mit anderen Worten. LLM benutzen diese Technik ebenfalls. Wenn Context-Search hierzu verwendet wird, führt es diese Vektorisierung beim aktuellen Text einmalig durch (sie muss allerdings wiederholt werden, sobald der Text irgendwie verändert wird. Voraussetzung, dass die Vektor-Suche benutzt werden kann ist zudem die Installation eines geeigneten Modells zum Ermitteln der Bedeu-

tungen von Sätzen (dazu nachfolgend). Steht die Vektor-Suche zur Verfügung, kann sie mit dem Trigger "**(embed)**" aktiviert werden und wird mit dieser Technik durchgeführt. Dafür braucht es dann kein LLM.

- Die **Bag-of-Words-Suche** ist eine einfache Methode, bei der ein Text zunächst in kleine Elemente zerteilt wird (Wörter, Tokens), die in einer Liste erfasst werden. Anschliessend wird gezählt, wie oft jedes dieser Wörter im Dokument vorkommt, ohne dabei die Reihenfolge oder den Kontext zu berücksichtigen. Möglich ist damit eine schnelle Stichwortsuche über den gesamten Text. Sie findet aber nur Stichwörter, die tatsächlich so vorkommen, ist also streng genommen keine Kontext-Suche. Sie kann mit dem Trigger "**(bow)**" aktiviert werden.

Bei beiden Such-Methoden kann mit dem Trigger "**(refresh)**" das System veranlasst werden, den Index neu zu generieren (z.B. nach Änderungen (inkl. Verschiebungen) im Text).

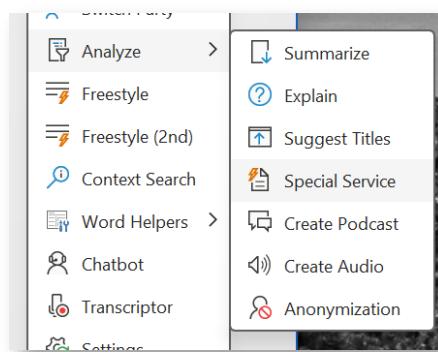
- 62 Aus Effizienzgründen wird standardmässig das jeweils gesamte Dokument auf ein Mal durchsucht. Es werden dabei alle Treffer mit einem Word-Kommentar versehen. Bei der Vektor-Suche und BoW-Suche wird noch die Treffer-Relevanz angegeben; dort kann als Parameter z.B. auch angegeben werden, wieviele Treffer angezeigt werden sollten und welche Minimalrelevanz sie haben sollen (findet er hierfür keine, wird die Relevanz ausgeweitet). Bei der Vektor- und Bag-of-Words-Suche kann angegeben werden, wie gross die Chunks sein sollen (z.B. jeweils zwei Sätze, mit einem Satz an Überlappung).
- 63 Wer nur den nächsten Treffer anzeigen lassen will, stellt der Suchabfrage den Prefix "**Next:**" voran, was allerdings nur bei der normalen Kontext-Suche mit einem Sprachmodell wirklich Sinn macht.
- 64 Wird das Context Search Fenster geöffnet, erscheint automatisch der letzte Suchbefehl.
- 65 Der Chatbot Inky (Rz. 176 ff. unten) kann übrigens auch solche Kontextsuchen durchführen. Sie sind etwas anders programmiert und können daher zu unterschiedlichen Ergebnissen führen.
- 66 Um die Vektor-Suche zu aktivieren, muss zuerst ein **geeignetes Modell** mit einer Datei für den Tokenizer installiert werden. Ausgangspunkt ist der Pfad, der im Parameter "LocalModelPath" angegeben ist (z.B. "D:\ModelsInUse\"). Dort muss das Unterverzeichnis "embed" erstellt werden, und darin ist das Modell als "model.onnx" (in den offiziellen Quellen wird die Datei typischerweise einen anderen Namen tragen, aber ebenfalls mit ".onnx" enden) und die Datei für den Tokenizer als "vocab.txt" abgespeichert werden.
- 67 Ein Open-Source-Modell wird auf <https://apps.vischer.com> als separater Download bereitgestellt ("all-MiniLM-L6-v2-onnx"); es kann alternativ auch von HuggingFace heruntergeladen werden (benötigt werden

nur die beiden genannten Dateien). Dieses Standardmodell hat 384 Dimensionen (Float32 Array), eine Maximum Sequence Length von 256 Token, Input Tensors vom Typ Int64, Shape [1, seqLen] "input_ids", "attention_mask" und "token_type_ids", und den Output-Tensor vom Typ Float32 (384), mit einem ONNX Opset von ≥ 11 . Es wird ein Wordpiece-Tokenizer verwendet. Es unterstützt diverse Sprachen, aber für Fachtexte eignet es sich nach unserer Erfahrung nur beschränkt. Wir werden hier nach weiteren Modellen Ausschau halten und diese testen.

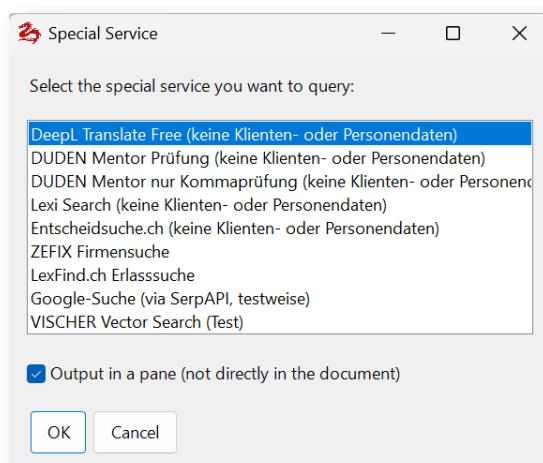
F. Zugriff auf weitere Datenquellen und Dienste: Special Services

68 Red Ink kann in Word auch für den Zugriff auf Online-Services von Dritten oder im eigenen Unternehmen zuzugreifen, zum Beispiel um Rechtsinformationen abzurufen, um spezialisierte KI-Services (wie z.B. den Übersetzungsdiensst von DeepL) zu nutzen oder Angaben aus einem internen Know-how-System (z.B. an einen Server mit einer Vektor-Datenbank mit allen internen Know-how-Dokumenten) abzurufen.

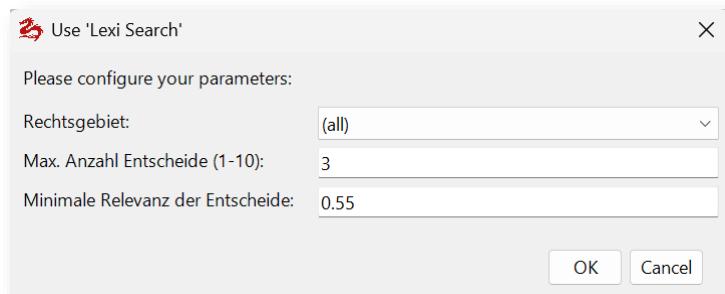
69 Der Zugang dazu erfolgt über den Befehl **Analyze** und dort über **Special Service**:



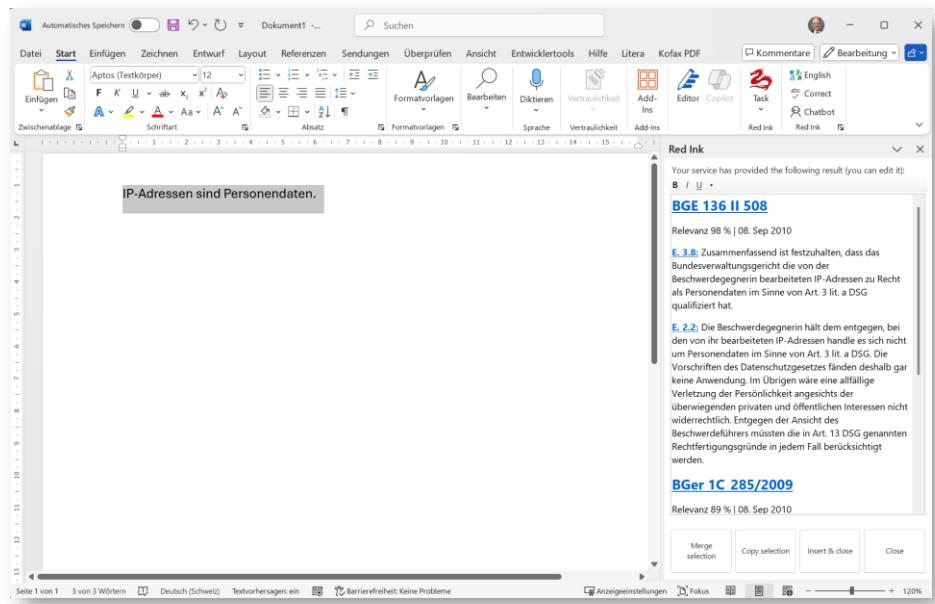
70 Er erscheint nur, wenn solches "speziellen Dienste" konfiguriert sind, und benutzt werden können sie nur, wenn zuvor eine Textstelle ausgewählt worden ist, die an diesen Dienst übermittelt werden soll. Ist das der Fall, wird die Liste der verfügbaren Dienste angezeigt:



- 71 Der gewünschte Service wird angewählt (hier z.B. ein Service, der Schweizer Gerichtsentscheid liefert, die zur Aussage im angewählten Textstück passen [www.lexisearch.ch]), und es können etwaige Parameter zur Abfrage erfasst werden. Diese sind bei jedem Dienst anders. Hier das Beispiel von Lexi Search:



- 72 Soweit dies entsprechend konfiguriert ist, kann ein **Query Assistant** die Abfrage unterstützen. Wird er angewählt (er erscheint in der obigen Abfrage als letzter Punkt), dann wird der selektierte Text vorab an das primäre LLM gesendet mit der Bitte, die nötigen Suchbegriffe für die Abfrage zu extrahieren (der dazu verwendet Prompt kann pro Special Service in dessen Konfiguration definiert werden). Die ermittelten Suchbegriffe werden vor der Verwendung in einem Fenster angezeigt und können noch geändert werden. Das ist auch dann hilfreich, wenn sichergestellt werden soll, dass keine vertraulichen Daten an den Special Service gesendet werden sollen.
- 73 Danach wird die Abfrage durchgeführt und das Ergebnis entweder direkt im Dokument eingefügt oder es wird damit ein neuer Fensterbereich (eine sog. *Pane*) rechts vom Dokument geöffnet, wo der Text gelesen und bearbeitet und weiterverwendet werden kann. Die Pane kann offen bleiben, solange dies gewünscht ist. Der Inhalt wird allerdings bei der nächsten Anfrage überschrieben.



- 74 Unten an der Pane hat es mehrere Knöpfe: **Merge Selection** erlaubt es, den selektierten Text mit Hilfe der KI in den im Dokument selektierten Text zusammenzuführen (z.B. hier könnte der gefundene Gerichtsentscheid mit Fundstelle eingefügt werden). Ist sowohl im Dokument als auch in der Pane eine Textstelle selektiert, und wird der Knopf gedrückt, geht ein Fenster auf, in welchem der Prompt zur Zusammenführung eingegeben werden kann. Für jeden Dienst kann ein eigener Prompt konfiguriert werden. Weiter steht ein Knopf zum Einfügen des in der Pane selektierten Textes in die Zwischenablage zur Verfügung und ein Knopf, mit welchem der vom Service gelieferte Originaltext (also ohne nachträgliche Bearbeitungen) mit der Originalformatierung im Dokument eingefügt werden kann.
- 75 Die **Konfiguration** erfolgt über eine separate Konfigurationsdatei, die ähnlich wie die Konfigurationsdatei für Modelle aufgebaut ist und deren Pfad in der Konfigurationsdatei hinterlegt wird ("SpecialServicePath"). Die Angaben sind dieselben wie bei einem LLM und die Datei ist gleich aufgebaut wie die Konfigurationsdatei für alternative Modelle (siehe dazu Rz. 331 ff.). Der einzige Unterschied besteht darin, dass hier für jeden Service bis zu vier zusätzliche Parameter sowie einen sog. "MergePrompt" und einen "QueryPrompt" erfasst werden können, die wie oben angezeigt abgefragt werden. Der Eintrag für Lexi Search sieht zum Beispiel so aus:

```
[Lexi Search (keine Klienten- oder Personendaten)]

APIKey = xxxxxxxx
APIKeyPrefix =
APIKeyEncrypted = False
Model = Lexi Search
Endpoint = https://www.lexisearch.ch/api/v1/search
HeaderA = Authorization
HeaderB = Bearer {apikey}
Response = response
APICall = {"search": {"query": "{promptuser}", "filters": {"decision__law_field": "{parameter1}", "top_k": {parameter2}, "min_score": {parameter3}}}, "locale": "de"}
Timeout = 200000
Parameter1 = Rechtsgebiet; String; (all); (all), Zivilrecht<civil>, Strafrecht<criminal>, Öffentliches Recht<public>
Parameter2 = Max. Anzahl Entscheide (1-25); Integer; 5; 1-25
Parameter3 = Minimale Relevanz der Entscheide; Double; 0.55
MergePrompt = Integriere den selektierten Auszug aus einem Bundesgerichtsentscheid als Zitat so in meinen Text, dass es diesem als Beleg mit Quellenangabe dient, wie dies in einer juristischen Fachschrift passen würde
```

76 Jede Parameter-Eintrag ist gleich aufgebaut und wird durch ein Semikolon getrennt:

- Beschreibung des Parameters (wird dem Benutzer angezeigt);
- Typ des Parameters (String, Boolean, Integer, Double);
- Default-Wert;
- Optional: Werte aus einem Drop-Down-Menü, getrennt durch Kommata (mit dem einzusetzenden Parameterwert in <...>) und im Falle eines Zahlenwerts der erlaubte Bereich (der Parameter wird dann automatisch limitiert).

77 Der erfasste Wert des Parameters wird dann im APICall-String an der betreffenden Stelle des Platzhalters (z.B. "{parameter2}") eingefügt. Auf diese Weise kann der Abfragebefehl für den Service individuell gesteuert werden. Die Antwort der Schnittstelle wird dann entweder im Dokument oder in der Pane eingefügt (im Falle einer Formatierung im Markdown-Format auch mit entsprechenden Formatierungen).

78 Sollte der einzusetzende Parameterwert selbst ";", ",", "<" oder ">" enthalten müssen, dann müssen diese Zeichen "Escaped" werden, d.h. sie sind im vorliegenden Fall mit zwei Backslashes zu versehen, also z.B. "\\" oder "\\.". Auf diese Weise können in den spitzigen Klammern auch komplexere JSON-Ausdrücke eingefügt werden (hier am Beispiel von LexiSearch):

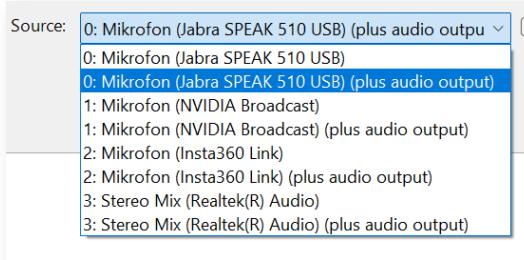
```
timeout = 200000
Parameter1 = Rechtsgebiet; String; Alle; Alle<>, Zivilrecht<civil>, Strafrecht<criminal>, Öffentliches Recht<public>
Parameter2 = Gerichte; String; Bundesgericht; Bundesgericht<["CH_BGE"\\", "CH_BGer"]>, Kanton Zürich["ZH_OG"\\", "ZH_HG"\\", "ZH_KG"]>, Alle <["CH_BGE"\\", "CH_BGer"\\", "ZH_OG"\\", "ZH_HG"\\", "ZH_KG"]>
Parameter3 = Max. Anzahl Entscheide (1-25); Integer; 5; 1-25
```

79 Es kann auch eine leere spitze Klammer eingefügt werden (wie im Beispiel ersichtlich); dann wird an der betreffenden Stelle ein leerer String eingefügt werden. Das ist auch der Fall, wenn als Text "(no selection)", "(keine Auswahl)" oder "---" steht. Verlangt die API hingegen, dass bei einer Wahl aller Optionen auch alle aufgeführt werden, sind

- diese entsprechend zu hinterlegen (siehe im Beispiel oben bei Parameter2 den Parameterwert für "Alle").
- 80 Der MergePrompt ist der Prompt, der bei Merge Selection angezeigt wird und vom Benutzer bearbeitet werden kann. Fehlt er, wird der Standard-MergePrompt verwendet, der entweder in der Konfigurationsdatei "redink.ini" hinterlegt ist ("SP_MergePrompt") oder sonst im Add-in vordefiniert ist. Dem Sprachmodell wird zusätzlich zum Merge-Prompt der Prompt im Parameter "SP_Add_MergePrompt" übergeben, mit den nötigen Hinweisen auf den einzufügenden Text und den Text des Hauptdokuments.
- 81 Der QueryPrompt ist der Prompt, mit welchem der Query Assistant die Suchbegriffe aus dem selektierten Text extrahiert. Er wird ohne weiteren Zusatz an das LLM gesendet. Er sollte darauf hinweisen, dass sich der Quelltext, aus welchem die Suchbegriffe zu ermitteln sind, nachfolgend zwischen den Tags <TEXTTOPROCESS> und </TEXTTOPROCESS> übergeben wird. Beispiel: *"MergePrompt = Extract from the TEXTTOPROCESS (provided to you between corresponding tags) precise and language-preserving search terms for the purpose of finding relevant court decisions addressing the same legal topic in a database of court decisions. Provide only the bare-bones search terms, separated by space, and nothing else, no wildcards, no quotes, no boolean operators, not comments, no commas."*
- 82 Der "Response"-Parameter kann komplexe Auswertungen enthalten, mit denen zugleich bestimmt werden kann, wie das Ergebnis in der Pane erscheint (im obigen Beispiel liefert der Dienst selbst einen passenden Markdown-formatierten Text, aber in vielen anderen Fällen ist das nicht der Fall). Siehe dazu Rz. 340 ff.
- G. Transcriptor**
- 83 Red Ink hat in Word eine Funktion für Transkriptionen eingebaut. Derzeit wird zur Umwandlung von Sprache in Text die Open-Source-Lösung **Vosk** mit ihren diversen "Speech-to-Text"-Modellen, die OpenAI-Spracherkennung "**Whisper**" sowie die Cloud-basierten STT-Modelle von **Google** (V1, via Vertex) unterstützt (siehe nachfolgend). Sobald die Modelle konfiguriert sind, steht die Funktion Transcriptor im Kachelmenü zur Verfügung. Das kann zum Beispiel benutzt werden, um Sitzungen zu protokollieren (auch Online), um Texte zu diktieren oder um Videos oder Vorträge zu transkribieren (z.B. um eine Zusammenfassung zu erstellen).
- 84 Die Live-Transkription-Funktion wurde bewusst so programmiert, dass das **Tonsignal nicht abgespeichert** wird, sondern laufend an die Spracherkennung weitergeleitet wird. Der Benutzer hat also via Red Ink keinen Zugriff auf das Audiosignal und kann es sich auch nicht wieder anhören (auch nicht über eine Temporärdatei – die Verarbeitung findet ausschliesslich im Arbeitsspeicher statt, wie sie beispielsweise die Videokonferenzlösung selbst auch macht). Das ist wichtig,

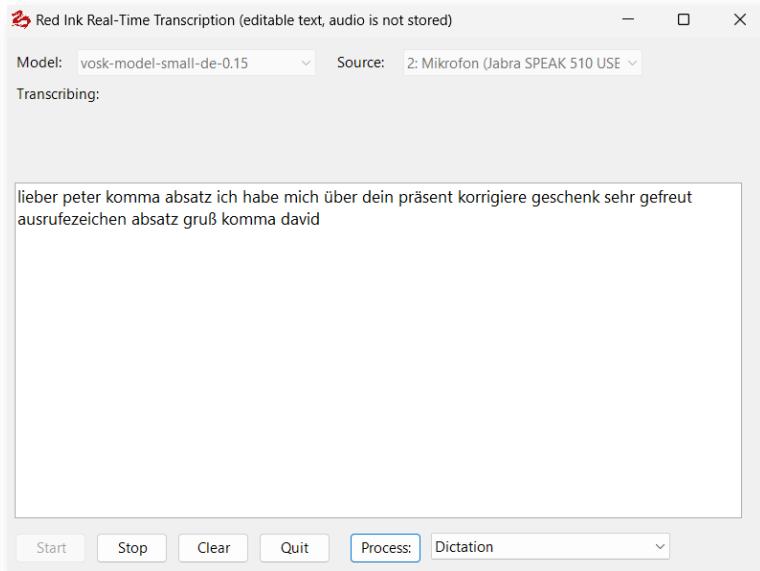
weil davon je nach Rechtsordnung abhängen kann, ob der Benutzer für die Verwendung der Live-Transkription die Einwilligung der anderen Gesprächsteilnehmer einholen muss (in der Schweiz ist es nach Art. 179ter StGB den Teilnehmern eines nichtöffentlichen Gesprächs ohne Einwilligung der anderen nur untersagt, dieses auf einen Tonträger aufzunehmen).

- 85 Wird die Transcriptor-Funktion gewählt, geht ein Fenster auf, in welchem oben zuerst das Modell (d.h. die Sprache) gewählt werden muss und die **Audio-Quelle**. Es kann hier das Mikrofon gewählt werden, jeweils mit und ohne Audio-Output. Ist mit Audio-Output gewählt, wird der Transcriptor neben dem Mikrofon-Input auch noch die Signale verarbeiten, die aus dem in Windows konfigurierten Hauptausgabegerät (d.h. der normalerweise verwendete Lautsprecher) übermittelt werden; mit dem Knopf "**Dev**" (für Device) kann das Audio-Output-Gerät vom vorkonfigurierten Hauptausgabegerät auf ein anderes geändert werden (zum Beispiel auf jenes, das für Videokonferenzen benutzt wird). Für das Transkribieren von Videokonferenzen sollte entweder das Mikrofon mit Audio-Output ("(plus audio output)", siehe Bild unten) oder aber "Stereomix" oder "Stereo-Mix" gewählt werden, da dies die eigene Stimme des Mikros und den Ton der anderen Teilnehmer kombiniert. Allerdings unterstützen das nicht alle Computersysteme. Es ist also nicht garantiert, dass der Transcriptor wirklich alle Inputs hört. Dies kann z.B. daran liegen, dass eine Videokonferenzsoftware z.B. ein Mikrofon exklusiv für sich "bucht" und sich der Transcriptor nicht mehr einklinken kann. Die Möglichkeit, Audio-Geräte exklusiv zu buchen kann in Windows auch deaktiviert werden. Was in der Praxis auch funktionieren kann ist, dass eine Videokonferenz über Lautsprecher gehört wird, der Sprecher das eine Mikrofon (z.B. im Lautsprecher) benutzt, während ein anderes Mikrofon den ganzen Dialog aufnimmt und er von dort transkribiert wird.

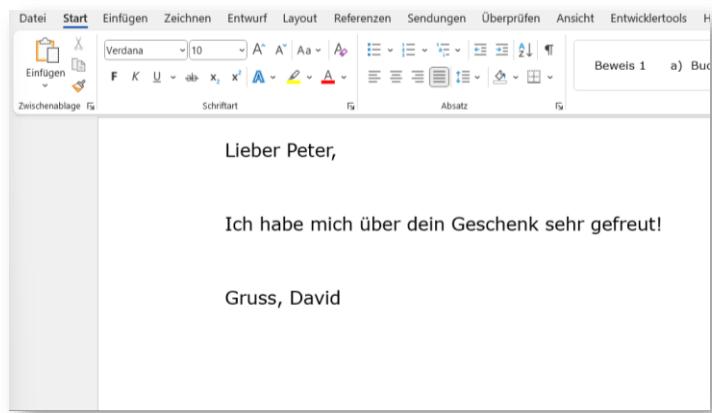


- 86 Ist die Konfiguration vorgenommen, kann die Transkription mit "**Start**" gestartet und mit "**Stop**" wieder gestoppt werden. Ist sie gestartet, erscheint der Text, sobald das Tonsignal von der Spracherkennung ausgewertet ist. Hier verhalten sich Vosk und Whisper unterschiedlich. Vosk zeigt an, wie er die Sätze aufbaut, während er sie hört. Dies ist im oberen Bereich unter "Transcribing" zu sehen. Hat er genug zusammen, erscheint der erkannte Text im darunterliegenden Fenster und kann dort auch während der Transkription frei bearbeitet werden.

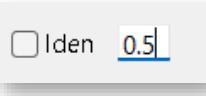
Whisper liefert seinen Text jeweils erst, wenn er ein Stück fertig transkribiert hat und braucht (je nach Modell und Rechenleistung des Geräts) auch länger, d.h. es muss länger gewartet werden, bis etwas erscheint. Es kann dadurch sein, dass die Transkription noch ein Stück weiterläuft, nachdem das Gespräch schon vorbei ist. In diesem Fall einfach weiterlaufen lassen. Wird frühzeitig "Stop" gedrückt, kann dies die Transkription abbrechen.



- 87 Bei Vosk muss jeweils das für **die Sprache abgestimmte Modell** gewählt werden. Weil Whisper mit multilingualen Modellen arbeiten kann, ist dies dort nicht nötig. Jedoch wird vor der Transkription in diesem Fall noch zusätzlich gefragt, welche Sprache gesprochen wird. Ist dies nicht bekannt oder soll Whisper dies selbst herausfinden, ist "auto" einzugeben, ansonsten der Zwei-Buchstaben-ISO-Code für die betreffende Sprache, also "en" für Englisch, "de" für Deutsch oder "fr" für Französisch. Wird die Sprache angegeben, ist das Ergebnis der Spracherkennung erfahrungsgemäss besser. Schweizer-Deutsch ist keine offizielle Sprache, wird aber von den grösseren Whisper-Modellen (Medium, Large) unter "de" unterstützt. Bei Google muss in einer Auswahl die gewünschte Sprachcodierung gewählt werden.
- 88 Soll die aktuelle **Transkription ins aktuelle Dokument in Word übertragen** werden, kann der Knopf "Process" benutzt werden, wobei in der Auswahl rechts davon gewählt werden soll, wie die KI den Text bearbeiten und bereinigen soll, also ob sie ihn z.B. als Diktat bearbeiten soll, wie im Beispiel oben, ob eine Zusammenfassung, ein Protokoll oder eine Aufgabenliste erstellt werden soll. Die Verarbeitungsmöglichkeiten basieren auf Prompts, die der Benutzer in einer separaten Prompt-Bibliothek (die genau gleich aufgebaut ist wie die normale Prompt-Bibliothek) definieren kann. Einige Beispiele werden mitgeliefert. Wird Diktat verwendet, erscheint in Word folgender Text (d.h. es werden auch gesprochene Kommandos berücksichtigt):



- 89 Die Process-Funktion verarbeitet den im Transkriptionsfenster selektierten Text oder, wenn nichts selektiert ist, den gesamten Text im Fenster. Die Process-Funktion kann während laufender Transkription benutzt werden. Wer Text ohne Process-Funktion übernehmen will, kann ihn selektieren und in die Zwischenablage kopieren und am Ziellort einfügen. Bei der Ausführung der Process-Funktion wird dem Sprachmodell jeweils auch das aktuelle Datum mitgegeben, damit es dieses für Protokolle berücksichtigen kann.
- 90 Es ist übrigens möglich, Transkripte aus anderen Programmen in das Fenster hineinzukopieren, um es von der KI verarbeiten zu lassen. Wenn eine **bestehende Aufnahme transkribiert** werden soll, ist auch dies möglich. Hierzu ist der Befehl "**Load**" gedacht: Es geht ein Fenster auf, in das die Datei hineingezogen werden kann. Danach startet die Transkription mit dem gewählten Modell unmittelbar. Da hier mehr Zeit zur Verfügung steht, können auch grössere Modelle verwendet werden, die mehr Rechenleistung benötigen. Bei Google kann gewählt werden, ob die Daten aus der Datei häppchenweise an die KI gehen oder laufend wie ein Live-Gespräch übermittelt werden.
- 91 Die Spracherkennung von Vosk und Google unterstützt zumindest theoretisch auch die **Erkennung von unterschiedlichen Sprechenden** (Whisper nicht), die sog. *Speaker Diarization*. Sie erkennt sie an Stimmunterschieden, weiss aber natürlich nicht, wer wer ist. Wer diese Funktion nutzen will, muss ein zusätzliches Modell installieren und kann die Identifikation der Sprecher durch die Checkbox "Iden" oben rechts aktivieren:



- 92 Ist sie aktiviert, erscheint bei Vosk vor jedem Text der identifizierte "Speaker" mit einer Nummer. Mit dem Wert hinter der Checkbox (von 0.5 bis 2.5) kann angegeben werden, wie stark sich die Sprechenden unterscheiden müssen, damit das System sie auseinanderhält. Ist der Wert zu tief, werden unterschiedliche Sprechende zusammengefasst,

ist der Wert zu hoch, hält das System denselben Sprechenden für unterschiedliche Personen:

- 0.5 - 0.7: Sehr nachsichtig – selbst geringe Unterschiede in der Stimme werden als derselbe Sprecher behandelt (nützlich für laute Umgebungen);
- 0.7 - 1.2: Ausgewogen – ein guter Bereich zur Sprecherunterscheidung unter normalen Bedingungen (eignet sich für Sitzungen);
- 1.2 - 1.8: Streng – nur Texte mit einem sehr ähnlichen Stimmuster werden als derselbe Sprecher gruppiert (höhere Werte sind nicht empfohlen).

Nach unserer Erfahrung ist die Qualität der Erkennung der Sprecher vor allem bei Videokonferenzen nicht gut. Oft ist es praktischer, sie einfach auszuschalten und ein Protokoll ohne Zuweisung zu den Personen erstellen zu lassen.

- 93 Bei Google funktioniert die Sprechererkennung im Prinzip auch, aber sie läuft etwas anders ab. In der Box neben Iden ist die maximale Anzahl an Sprecher anzugeben, die vorkommen werden (also z.B. "3" falls drei Sprecher erwartet werden). Die Spracherkennung von Google funktioniert so, dass Google den finalen Text erst bei einer Rede-Pause ausgibt. Zudem passt Google den finalen Text bei Bedarf an. Es sollte also während dem Transkribieren mit Google mit Sprecher-Erkennung nichts im Ergebnisfenster geändert werden. Auch sollte die Transkription erst gestoppt werden, wenn die Sprecher aufgehört haben zu reden, weil sonst der letzte Teil des bereits erkannten Textes verloren gehen kann. Wir haben mit der Sprechererkennung bei Google bisher allerdings keine guten Erfahrungen gemacht und benutzen den Transcriptor normalerweise ohne (die "Process"-Funktion kann trotzdem relativ gut den Text den einzelnen Sprecher zuordnen, wenn sie genannt werden).
- 94 Ist ein **Whisper**-Modell gewählt, ist anstelle der Sprecher-Identifikation eine **automatische Übersetzung auf Englisch** möglich, d.h. eine Art Simultan-Dolmetschen. Dort, wo bei den Vosk-Modellen der Schalter für die Identifikation von Sprechenden erscheint, wechselt der Schalter bei Whisper-Modellen auf "Trans" für "Translation" (Übersetzung). Ferner kann im Eingabefeld rechts daneben bei Whisper angegeben werden, wie empfindlich die Spracherkennung sein soll, um Sprache von Hintergrundgeräuschen zu unterscheiden. Ein mittlerer Wert liegt bei 0.6, und in geräuschreicher Umgebungen wird ein höherer Wert empfohlen (0.7-1.0).
- 95 Noch nicht unterstützt werden teilweise Satzzeichen und Gross- und Kleinschreibung. Die Gross- und Kleinschreibung und Interpunktions ist jedoch insofern kein Hindernis, als die Process-Funktion diese Dinge ausgleicht.

- 96 Läuft die Transkription, versucht Red Ink den Computer so einzustellen, dass er nicht mehr in den **Schlafmodus** fallen kann. Diese Einstellung wird nach Ende der Transkription wieder rückgängig gemacht.
- 97 Zur **Installation**: Wer den Transcriptor nutzen will, muss sich zuerst ein passendes Vosk-Modell herunterladen und die Konfigurationsdatei von Red Ink darauf einstellen (zur Konfigurationsdatei siehe Rz. 309 ff. unten). Zu beachten ist, dass die Transkription nicht auf allen Geräten funktioniert, was verschiedene Gründe haben kann (z.B. fehlende Leistung, fehlende Inputs). Sie wird auch unterschiedlich gut funktionieren.
- 98 Modelle können auf der Website <https://alphacephal.com/vosk/models> als ZIP-Datei kostenlos bezogen werden. Der Inhalt der ZIP-Datei (es ist dies ein Verzeichnis mit mehreren Unterverzeichnissen) kommt in ein lokales Verzeichnis und der Pfad zu diesem lokalen Verzeichnis wird mittels Parameter "SpeechModelPath" in der Konfigurationsdatei von Red Ink hinterlegt. Das sieht dann beispielsweise so aus:
- ```

 └── vosk-model-small-de-0.15
 ├── vosk-model-small-en-us-0.15
 └── promptlib-transcript.txt

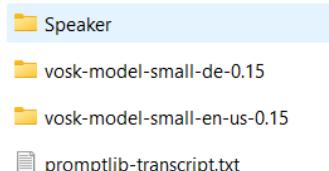
```
- 99 Die **Whisper Modelle** sind ebenfalls kostenlos zu beziehen. Für den Transcriptor sind allerdings jene Modelle nötig, die auf C++ portiert worden sind (eine Programmiersprache), damit sie mit höherer Leistung und Effizienz genutzt werden können, insbesondere auf Geräten mit wenig Ressourcen. Es gibt sie unter der Adresse <https://huggingface.co/ggerganov/whisper.cpp/tree/main>. Der Name der Dateien beginnt jeweils mit "ggml" (das "q" im Dateinamen verweist auf Quantized-Modelle, ein Verfahren zur Reduktion des Speicherbedarfs). Die Dateien müssen lediglich im "SpeechModelPath"-Verzeichnis abgelegt werden:
- ```

    └── ggml-base-q8_0.bin
    └── ggml-small.bin
    └── ggml-tiny-q8_0.bin

```
- Neben den Whisper-Modellen ist es auch erforderlich, die **Runtime-Bibliotheken** (d.h. einige Programmdateien) von Whisper.net (Quelle: <https://github.com/sandrohanea/whisper.net>) im selben Verzeichnis zu speichern. Diese kommen in ein Unterverzeichnis mit dem Namen "runtimes" und sind im Installationspaket enthalten (mit Unterverzeichnis "runtimes", so dass es direkt mittels Drag & Drop ins Verzeichnis gezogen werden kann). Das Installationspaket kann unter <https://apps.vischer.com> heruntergeladen werden.
- 100 Sowohl bei Vosk als auch bei Whisper sollten nur **die "kleinen" Modelle für die Live-Transkription** benutzt werden, da die grossen Modelle die Rechenleistung eines normalen Arbeitsplatzcomputers über-

fordern. Ist das Modell zu gross, dauert die Transkription auch zu lange bzw. erfolgt verzögert.

- 101 Im selben Verzeichnis kann auch die **Prompt-Bibliothek** für die Verarbeitung der Transkripte hinterlegt werden. Sie wird über den Parameter "PromptLib_Transcript" in der Konfigurationsdatei hinterlegt. Es ist der volle Pfad mit Dateiname anzugeben. Pfadangaben können immer auch Platzhalter enthalten (Rz. 319 unten). Die Prompt-Bibliothek für Transkripte verwendet dieselbe Syntax wie die normale Prompt-Bibliothek (Rz. 324 ff. unten).
- 102 Wird die **Sprecheridentifikation** bei Vosk benutzt, muss ein zusätzliches Unterverzeichnis "Speaker" geschaffen werden, in welches dann das Modell (d.h. das Verzeichnis mit dem Modell) hineinkopiert wird (also z.B. vosk-model-spk-0.4; auch dieses ist auf der obengenannten Website verfügbar):



- 103 Der Zugriff auf **Google** ist bereits vorkonfiguriert, da diese Spracherkennung Cloud-basiert erfolgt, d.h. nicht auf dem lokalen Rechner. Red Ink ist so konfiguriert, dass es die Rechenzentren von Google in der EU für die Spracherkennung benutzt. Die Google-Spracherkennung steht nur zur Verfügung, wenn das primäre oder sekundäre Modell für die Vertex-API von Google und folglich ein OAuth-Verfahren konfiguriert ist (es werden dann dieselben Zugangscodes verwendet). Ist dies der Fall, erscheint Google automatisch in der Auswahl der Modelle. Beim Start der Transkription muss die jeweilige Sprache gewählt werden.
- 104 In einer künftigen Version werden wir noch Modelle von weiteren Anbietern versuchen anzubieten.

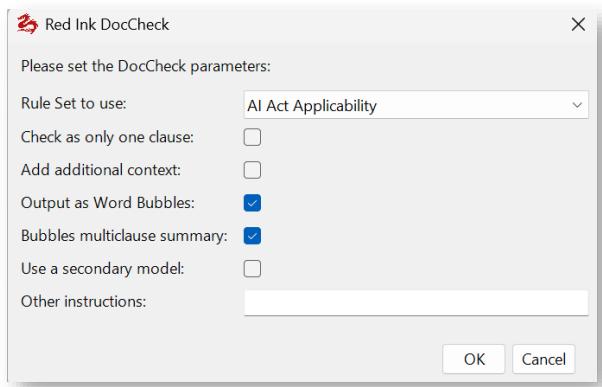
H. Document Check

- 105 Red Ink ist in der Lage, **Word-Dokumente** (bzw. die selektierten) Teile nach vordefinierten Kriterien-Katalogen **von der KI prüfen zu lassen**. Die Kriterien-Kataloge enthalten eine beliebige Anzahl von Prüfkriterien, nach denen ein Vertrag, eine Datenschutzerklärung, ein Projektvorhaben etc. geprüft werden muss (z.B. ob eine bestimmte Regelung enthalten ist oder ein bestimmter Anwendungsfall vorliegt). Diese Prüfkriterien sind in einer separaten Text-Datei hinterlegt, wahlweise zusammen mit dem Prompt, der zur Prüfung verwendet wird.
- 106 Die Funktion kennt zwei Prüfmethoden:
- **Mehrere Klauseln:** Bei dieser Methode wird jedes Prüfkriterium auf jeweils den gesamten selektierten Text angewandt, also z.B. geprüft, ob ein Vertragstext eine bestimmte Regelung enthält.

Der gesamte Text wird also jeweils gegen ein einzelnes Prüfkriterium geprüft.

- **Eine Klausel:** Bei dieser Methode wird geprüft, ob die selektierte Klausel einem oder mehreren der Prüfkriterien entspricht. Diese Prüfung erfolgt also gerade umgekehrt – eine Klausel gegenüber allen Prüfkriterien.

107 Die Funktion wird über "Analyze" und dort "**Document Check**" aufgerufen. Es erscheint die Konfigurationsseite:



108 Es kann ausgewählt werden:

- **Rue Set to use:** Diese werden aus der konfigurierten zentralen und lokalen Ablage aus den dort vorhandenen Rule-Set-Dateien zusammengetragen und zur Auswahl angeboten;
- **Check as only one clause:** Wenn ausgewählt, wird die Methode "Eine Klausel" gewählt, soweit das Rule Set diese Methode überhaupt zulässt;
- **Add additional context:** Wenn ausgewählt, dann können der KI zur Prüfung nebst dem selektierten Text auch weitere Word-Dokumente als Kontext übergeben werden; diese müssen offen sein, damit Red Ink sie verwenden kann (der Benutzer kann sie dann auswählen);
- **Output as Word bubbles:** Wenn ausgewählt, dann wird Red Ink die Antwort in Form von Word-Kommentaren liefern, ansonsten als Bericht in einem separaten Fenster;
- **Bubbles multiclause summary:** Falls angewählt, wird nach Abschluss der Prüfung des Dokuments ein Word-Kommentar ganz zu Beginn der selektierten Stelle angefügt, der eine Zusammenfassung des Ergebnisses enthält (hierfür wird der Prompt "SP_DocCheck_MultiClauseSum_Bubbles" benutzt). Bei einem Bericht wird der Prompt "SP_DocCheck_MultiClauseSum" benutzt, um aus den einzelnen Beurteilungsergebnissen ein Bericht zu erstellen, was zugleich auch für die Erstellung einer Zusammenfassung genutzt werden kann;

- **Use a secondary model:** Dieser Punkt ist auszuwählen, um statt dem primären das sekundäre KI-Modell für die Prüfung zu verwenden oder – falls konfiguriert – eines der alternativen Modelle. Die Selektion erfolgt im nächsten Schritt;
- **Other instructions:** Hier können weitere Instruktionen an die KI übergeben werden. Sie werden im Prompt beim Platzhalter "{OtherPrompt}" eingefügt, falls er angegeben ist.

- 109 Sind die Parameter erfasst und wird OK gedrückt, werden die weiteren Angaben je nach Konfiguration abgefragt und die Prüfung beginnt. Ist kein Text selektiert, wird der gesamte Text selektiert.
- 110 Nach den Kriterien-Katalogen (den sog. **Rule Sets**) sucht Red Ink an den beiden Pfaden, die in der Konfigurationsdatei hinterlegt sind ("DocCheckPath" und "DocCheckPathLocal", vgl. Rz. 309 ff.). Diese Aufteilung ist für Betriebe gedacht, in denen mehrere Personen das Tool einsetzen. Der erste Pfad kann für gemeinsame Rule Sets benutzt werden (z.B. auf einem Netzwerklaufwerk), der zweite Pfad für persönliche Rule Sets (z.B. auf einem lokalen Laufwerk). Die Rule Sets sucht Red Ink in Script-Dateien mit der Bezeichnung "**redink-dc-*.txt**", d.h. alle Dateien mit dieser Signatur werden in den beiden Verzeichnissen eingelesen und nach Rule Sets geprüft. Jede Datei kann dabei mehrere Rule Sets enthalten. Sie sind jeweils durch eine Titelzeile in eckigen Klammern getrennt, welche den Titel bzw. die Bezeichnung des Rule Set umfasst, wie im Beispiel "[AI Act Applicability]":

```
; AI Act
; RedInk DocCheck Script - david.rosenthal@vischer.com - 13.9.2025

SP_DocCheck_MultiClause = You task is to find out whether the EU AI Act applies to a
Usecase MEETS the Project, create a brief report: (a) provide the portion of the Proj
ngle portion of the text. \n- NEVER provide a response where the Project does not mat

SP_DocCheck_Clause = X

[AI Act Applicability]

{
  "Records": [
    {
      "Usecase": "AI deploys subliminal techniques beyond a person's consciousness or",
      "Consequences": "Prohibited under AI Act Art. 5(1)(a)."
    },
    {
      "Usecase": "AI exploits vulnerabilities due to age, disability, or a specific s",
      "Consequences": "Prohibited under AI Act Art. 5(1)(b)."
    },
    {
      "Usecase": "AI performs social scoring over time based on social behaviour or k",
      "Consequences": "Prohibited under AI Act Art. 5(1)(c)."
    }
  ]
}
```

- 111 Jedes Rule Set besteht aus einem sog. **JSON-Array** bzw. einzelnen JSON-Records (wie im obigen Beispiel ersichtlich), d.h. ist in einer Syntax verfasst, die ein LLM besonders gut als fixe Struktur verarbeiten kann. Wichtig ist, dass jedes Kriterium in einem eigenen Datensatz der Struktur enthalten ist, da Red Ink diese Datensätze bei der Methode mit mehreren Klauseln nacheinander abarbeitet, d.h. jeweils ein Krite-

rium extrahiert und zusammen mit dem selektierten Text der KI über gibt, damit diese das Kriterium anhand des Textes analysiert. Wie der Datensatz strukturiert ist, spielt für das Add-in hingegen keine Rolle. Im obigen Beispiel besteht jeder Datensatz z.B. aus einem Feld "Use-case" und einem Feld "Consequences" sowie einem entsprechenden Text-Wert. Es sind auch andere Strukturen möglich. Entscheidend ist, dass der Prompt, der für die Analyse benutzt wird, der KI erklärt, welche Inhalte (z.B. Kriterien, Folgen, Bedingungen, Empfehlungen) in der Struktur enthalten ist und wie sie den ihr vorgelegten Text entsprechend prüfen soll. Daher können in jeder Rule Set-Datei auch die beiden dazugehörigen Prompts abgelegt werden, und zwar bezeichnet als "SP_DocCheck_Multiclause" und "SP_DocCheck_Clause" (sowie zur Zusammenführung bei Berichten "SP_DocCheck_MultiClauseSum" und für eine Zusammenfassung der Ergebnisse bei Word Kommentaren "SP_DocCheck_MultiClauseSum_Bubbles"). Analog zur Konfigurationsdatei folgt nach dem Parameter ein "=" -Zeichen und dann auf einer einzigen Zeile der Prompt. Soll die Prüfung nach der Methode der einzelnen Klausel für ein bestimmtes Rule Set (oder alle Rule Sets der Datei) nicht möglich sein, ist stattdessen ein "X" anzugeben (wie im obigen Beispiel). Steht die Zeile mit dem Prompt vor dem ersten Titel in eckigen Klammern, gilt der Prompt für alle Rule Sets in dieser Datei (wie im obigen Beispiel). Soll ein Prompt nur für ein bestimmtes Rule Set gelten, ist es nach dem betreffenden Titel einzufügen.

112 Ein Rule Set kann also wie folgt aufgebaut sein:

- { "Records": [{record}, {record}, ...] }
- [{record}, {record}, ...]
- {record}

113 Wird kein Prompt in der Rule Set-Datei angegeben, dann verwendet Red Ink den **Standard-Prompt** für die beiden Prüfmethoden und geht von folgender Struktur eines Rule Sets aus:

```
[Sponsoringvertrag]
{
  "Records": [
    {
      "Topic": "Auftritts- und Nennungsrechte",
      "Issue": "Rechte zur Nennung und Darstellung des Sponsors",
      "Criteria": [
        {
          "Condition": "Sponsor darf als offizieller Sponsor auftreten und ausschließlich definierte Bezeichnungen verwenden",
          "IfTrue": { "Consequence": "", "Risk": 0 },
          "IfFalse": { "Consequence": "Ergänzen: Sponsor muss diese Rechte explizit haben", "Risk": 3 }
        },
        {
          "Condition": "Definition, was als 'Auftritt als Sponsor' gilt, inklusive Nutzung von Namen, Logos und Bezugnahmen",
          "IfTrue": { "Consequence": "", "Risk": 0 },
          "IfFalse": { "Consequence": "Definition ergänzen, um Missverständnisse zu vermeiden", "Risk": 2 }
        },
        {
          "Condition": "Sponsor darf nur die in Anhang XY definierten Logos und Composite-Logos verwenden",
          "IfTrue": { "Consequence": "", "Risk": 0 },
          "IfFalse": { "Consequence": "Anhänge mit expliziter Logoverwendung hinzufügen", "Risk": 3 }
        },
        {
          "Condition": "Sponsor darf Logos nicht ohne Zustimmung der Sportorganisation mit Drittangeboten oder Werbung nutzen",
          "IfTrue": { "Consequence": "", "Risk": 0 },
          "IfFalse": { "Consequence": "Klausel zur Zustimmungspflicht ergänzen", "Risk": 3 }
        }
      ]
    }
  ]
}
```

114 Neutral dargestellt ist das **Standard-Record-Schema** somit wie folgt:

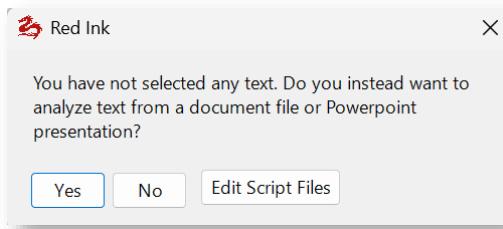
{

```

    "Topic": "String",
    "Issue": "String",
    "Criteria": [
      {
        "Condition": "String",
        "IfTrue": { "Consequence": "String", "Risk": 1..3 },
        "IfFalse": { "Consequence": "String", "Risk": 1..3 }
      },
      ...
    ]
  }
}

```

- 115 Es gibt somit jeweils ein Feld "Topic" und "Issue" und dann mehrere "Criteria", nach denen geprüft wird. Die "Criteria" formen wiederum ein JSON-Array, d.h. sind ein Set von Datensätzen, die jeweils aus einer "Condition" (Bedingung) und einer "Consequence" und "Risk" für den Fall versehen sind, dass die "Condition" erfüllt ist ("IfTrue") oder nicht erfüllt ist ("IfFalse"). Pro Prüfschritt wird jeweils ein Topic-Datensatz übergeben (d.h. er enthält jeweils 1-n "Condition"-Prüfbedingungen).
- 116 Soll **dem Benutzer einen Hinweis angezeigt** werden, wohin er sich bei weiteren Fragen wenden soll (z.B. an die interne Rechtsabteilung), so kann dies für die ganze Datei (d.h. für alle Rule Sets) oder für ein einzelnes Rule Set mit dem Parameter "Notice = *Hinwestext*" eingebaut werden, abhängig davon, wo der Parameter eingefügt wird (ganz zu Beginn oder nach einem Titel). Der Hinwestext wird als Word-Kommentar am Ende des Textes oder im Bericht am Ende eingefügt (und in der Option Word-Kommentare auch am Ende angezeigt).
- 117 Mittels Document Check kann nicht nur das aktuell geöffnete Word-Dokument geprüft werden, sondern auch **PDF- und andere Dokumenten** (Word, Text-Dateien diverser Formate) sowie **Powerpoint-Dateien**. In diesem Fall sollte kein Text selektiert sein, wenn die Funktion "Document Check" aufgerufen wird. Der Benutzer wird gefragt:



- 118 Klickt der Benutzer auf "Yes", geht ein Fenster auf, in welches die zu überprüfende Datei hineingezogen (oder geöffnet) werden kann. In diesen Fällen steht die Verwendung von Word-Kommentaren allerdings nicht zur Verfügung, d.h. es wird immer (in Word) ein Bericht mit den Ergebnissen generiert.
- 119 Der Benutzer kann die Abfrage übrigens auch benutzen, um die ihm zugänglichen **Script-Dateien** zu **bearbeiten**. Hierzu die Option "Edit Script Files" auswählen und es erscheint die Liste der verfügbaren Script-Dateien (nicht die einzelnen Rule Sets). Wird eine ausgewählt,

kann sie in einem einfachen Editor bearbeitet und die Änderungen gespeichert werden. Sie werden sofort aktiv.

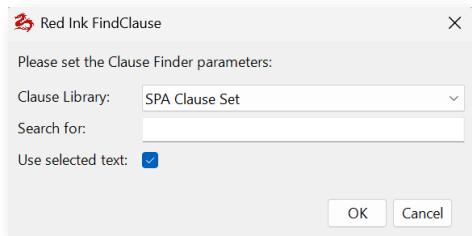
I. Klauseldatenbank

- 120 Red Ink bietet die Möglichkeit, häufig gebrauchte **Klauseln** in einem Textdokument als **einfache Datenbank abzulegen** und in Word inner Sekunden nach Kontext daraus abzurufen. Das kann beispielsweise beim Entwerfen von Verträgen helfen, aber auch für beliebige andere Textbausteine benutzt werden. Die Funktion ist einfach: Alle Klauseln werden in einer Textdatei in einer bestimmten Struktur abgelegt und zentral oder auf dem eigenen Computer abgespeichert. Wird eine Klausel benötigt, kann die Funktion aufgerufen werden und es kann ein Thema als Suchbegriff eingegeben werden. Alternative kann nach einer Klausel passend zum markierten Text gesucht werden. Danach zeigt das Add-in in einer Pane die passenden Klauseln an. Sie können selektiert und in das Hauptdokument kopiert oder mit diesem zusammengeführt werden (wofür ein eigener Merge-Prompt definiert werden kann). Die Klauseln werden mittels KI aus dem Textdokument extrahiert.
- 121 Aufgerufen wird die Funktion "**Find Clause**" über das "Analyze" Menü. Sie setzt voraus, dass in der Konfigurationsdatei ein Pfad zu einem Verzeichnis für entsprechende Klauseldatenbanken definiert worden ist. Es kann sowohl ein zentrales als auch ein lokales Verzeichnis konfiguriert werden (Parameter "FindClausePath" und "FindClausePathLocal", vgl. Rz. 309 ff.). Damit ist es möglich, sowohl in einem Unternehmen oder einer Gruppe zusammen geteilte Klauseldatenbanken zu nutzen als auch solche, auf die nur jeweils der Benutzer zugriff hat (gekennzeichnet an der Bezeichnung "(local)").
- 122 Wird die Funktion aufgerufen, sucht sie in diesen Verzeichnissen nach Dateien, die den Namen "**redink-lib-*.txt**" tragen, wobei der Stern für einen beliebigen, als Dateinamen zulässigen Text stehen kann. Jedes dieser Dokumente kann eine oder mehrere Klauseldatenbanken enthalten. Das System ist das gleiche wie beim Document Check: Jede dieser Textdokument (reiner ASCII Code) enthält ein oder mehrere **Segmente**, wobei der Beginn jeweils mit einem Titel in eckigen Klammern markiert wird (z.B. "[Klauseln für käuferfreundliche Aktienkaufverträge]") und danach folgt eine JSON-Struktur, in welcher die Klauseln abgelegt sind. Pro Segment können die **Prompts** zur Abfrage der Klauseln ("SP_FindClause") und später für das Zusammenführen mit dem Text unter Verwendung der Merge-Funktion ("SP_MergePrompt") segmentspezifisch definiert werden oder aber, wenn sie ganz vorne eingefügt werden, für die gesamte Datei spezifisch festgelegt werden. Wird kein Prompt eingegeben, werden die Standardprompts verwendet. Für weitere Ausführungen vgl. Rz. 110 ff. oben. Die JSON-Struktur ist allerdings viel einfacher als bei Document Check. Es genügt für jede Klausel ein Eintrag mit der Property "clause" und einem zugewiesenen Text, wie in diesem Beispiel ersichtlich:

```
{
  "clause": "Construction\nUnless the context otherwise requires, words denoting the singular shall include the plural and vice versa and references to any gender shall include all other genders.\nWhenever the words \"include\", \"includes\", \"including\" and \"in particular\" are used in this Agreement, they shall be deemed to be followed by the words \"without limitation\".\nAny document that must be delivered in \"writing\" or \"written\" form must be signed in accordance with article 14 of the CO.\n\nThe word \"or\" is to be construed as inclusive disjunction.\nA reference to CHF is to Swiss Francs, to USD is to United States Dollars and EUR is to Euro."
}

{
  "clause": "Object of Purchase\nSubject to the terms and conditions of this Agreement,\n[the Sellers hereby sell to the Buyer and the Buyer purchases from the Sellers the [Shares][Object of Purchase].]\n[each Seller hereby sells to the Buyer the Shares as set forth in Annex C and the Buyer purchases from the Sellers the Shares as set forth in Annex C.]"
}
```

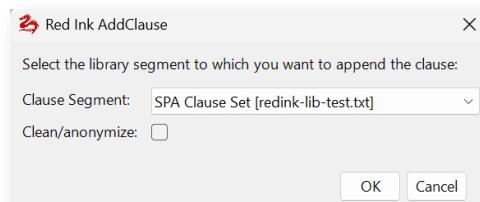
- 123 Zeilenschaltungen werden mit "\n" markiert, und gewisse Zeichen (wie das Anführungszeichen) werden "escaped", d.h. mit einem Code versehen, damit es zu keinem Durcheinander mit der JSON-Syntax kommt. **Formatierungen** sind auch denkbar, müssen dann aber im Markdown-Format erfasst werden (fetter Text z.B. durch klammernde Doppelsterne: "***text***").
- 124 Diese Klauseln können in einem Array oder einfach **als JSON-Records** hintereinander innerhalb des betreffenden Segments abgelegt werden (d.h. nach der Titelzeile mit den eckigen Klammern und vor der nächsten Titelzeile).
- 125 Wird ein **individueller Prompt** verwendet, können natürlich auch komplexere JSON-Strukturen abgebildet werden, die auch weitere Kriterien enthalten. Dem Prompt angehängt werden bei der Suche zwischen den Tags "<LIBRARY>" und "</LIBRARY>" die gesamte JSON-Struktur, zwischen den Tags "<SEARCHQUERY>" und "</SEARCHQUERY>" ein etwaiger Suchbegriff (der auch aus mehreren Wörtern bestehen kann) und zwischen den Tags "<TEXTFORSEARCH>" und "</TEXTFORSEARCH>" ein allfällig selektierter Text, wobei der Benutzer beim Aufruf von "Find Clause" wie oben erklärt wählen kann, was tatsächlich übergeben wird.
- 126 Wird die Funktion "Find Clause" aufgerufen, erscheint das folgende Parameter-Fenster:



- **Clause Library:** Hier kann die Klauseldatenbank (d.h. das Segment) gewählt werden, das durchsucht werden soll.

- **Search for:** Hier kann ein (auch mehrteiliger) Suchbegriff eingegeben werden; der Suchbegriff muss nicht genau so in der Klausel vorkommen um gefunden zu werden, vielmehr wird nach Klauseln gesucht, die der Bedeutung des Suchbegriffs möglichst nahe kommen.
- **Use selected text:** Falls ein Text selektiert worden ist, kann auch dieser als Suchkontext verwendet werden, z.B. um eine Klausel passend zu einer bestehenden Klausel zu finden. In diesem Fall sollte der Suchbegriff leer gelassen werden (muss aber nicht).

- 127 Wenn **alternative Sprachmodelle** definiert worden sind (Rz. 39), dann sucht die Funktion zuerst, ob dort eines für die FindClause-Funktion ausgewählt worden ist (mit dem Parameter "FindClause = True" im betreffenden Segment der Konfigurationsdatei). Falls ja, wird dieses verwendet für die Abfrage. Auf diese Weise ist es möglich, ein einfacheres, aber schnelleres Modell als das Hauptmodell zu verwenden, um die Wartezeit zu reduzieren.
- 128 Die betreffende Klauseldatenbank kann manuell erweitert werden (durch Bearbeitung der Textdatei mit einem Editor oder mit Word) oder weitere Klauseln können mit Hilfe der "**Add Clause**-Funktion, ebenfalls zugänglich via "Analyze"-Menü, hinzugefügt werden. Hierzu wird die betreffende Klausel (sie kann auch aus mehreren Absätzen bestehen) ausgewählt und "Add Clause" aufgerufen. Es erscheint folgendes Parameter-Fenster:



- 129 Es wird die Klauseldatenbank ausgewählt, zu welcher der markierte Text als Klausel am Ende angefügt werden soll und es kann angegeben werden, ob die Klausel vorab durch die KI **anonymisiert und bereinigt** werden soll. Hierfür wird der vordefinierte Prompt "SP_FindClause_Clean" benutzt. Er behebt auch Schreibfehler. Das Ergebnis wird jedoch vor der Verwendung angezeigt und kann so überprüft und noch angepasst werden, bevor die Klausel Eingang in die Datenbank nimmt.
- 130 Alle Klauseln können auch **später noch geändert** oder wieder gelöscht werden. Es genügt, das betreffende Dokument zu öffnen und manuell zu ändern. Es sollte jedoch darauf geachtet werden, dass die Struktur nicht geändert wird, da dies zu einer Störung im Abruf führen kann. Wer keinen **Editor** hat, kann einen solchen direkt in Red Ink aufrufen. Hierzu (1) eine etwaige bestehende Textselektion aufheben (es darf nichts angewählt sein), (2) "Add Clause" aufrufen und (3) bestäti-

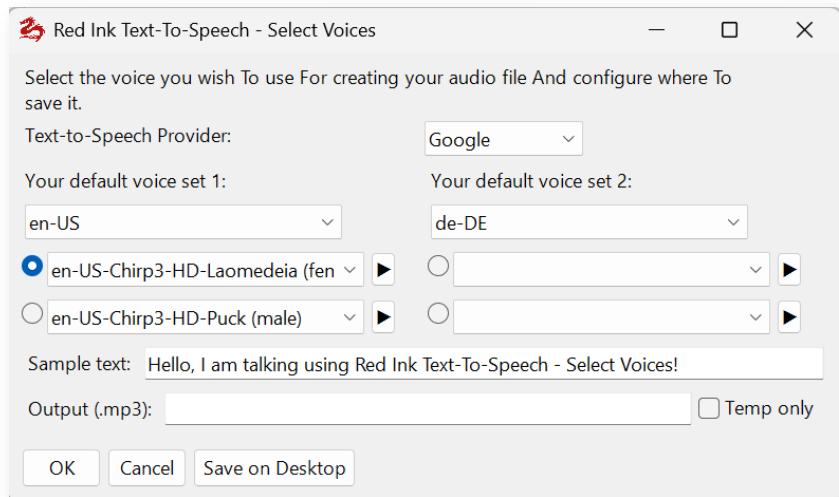
gen, dass die Klauseldatenbank-Dateien ("library files") manuell bearbeitet werden sollen. Es kann nun die gewünschte Klauseldatenbank-Datei (nicht die einzelnen Segmente) ausgewählt werden und sie wird in einem einfachen Editor geöffnet, wo Änderungen vorgenommen und gespeichert werden können. Sie werden sofort wirksam.

J. Podcasts und Audiobooks erstellen

- 131 Über das Analyze-Menü kann die Funktion zum Erstellen von Podcasts und von Audiobooks (d.h. Sprachaufnahmen von Texten) aufgerufen werden. Beide setzen allerdings voraus, dass Red Ink auf die Vertex-API von **Google** oder **OpenAI** (bei OpenAI selbst, nicht Azure) konfiguriert ist, da deren Text-to-Speech-Modelle benutzt werden. Es genügt jedoch, wenn nur das sekundäre Modell von Google oder OpenAI genutzt wird. Nötig ist das deshalb, weil für den Zugang zu den Google und OpenAI Text-to-Speech-Modellen dieselbe Authentifizierung benötigt wird wie für das Sprachmodell.
- 132 Die Funktion **Create Podcast** ähnelt derjenigen des beliebten Google-Services "NotebookLM". Der selektierte Text wird in einem ersten Schritt mithilfe der KI in einen Podcast-Dialog zwischen einem Moderator ("Host") und einem Guest ("Guest") umgewandelt. Dieser kann bearbeitet werden. Danach wird dieser Dialog in eine MP3-Datei mit zwei Sprechern umgewandelt.
- 133 Die Funktion **Create Audio** liest den selektierten Text Absatz für Absatz vor und erstellt daraus ebenfalls eine MP3-Datei. Verfügt der selektierte Text über einen Dialog von Host und Guest (dazu müssen die Absätze jeweils mit "H:" und "G:" beginnen und der erste ein "H:" aufweisen), dann wird der Text mit zwei Sprechern vorgelesen, als wäre es ein Podcast (auf diese Weise können Podcast-Scripte nachträglich vertont werden). Ist dies nicht der Fall, können Sie wählen, ob Red Ink zwischen zwei Sprechern wechselt soll (jedes Mal, wenn ein neues Kapitel beginnt, basierend auf dem, was als Titel erkannt wird) oder nur einem Sprecher. Es können auch JSON-codierte Texte verarbeitet werden, sofern sie dem für die Google-Text-to-Speech-API nötigen Format entspricht, wie z.B. der Multispeaker-Codierung (sofern freigeschaltet). JSON-Texte werden nur bis zu einer Länge von maximal 5'000 Zeichen unterstützt (das Long-Speech-API von Google wurde nicht implementiert). Andere Texte dürfen auch länger sein; um die Zeichenbegrenzung zu umgehen, werden Texte von Red Ink mit Create Audio und bei Podcasts absatzweise vertont. Nach Titeln in normalen Texten wird dabei jeweils eine kurze Pause gemacht. Zeilen mit reinen Zahlen oder Spezialzeichen werden ignoriert.
- 134 Alternativ kann mit Create Audio auch eine **Powerpoint-Datei mit Sprechernotizen vertont** werden. Hierzu wird die Funktion aufgerufen, ohne, dass vorher ein Text selektiert worden ist. Red Ink fragt in diesem Fall, ob eine Textdatei oder eine Präsentation vertont werden soll. Falls ja, geht ein Fenster auf und es kann die gewünschte Datei

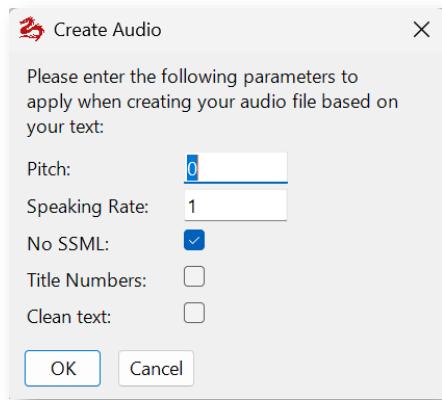
auf das Fenster gezogen (oder sie über den Dialog geöffnet werden). Ist es eine Textdatei, öffnet Red Ink ihren Inhalt in einem neuen Dokument, ist es eine Powerpoint-Datei, wird bei jeder Folie, welche Sprechernotizen enthält, diese vorgelesen und so in die Präsentation integrieren, dass sie später in Powerpoint automatisch abgespielt werden kann. Es ist dort dann auch möglich, daraus einen Film erzeugen zu lassen – quasi eine Videopräsentation mit künstlichem Sprecher (oder zwei Sprechern, falls pro Folie abwechselnde Stimmen gewählt werden).

- 135 In allen Fällen kann der Benutzer die für die Spracherzeugung gewünschten **Stimmen auswählen**. Es stehen dabei je nach dem gewählten Server-Standort (der über den Parameter "TTSEndpoint" gesteuert wird, siehe dazu Rz. 309 ff. unten) mehr oder weniger Sprachen und Stimmen zur Verfügung (bei Google werden sie vom Server abgerufen; bei OpenAI sind sie fest einprogrammiert und für alle Sprachen gleich). Zuerst muss jeweils die Sprache gewählt werden, es werden daraufhin die verfügbaren Stimmen angezeigt, von denen jeweils zwei ausgewählt werden können:



- 136 Dass hier zwei und nochmals zwei Stimmen gewählt werden können, ist reiner Komfort: Das Tool merkt sich alle vier zuletzt gewählten Stimmen, so dass sie beim nächsten Mal wieder erscheinen. Der Benutzer kann sich also seine "Lieblingsstimmen" für zwei Sprachen abspeichern und muss sie nicht jedes Mal von Neuem suchen. Wird ein Podcast vertont, muss er wählen, ob er den linken oder den rechten Stimmensatz möchte, beim Vorlesen eines Textes kann zwischen allen vier Stimmen ausgewählt werden.
- 137 Mit dem Knopf rechts von der Stimmenwahl kann die betreffende Stimme angehört werden; es wird der in "**Sample text**" eingetippte Text vorgelesen, inklusive etwaiger darin enthaltener SSML-Codierungen (so kann einfach getestet werden, ob sie von einer Stimme unterstützt werden – ist nichts zu hören, funktionieren sie nicht).

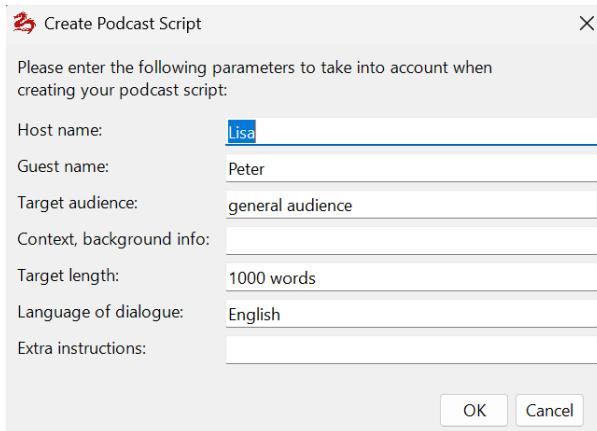
- 138 Im Feld "**Output**" kann angegeben werden, wohin die MP3-Datei geschrieben werden und wie sie heißen soll. Der Knopf "**Save on Desktop**" passt den Pfad so an, dass die Datei auf dem Desktop gespeichert wird. Wird "**Temporary**" gewählt, dann wird die generierte MP3-Datei abgespielt und sogleich wieder gelöscht (es entstehen beim Generieren von MP3-Dateien aber immer temporäre Dateien, die dann wieder gelöscht werden; benutzt wird das "%TEMP%" -Verzeichnis).
- 139 Die Stimmen klingen unterschiedlich natürlich; bei Google existiert die grösste und beste Auswahl für Englisch und auf dem US-Endpoint. Nicht alle Stimmen unterstützen SSML-Codierungen (damit kann Text mit Befehlen zur Betonung und Aussprache ergänzt werden) oder das Ändern von Stimmhöhe ("Pitch") und Sprech-Geschwindigkeit ("Speaking Rate"); bei OpenAI werden sie derzeit gar nicht unterstützt. Sollte das Erzeugen von Sprache nicht funktionieren, kann es also daran liegen, dass einer dieser Parameter verändert oder SSML benutzt wurde, obwohl die Stimme dies nicht unterstützt. Diese Parameter werden jeweils vor der Generierung abgefragt:



- 140 Der Standardwert für "**Pitch**" ist 0; ein Wert von z.B. -0.5 reduziert die Stimmhöhe etwas, was die Stimme üblicherweise etwas gehaltvoller klingen lässt. Die "**Speaking Rate**" ist normalerweise bei 1; ein Wert von 1.1 lässt den Sprecher etwas schneller reden, 0.9 hingegen verlangsamt ihn etwas. Dies gilt wie erwähnt nur für Google und bei Stimmen, die dies unterstützen. Ist "**No SSML**" angeklickt, wird sichergestellt, dass keine SSML-Kommandos an die Sprachgenerierung gelangen und damit Fehler vermieden werden, weil nicht alle Stimmen diese SSML-Codes unterstützen (die generierten Podcast-Dialoge enthalten standardmäßig SSML-Codierungen).
- 141 Der Parameter "**Title Numbers**" erscheint nur bei Create Audio und sorgt dafür, dass bei Titeln etwaige Nummerierungen mitgelesen werden; in den meisten Fällen werden sie allerdings eher störend sein (nötigenfalls kann der Text vorgängig mit gesprochenen Nummern versehen werden, falls z.B. Kapitelnummern gelesen werden sollen).
- 142 Wenn Sie möchten, dass jeder Absatz einer Bereinigung durch die KI unterzogen wird, bevor er gelesen wird, z.B. um Kapitelverweise und

andere nicht leicht lesbare Inhalte zu entfernen, können Sie "**Clean text**" auswählen und erhalten die Möglichkeit, einen Prompt für die Be-reinigung zu definieren. Die Clean text-Funktion kann insbesondere auch benutzt werden, wenn der vorzulesende Text zu komplizierte Sätze enthält (was zu einer Fehlermeldung führen kann). Die KI wird dann mit dem Prompt gebeten, aus einem komplizierten Satz z.B. zwei Sätze zu machen, ohne den Inhalt zu verändern. Wird die Clean text Funktion verwendet (und nicht die Option einer nur temporären MP3-Datei gewählt wird), erstellt Red Ink am selben Ort und mit dem selben Namen wie die MP3-Datei eine Text-Datei die den bereinigten Text wie vorgelesen enthält. Mit einem Compare-Programm kann so nach-verfolgt werden, welche Änderungen es gab.

- 143 Wird ein Podcast-Script generiert, können ebenfalls verschiedene Parameter erfasst werden:



- 144 Der **Name von Host und Guest** wird benötigt, weil sich die beiden Rollen im generierten Dialog ansprechen werden. Das Zielpublikum ("**Target audience**") wie auch der Hintergrund bzw. Anlass des Podcasts ("**Context, background info**") wird der KI für die Generierung des Podcasts ebenfalls mitgegeben. Die "**Target length**" enthält Angaben zur gewünschten Länge, wobei die Länge des Podcasts auch davon abhängt, wie viel das Quellmaterial hergibt; nach unserer Erfahrung haben gewisse Sprachmodelle Mühe, sich an Vorgaben zur Länge zu halten, wenn diese in Zeiteinheiten angegeben werden. Es ist besser, die Anzahl Wörter zu nennen. Die "**Language**" gibt die Ausgabe-sprache an und im Feld "**Extra instructions**" können noch weitere Be-fehle eingefügt werden, die mit dem Prompt zur Generierung des Dia-logs mitgegeben werden (nebst "PreCorrection").

- 145 Der generierte Podcast wird in einem separaten Fenster angezeigt und kann dort bearbeitet werden. Wird dies nicht mit "Cancel" abgebro-chten, wird Red Ink versuchen, das Script zu vertonen. Es kann aber auch über die Zwischenablage in ein normales Word-Dokument kopiert und später über "Create Audio" vertont werden; Create Audio erkennt Podcast-Scripte am wechselnden "H:" (für Host) und "G:" (für Guest) automatisch.

146 Alle vertonten MP3-Dateien werden nach der Vertonung mit einem internen MP3-Player abgespielt. Ein Abbruch ist bei Create Audio mit dem "Cancel"-Knopf möglich. Die Vertonung eines Podcasts läuft (unsichtbar) im Hintergrund ab, d.h. es kann weitergearbeitet werden; bei Create Audio wird gezeigt, bei welchem Absatz sich die Vertonung befindet (kommt es zu einem Fehler, wird der Absatzbeginn angezeigt). Red Ink meldet, sobald der Podcast zum Abspielen bereit ist. Kommt es bei der Vertonung zu Fehlern, dann fehlen die entsprechenden Sequenzen im Endergebnis (oder die Datei ist gar nicht abspielbar). Ist z.B. eine Stimme mit den gewählten Einstellungen nicht kompatibel, wird nur die andere Stimme zu hören sein. In solchen Fällen ist die Vertonung mit einer neuen Auswahl an Stimmen zu wiederholen und bei den Parametern auf die Standardwerte zu achten und kein SSML zu verwenden bzw. es herausfiltern zu lassen.

147 Läuft die Audio-Generierung (Create Audio und Podcast), versucht Red Ink den Computer so einzustellen, dass er nicht mehr in den **Schlafmodus** fallen kann. Diese Einstellung wird nach Ende der Audio-Generierung wieder rückgängig gemacht.

148 In Bezug auf den **Schutz der Daten** ist zu beachten, dass bei Google die Audioinhalte auf einem anderen System generiert werden als die Antworten des Sprachmodells. Während letzteres womöglich im eigenen Land läuft, kann es je nach gewähltem "Endpoint" sein, dass die Sprachinhalte auf einem ausländischen Server erzeugt werden.

K. Integrierte Anonymisierungs-Funktion

149 Red Ink verfügt über eine einfache, eingebaute Anonymisierungsfunktion, mit welcher alle Texte, die an ein Modell oder einen Special Service gesendet werden, vorab anonymisiert und die zurückgegebenen Texte wieder re-identifizieren kann. Mit dieser Funktion können vertrauliche Inhalte auch solchen Dienstleistern eingesetzt werden, die nicht hinreichend vertrauenswürdig sind oder nicht die nötigen vertraglichen Zusicherungen geben. Die Anonymisierung läuft vollständig lokal ab und basiert derzeit auf dem Prinzip, das vordefinierte Suchbegriffe in einer Liste durch Platzhalter ersetzt werden. Wir experimentieren mit Anonymisierungsmodellen, die auch auf einem lokalen Computer mit mässiger Rechenleistung laufen, haben bisher aber noch kein geeignetes Modell gefunden, das sich vernünftig implementieren lässt und die nötige Qualität liefert.

150 Es müssen und können zwei Dinge konfiguriert werden:

- Bei welchem Modell bzw. Special Service die Anonymisierung wie ablaufen soll: Dies kann einerseits für jedes Modell in der Konfiguration des Modells hinterlegt werden, und andererseits vom Benutzer mit der Datei "redink-anon.txt" auf seinem Desktop selbst bestimmt werden.

- Welche Suchbegriffe durch Platzhalter ersetzt und danach wieder eingefügt werden sollen. Dies wird vom Benutzer mit der Datei "redink-anon.txt" auf seinem Desktop bestimmt.
- 151 Red Ink nutzt eine einfache, aber leistungsfähige Konfiguration über die Datei **redink-anon.txt** auf dem Desktop des Benutzers, um sowohl das "WANN" als auch das "WIE" der Anonymisierung festzulegen. Standardmäßig ist **kein** Anonymisierungsschritt aktiv ("none; 0"), bis diese Datei entweder vom Benutzer angelegt und angepasst wird oder sie für das entsprechende Modell oder den Special Service in deren bzw. dessen Konfiguration vorgesehen ist (Parameter "Anon" und "Anon_2" in "redink.ini").
- 152 Die **Struktur der Datei** gliedert sich in Abschnitte, die mit eckigen Klammern markiert werden:
- Was nach **[All]** kommt, gilt für alle Modelle und Services;
 - Was nach **[ModelA, ModelB, etc.]** überschreibt für genau diese Modelle. Es kann also ein Abschnitt für ein Modell oder mehrere Modelle definiert werden. Sie sind durch Kommazeichen getrennt. Der Modellname entspricht demjenigen aus der Konfiguration des Modells bzw. Special Service;
 - Zeile, die mit ";" beginnen werden ignoriert. Das kann für Kommentare genutzt werden.
- 153 Innerhalb jedes Abschnitts wird zunächst auf einer Zeile der Anonymisierungsmodus und -typ definiert:
- ```
Anon = <mode>; <type>
```
- wobei:
- <mode> ein Parameter wie *none*, *silent*, *ask*, *askshow*, *show* ist, und
  - <type> eine Zahl von 0 bis 4 (siehe unten).
- Nach derselben Struktur erfolgt übrigens auch die Bestimmung des Parameters "Anon" und "Anon\_2" in der Konfiguration der Modelle und Special Service.
- 154 Anschliessend werden die **Suchbegriffe und -muster** aufgelistet, nach denen Platzhalter eingefügt werden sollen, und zwar entweder als einfache Begriffe oder Suchbegriffe mit Wildcards, oder als Regex-Zeilen. Einige Beispiele:
- Regex:\b[A-Z]{2}\d{4}\b
  - "Max Mustermann" (genauer Text)
  - Kunde\* (Wildcard \* → beliebige Buchstaben/Ziffern)
  - ACME\*{{Firma}} (statt des Platzhalters "redacted" wird der Platzhalter "Firma" verwendet)
- 155 Die **Platzhalter** folgen immer dem Format

<<prefix>\_<GruppenID vierstellig>\_<SubIndex>>

also "<redacted\_0003\_1>" für die erste Entität der dritten Regel. Die Gruppen-IDs werden intern fortlaufend vergeben, Sub-Indizes zählen pro Regel neu gefundene, unterschiedliche Vorkommen eines Begriffs. Gleichzeitig legt das Modul in einer Liste fest, welche Originaltexte welchem Platzhalter entsprechen.

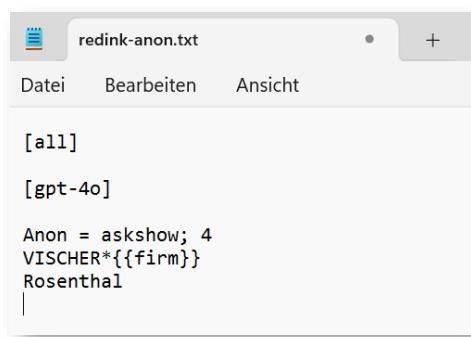
- 156 Für den Benutzt sind die Platzhalter nicht unbedingt relevant, weil sich das System automatisch merkt, welchen Begriff durch welchen Platzhalter ersetzt hat und am Ende wieder diesen Begriff einsetzt. Die Verwendung von selbstdefinierten Platzhaltern können aber dem Sprachmodell helfen, den Kontext auch bei anonymisierten Texten zu verstehen und so eine besser Antwort zu liefern. Wenn im obigen Beispiel alle Vorkommen von ACME wie "ACME" durch einen Platzhalter mit dem Wort "Firma" statt "redacted" ersetzt werden, weiss das Modell, dass es sich um einen Firmennamen handelt und kann dies berücksichtigen, auch wenn es diesen nicht kennt. Weil alle Vorkommen von ACME dieselbe GruppenID haben, bleibt auch dieser Zusammenhang erhalten. Wo diese Dinge alle nicht erforderlich sind, kann auch auf die Definition eigener Platzhalter verzichtet werden.
- 157 Die unterstützten **Modi** sind:
- **none**: keine Anonymisierung (das ist der Standard);
  - **silent**: automatisch, ohne Nachfrage (der Benutzer merkt nichts davon und erhält den anonymisierten Text nicht zur vorgängigen Prüfung und Anpassung; wer wissen will, wie die Anonymisierungsfunktion seinen Text bearbeitet, kann im Analyse-Menü die Funktion Anonymization benutzen);
  - **ask**: fragt per Ja/Nein-Dialog, ob anonymisiert werden soll und tut dies dann stumm (wie silent);
  - **askshow**: fragt, ob anonymisiert werden soll und falls ja, wird der anonymisierte Text vor der Verwendung noch angezeigt, so dass dieser noch angepasst (d.h. manuell anonymisiert) oder abgebrochen werden kann. Wird manuell anonymisiert, muss beim Ergebnis auch manuell re-identifiziert, d.h. der Platzhalter muss von Hand wieder durch den gewünschten Wert ersetzt werden;
  - **show**: anonymisiert immer und zeigt den anonymisierten Text vor der Verwendung zur Prüfung an.
- 158 Die **Typen** im Überblick sind:
- **0**: keine Anonymisierung (Standardwert);
  - **1**: Der Benutzer kann seine Suchbegriffe (z.B. ein Name), getrennt durch Kommas, in einem Eingabefenster eingeben; es werden die zuletzt verwendeten Suchbegriffe angezeigt (Regex funktioniert hier nicht);
  - **2**: Wie 1, aber das Eingabefeld ist jeweils leer;

- **3:** Der Benutzer wird nicht nach Suchbegriffen gefragt, sondern es wird gleich auf die Datei redink-anon.txt auf dem Desktop des Benutzers zurückgegriffen und ein Fehler angezeigt, wenn diese fehlt. Dies ist die einzige Variante, in welcher Regex funktioniert;
- **4:** Der Benutzer kann seine Suchbegriffe wie bei 1 in einem Eingabefenster erfassen; es werden die für das Modell vorgesehenen Suchbegriffe angezeigt, die übernommen werden können (Regex funktioniert hier nicht).

159 Die Anonymisierung wird in Word, Excel und Outlook basierend auf der Konfiguration automatisch vor jedem Zugriff auf ein Sprachmodell bzw. Special Service aufgerufen, bezieht sich aber **nur auf den im eigenen Dokument selektierten Text**. Die Prompts an das Modell, einschliesslich allem, was über das Freestyle-Kommando an das Modell gesendet wird (wie zum Beispiel ein mittels "{doc}" eingefügter Dokumenteninhalt oder der Inhalt einer Datei mit "(file)" oder der Zwischenablage "(clip)") wird **nicht anonymisiert**.

160 Die Anonymisierung verwendet standardmässig die beim Modell bzw. Special Service hinterlegte Konfiguration von Modus und Typ. Hat der Benutzer jedoch die Datei **redink-anon.txt** auf seinem Desktop, überschreiben die dortigen Werte die Standardkonfiguration. Enthält die lokale redink-anon.txt Datei Angaben sowie unter "[all]" als auch für ein spezifisches Modell, haben die Angaben des spezifischen Modells Vorrang. So kann jeder Benutzer die für ihn oder sie selbst am besten passende Einstellung rasch und einfach vornehmen; es braucht nur die Datei redink-anon.txt angepasst zu werden mit einem einfachen Editor.

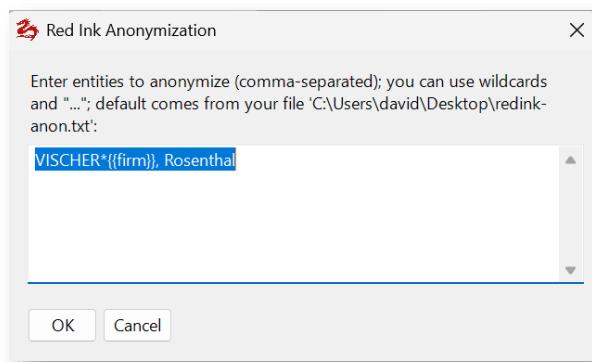
161 Beispiel (anonymisiert wird nur bei der Verwendung von gpt-4o, der Benutzer wird gefragt, ob anonymisiert werden soll, er kann die Suchbegriffe basierend auf dem Inhalt dieser Datei eingeben und das Ergebnis wird vorgängig angezeigt; anonymisiert werden alle Vorkommen von "VISCHER" und des Suchbegriffs "Rosenthal", wobei bei "VISCHER" als Platzhalter "firm" verwendet wird, bei Rosenthal hingegen der Standard-Platzhalter redacted.):



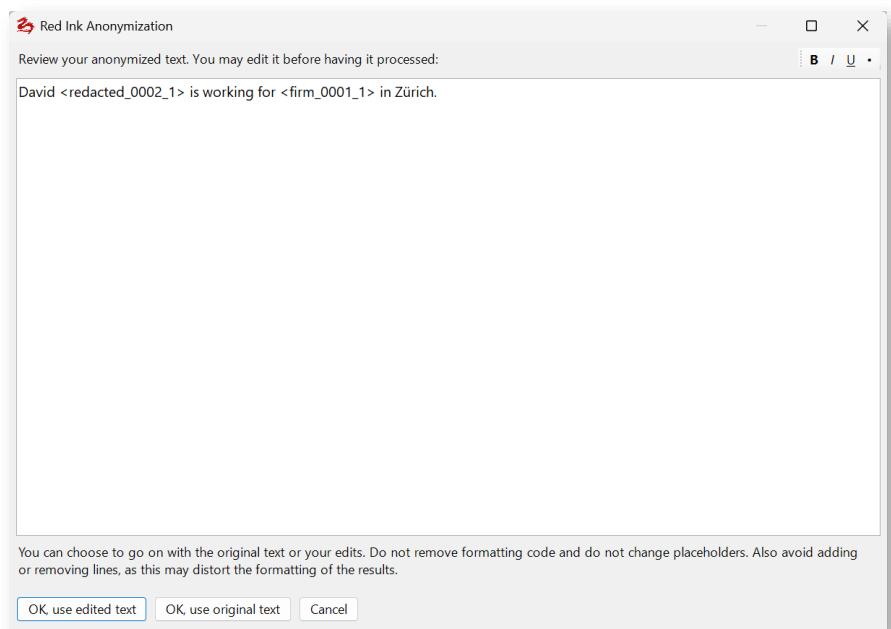
162 Ist die Anonymisierung durchgeführt (es wird der gesamte Text abgearbeitet), und falls die Modi **show** oder **askshow** gewählt wurden, öffnet Red Ink dann ein Bearbeitungsfenster, in dem das anonymisierte

Ergebnis manuell geprüft und angepasst werden können (Platzhalter und Formatierung sollten Sie nicht verändert werden). Sobald das Fenster mit dem entsprechenden Freigabe-Knopf geschlossen wird, fährt das Add-in mit dem finalen, anonymisierten Text wie gewohnt weiter. Wird abgebrochen, wird nichts weitergegeben und es wird abgebrochen bzw. das Add-in verhält sich so, als wäre vom Sprachmodell nichts zurückgeliefert worden.

- 163 Wurde in Word der Text "*David Rosenthal is working for VISCHER in Zürich*" und eine redink-anon.txt-Datei mit obigem Inhalt verwendet, erscheint nach der Ja/Nein-Abfrage, ob anonymisiert werden soll, folgendes Prompt-Fenster:



- 164 Das Ergebnis sieht dann so aus:

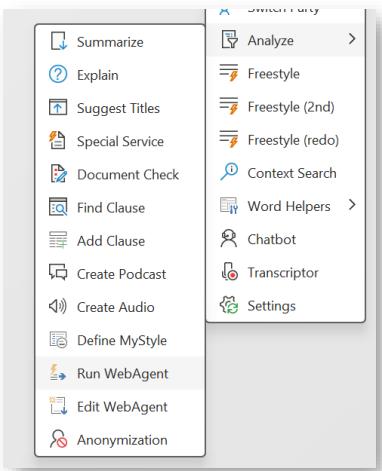


- 165 Hat das Sprachmodell oder der Special Service einen Text zurückgeliefert, ersetzt Red Ink automatisch die Platzhalter mit den vorherigen Werten. Es muss manuell geprüft werden, ob die ersetzen Begriffe am richtigen Ort sind, denn das Sprachmodell hat diese ja nicht im Klar-

text gesehen. Hier können wie erwähnt beschreibende Platzhalter die Antwortqualität erhöhen.

- 166 Die Anonymisierung kann auch ohne Sprachmodell getestet werden, indem im Word-Add-in die Funktion "**Anonymization**" im Untermenü von "Analyze" verwendet wird. Dort kann zwischen Anonymisierungs-Typ 3 und 4 gewählt werden. Dort wird dann am Ende des Textes auch die Tabelle mit den Zuordnungen der Platzhalter angezeigt
- 167 Wer selbst prüfen will, was wirklich an das Sprachmodell gesendet wird, kann den Code von Red Ink in seiner eigenen Entwicklungsumgebung im Debugging-Modus laufen lassen. Ist der "Debug"-Parameter auf True gesetzt, wird im Output der an den Endpoint gesendete String und der von diesem empfangene String ausgegeben.

## L. WebAgent

- 168 Das Add-in in Word ist in der Lage, basierend auf vom Benutzer definierten Skripten Aufgaben im Internet auszuführen, bei welchen Seiten im Internet abgerufen und deren Inhalt verarbeitet werden kann, insbesondere mit Hilfe der Sprachmodelle. Diese Funktion kann z.B. benutzt werden, um die von einer Behörde von Zeit zu Zeit auf ihrer Website publizierten Entscheide abzurufen, zusammenzufassen und das Ergebnis in einem Bericht darzustellen. Der Benutzer spart sich Zeit, weil er nicht alle Seite von Hand abrufen und alles im Detail lesen muss. Das Abrufen übernimmt der Web Agent und die Zusammenfassung die KI.
- 
- 169 Aufgerufen wird die Funktion über das Untermenü von "Analyze" und dort als Funktion "Run WebAgent". Wird sie aufgerufen, werden dem Benutzer die zur Verfügung stehenden Skripte angezeigt. Es kann ein Pfad für lokale Skripte definiert werden und einer für zentral verwaltete ("WebAgentPath", "WebAgentPathLocal", vgl. Rz. 309 ff.). Die Dateien heissen "redink-ag-xxx.json", wobei xxx für einen wählbaren Namen steht, der für Dateinamen zulässig ist. Die Endung ".json" gibt an, dass es sich um eine Datei im Format JSON handelt, einem Standardformat für solche Zwecke. Es gibt viele Editor, mit denen diese Dateien bearbeitet werden können. Die Bearbeitung ist aber auch in Red Ink selbst möglich.
- 170 Wir ein Skript ausgewählt, wird es zunächst darauf geprüft, ob es Benutzerparameter enthält, d.h. Parameter, die der Benutzer vor jedem Laufenlassen eingeben muss, damit das Skript auf aktuelle Gegebenheiten ausgerichtet werden kann, z.B. mit einem Datum arbeiten kann oder einen Suchbegriff erfragen kann. Muss ein Passwort eingegeben werden, weil das Skript die Eingabe eines solchen für eine Website

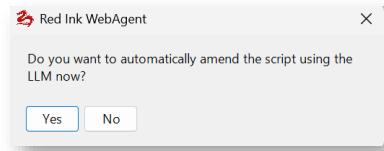
vorsieht, wird dieses ebenfalls abgefragt. Sieht das Skript das Versenden eines E-Mails mit dem erstellten Bericht vor, wird der Benutzer sicherheitshalber ebenfalls gefragt, ob er damit einverstanden ist. Es erfolgt auch eine kurze Prüfung, ob das Skript syntaktisch den Grundanforderungen an eine JSON-Datei genügt (ist das nicht der Fall, sollte das Skript von einem LLM oder manuell in einem JSON-Editor geprüft werden). Danach läuft das Skript durch, bis es ein Ergebnis anzeigt (oder Fehlermeldungen). Läuft das Skript ab, geht immer wieder ein Fenster auf, in welchem angezeigt wird, was der WebAgent gerade tut. Mit der Taste "Esc" kann abgebrochen werden. Als Sprachmodell wird für das Skript jenes Modell benutzt, welches in der Datei für alternativen Modelle, sofern konfiguriert, in dessen Segment den Parameter "WebAgent = True" hat. Damit kann z.B. ein kleineres, effizientes Modell für das Skript gewählt werden. Es braucht nur im betreffenden Abschnitt der Konfigurationsdatei der alternativen Modelle der entsprechende Eintrag vorgenommen zu werden (sonst wird das Hauptmodell verwendet).

 [step:summarize\_decision]  
[url:https://search.bger.ch/ext/eurospider/live/de/php/aza/http/index.php?highlight\_docid=aza://26-08-2025lang=dezoom?type=show\_document][id:"1C\_440/2025","date":"26.08.2025","url":"https://search.bger.ch/ext/eurospider/live/de/php/aza/http/index.php?highlight\_docid=aza://26-08-2025-1C\_440-2025lang=dezoom?type=show\_document","summa...]

- 171 Am Ende wird, wenn das Skript entsprechend programmiert worden ist, ein Bericht angezeigt (es kann das Ergebnis aber auch in einer Markdown-Datei abgelegt werden oder beides). Hier als Beispiel, was das Musterskript produziert (es analysiert die neusten Entscheide des Schweizerischen Bundesgerichts, die auf seiner Website publiziert wurden und fasst sie kurz zusammen):

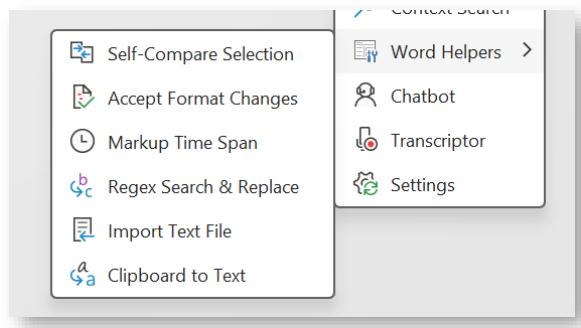


- 172 Es ist nicht ganz trivial, ein gutes, funktionierendes Skript zusammenzustellen. Um die Arbeit etwas zu erleichtern, steht zunächst die Funktion "**Edit WebAgent**" zur Verfügung. Mit ihr kann eines der konfigurierten Sprachmodelle benutzt werden, um ein bestehendes Skript mit Hilfe von KI zu bearbeiten oder auch neu erstellen zu lassen. Das besondere daran ist, dass Red Ink selbst über eine Dokumentation verfügt, wie die Skripte zu programmieren sind und daher in natürlicher Sprache gebeten werden kann, ein solches Skript zu erstellen ("Gehe zuerst auf die Seite XYZ und rufe sie ab. Analysiere daraufhin mit dem LLM, von ob es darauf einen Link hat mit dem Datum ABC und rufe diesen Link ab. ...", wobei ABC z.B. ein benutzerdefinierter Parameter sein kann, der vor jedem Laufenlassen abgefragt wird). Wird die Funktion aufgerufen, muss zuerst entschieden werden, ob ein Skript neu erstellt wird (mit Hilfe des LLM) oder ob ein bestehendes Skript verändert werden soll. Wird letzteres gewählt, kann das Skript ausgewählt werden und es wird in einem Editor geöffnet, mit welchem es manuell bearbeitet (und auch abgespeichert) werden kann. Schliesst er den Editor, wird der Benutzer gefragt, ob er das Skript automatisch mit Hilfe des LLM anpassen will. Wird das bejaht, kann ein Prompt eingegeben und danach das Sprachmodell ausgewählt werden, welches diesen Prompt ausführen soll (unter Zuhilfename der Red Ink gespeicherten Anleitung). Es sollte ein hinreichend leistungsfähiges Modell für die Erstellung verwendet werden. Dasselbe Vorgehen gilt auch für die Neuerstellung des Skripts. Ist das Skript von der KI erstellt, wird es kurz auf Formalien (JSON-Struktur) überprüft und allenfalls nochmals auf Wunsch von der KI korrigiert. Vor der Überarbeitung durch die KI empfiehlt es sich, eine Sicherheitskopie des bisherigen Skripts zu erstellen.
- 173 Mit den Angaben in Anhang 3 können Skripte auch von Hand erstellt und vor allem bearbeitet werden. Appendix 3 enthält auch ein Beispiel. Es ist normal, dass ein Skript nicht von Anfang an problemlos durchläuft. Es werden typischerweise mehrere Durchgänge nötig sein. Das Auslesen und Analysieren von HTML-Seiten, ob mittels Sprachmodell oder deterministisch über die HTML-Struktur, braucht etwas Experimentieren. Hierzu steht auch eine Debug-Option zur Verfügung, die über das Skript aktiviert werden kann und ein Textfile mit Debugging-Angaben erzeugt, welches auf dem Desktop gespeichert wird.
- 174 Eine weitere Möglichkeit zur Erstellung eines Skripts ist es natürlich, den Quellcode von Red Ink anzuschauen und ggf. mit Hilfe der KI ein Skript zu erzeugen und zu überarbeiten. Die interne Funktion geht nicht ganz soweit: Für sie wurde aus dem Quellcode eine Spezifikation erzeugt, die etwas schlanker ist als der Quellcode.



## M. Word-Helpers – praktische Alltags-Helper (fast) ohne KI

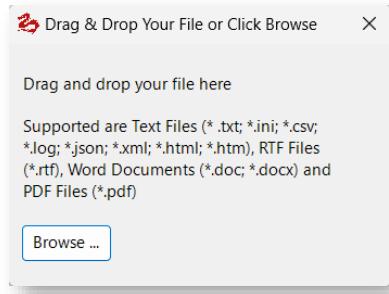
175 Die Word-Helpers sind Zusatzfunktionen, die mit einer Ausnahme eigentlich nichts mit KI zu tun haben, aber in Word fehlen und im Alltag sehr praktisch sein können. Darum haben wir sie gleich auch eingebaut. Sie können über ein eigenes Submenü in der Menükachel und über das Kontextmenü abgerufen werden:



- **Self-Compare Selection:** Die erste Hälfte des selektierten Textes wird mit der zweiten Hälfte des selektierten Textes im selben Dokument verglichen. Sind z.B. zwei oder vier Absätze selektiert, wird ein Markup des zweiten mit dem ersten bzw. es werden die ersten beiden Absätze mit den zweiten zwei Absätzen verglichen. Oft besteht jedenfalls in unserer Arbeit der Bedarf kurz zu prüfen, was sich zwischen zwei Klauseln verändert hat. Bisher mussten dafür zwei separate Dokument erstellt und verglichen werden. Das fällt mit dieser Funktion weg. Es kann über Settings gesondert definiert werden, welche Markup-Technik (Word, Diff oder DiffW) zum Einsatz kommt (siehe Rz. 25 oben). Da hier meist kurze Texte verglichen werden, ist Diff oder DiffW in der Praxis oft sinnvoll, auch wenn die Funktion weniger zuverlässig ist.
- **Accept Format Changes:** Im selektierten Abschnitt werden alle Formatierungsänderungen akzeptiert, aber nur diese. Sie sind in der Darstellung am Bildschirm oder in Ausdrucken oftmals störend. Sie könnten zwar ausgeblendet werden, aber das erfordert immer wieder einen zusätzlichen Handgriff. So kann das Problem mit einem Griff beseitigt werden.
- **Markup Time Span:** Mit dieser Funktion kann berechnet werden, wieviel Zeit seit dem ersten und letzten Markup oder Kommentar im selektierten Bereich vergangen ist, basierend auf den Zeiteinträgen, die Word bei Markups erstellt. Wird die Funktion angewählt, kann angegeben werden, ob die Berechnung nur auf den Kommentaren eines bestimmten Autors berücksichtigt werden soll (ansonsten werden alle Kommentare und Markups beachtet). Diese

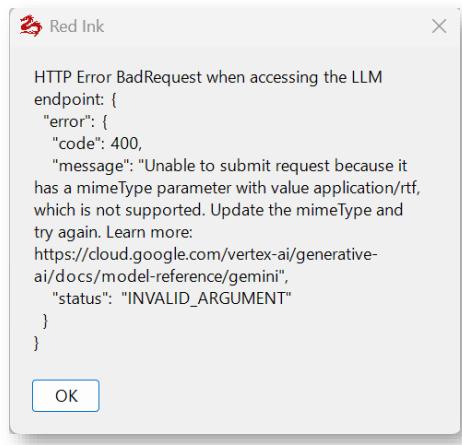
Funktion kann nützlich sein, wenn nachträglich berechnet werden soll, wie lange eine Überarbeitung ungefähr gedauert hat, etwa für den Eintrag in einem Timesheet.

- **Regex Search & Replace:** Mit dieser Funktion kann eine sog. Regex-Suche. Regex (kurz für "Regular Expressions") ist ein leistungsstarkes Werkzeug zur Textsuche und -manipulation, das Muster wie Wortgruppen, Zahlen oder spezielle Formate in Texten erkennt, anstatt nur exakte Übereinstimmungen wie bei einer normalen Suche zu finden. Es steht über die Word-Oberfläche nicht zur Verfügung. Mit dieser Funktion können ein oder mehrere Regex-Suchmuster eingegeben und das Add-in beauftragt werden, Treffer mit einem Text oder mehreren Texten zu ersetzen. Das Add-in fragt zuerst nach dem Suchmuster, dann Suchoptionen und dann Ersatztexte. Sollen mehrere Suchmuster eingegeben werden, ist jedes auf einer neuen Zeile zu erfassen (ohne Leerzeilen) und der passende Ersatztext ebenfalls auf jeder entsprechenden Zeile. Die Suchoptionen gelten für alle Suchmuster. Ist ein Regex-Suchmuster non-konform, wird eine Fehlermeldung ausgegeben. Gibt es keine Ersatztexte, wird der erste Treffer angezeigt. Die Suche und das Ersetzen findet nur im selektierten Text statt. Wer mehr Informationen über die möglichen Suchmuster und Optionen haben möchte, findet diese auf einer Hilfeseite von Microsoft, die via <http://vischerlnk.com/regexinfo> abrufbar ist.
- **Import Text File:** Es ist dies dieselbe Funktion, die auch in Freestyle zur Verfügung steht und es erlaubt, den Inhalt eines Textdokuments direkt in das aktuelle Dokument als Text einzufügen. Das geht in Word nicht ohne Weiteres, insbesondere nicht bei PDF-Dokumenten (es wird vom Helper allerdings nur der Text von PDFs verarbeitet, der als solcher darin vorhanden ist, d.h. nach welchem gesucht werden kann; wird zusätzlich eine Texterkennung benötigt, weil der Text z.B. bei gescannten PDFs nur als Bild vorliegt, ist das vorher mit einem separaten PDF-Programm durchzuführen oder – wo das Modell dies unterstützt – mit dem nächsten Word-Helper oder mit Freestyle zu machen, siehe sogleich). Wird der Befehl angewählt, öffnet sich das folgende Fenster: Die Datei kann einfach eingelesen werden, indem sie mit der Maus auf das Fenster gezogen wird. Alternativ kann sie über den Button ausgewählt werden:



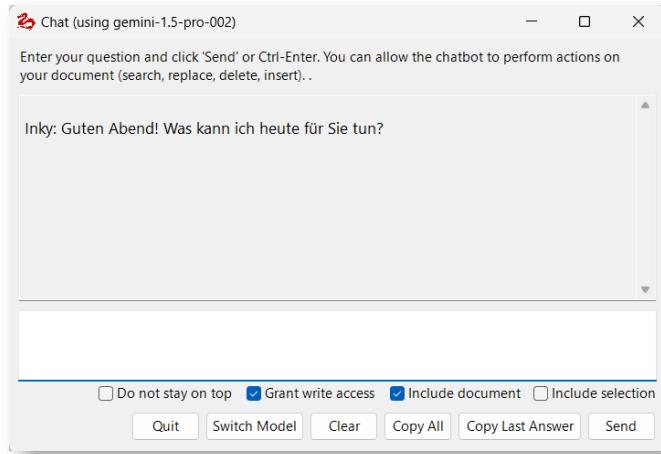
Alternativ zu dieser Funktion kann bei der Verwendung von Sprachmodellen, die nicht nur Texteingaben, sondern auch Dateien verarbeitet werden können, die nachfolgende "Clipboard to Text" oder Freestyle-Funktion mit dem Trigger "(file)" oder "(clip)" benutzt werden. Dann übersetzt das Sprachmodell den Inhalt der Datei, was bei einem PDF z.B. auch dann funktionieren kann, wenn eine Texterkennung erforderlich ist).

- **Clipboard to Text:** Diese Funktion ist im Gegensatz zu den anderen Word-Helper-Funktionen KI-basiert und funktioniert nur, falls das primäre Modell die Verarbeitung von binären Objekten unterstützt ("APICall\_Object"-Parameter). Wird sie angewählt, wird die KI gebeten, den Inhalt der Zwischenablage in Text zu konvertieren, ohne, dass etwas eingegeben werden muss, und unabhängig davon, ob es sich bei den Daten in der Zwischenablage um ein Bild, ein Textdokument oder eine Audio- oder Videodatei handelt. Diese Funktion ist sehr praktisch, wenn z.B. von einem neben Word offenen Dokument ein Teil in Word hineinkopiert werden soll, dies aber mit Copy & Paste nicht funktioniert. Es genügt dann, mit Shift-Windows-S einen Screenshot des Bereichs zu machen und diese Funktion zu wählen – und der Text (oder die Bildbeschreibung) wird eingefügt. Wird das Format des Inhalts in der Zwischenablage nicht unterstützt, gibt das Modell eine Fehlermeldung wie diese zurück:



#### N. Integrierter Chatbot "Inky"

- 176 Das Word-Add-in von Red Ink bietet einen integrierten KI-Chatbot an. Er wird aufgerufen und erscheint in einem separaten Fenster, das auch offen bleiben kann, während weitergearbeitet wird; es lässt sich einstellen, ob es immer sichtbar bleiben soll oder nicht ("Do not stay on top").

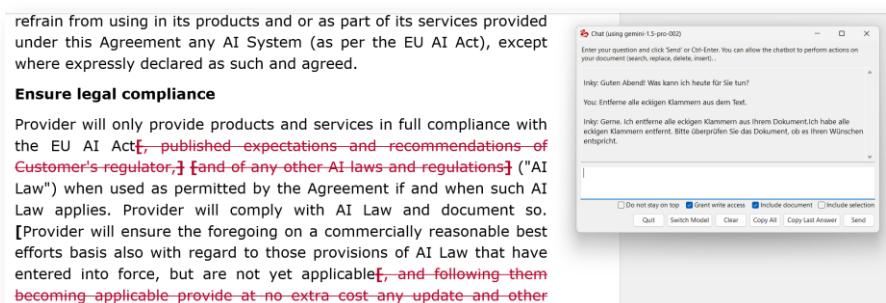


- 177 Der Chatbot (der sich als "Inky" kennt) ist einerseits ein normaler KI-Chatbot, der mit dem Benutzer einen Dialog führen kann und sich dabei auch den Dialog merkt (im Rahmen der konfigurierbaren Begrenzung), d.h. frühere Fragen und Antworten einbeziehen kann. Er kommuniziert mit dem Benutzer initial in der Sprache, in der Word konfiguriert ist, stellt sich aber um, wenn der Benutzer in einer anderen Sprache spricht. Der Chatbot hat derzeit allerdings noch keinen eigenen Internet-Zugang, kann aber beim Erstellen von Texten für kurze Fragen, zum Beispiel zu Formulierungen oder zur Erklärung von Begriffen, spontan darauf zurückgreifen. Der Benutzer gibt seine Frage in das weisse Eingabefeld ein; abgeschlossen wird mit Ctrl-Enter oder durch den "Send"-Knopf. Die jeweils jüngste Antwort des LLM kann dann in die Zwischenablage kopiert und so im Text verwendet werden ("Copy Last Answer"). Es können auch alle Antworten kopiert oder ein Dialog gelöscht werden, wenn das Thema gewechselt werden soll. Ein Hin-

und Herschalten zwischen dem primären und allenfalls konfigurierten zweiten Modell ist ebenfalls möglich (der Kontext bleibt erhalten). Mit "Quit" kann das Fenster geschlossen werden, aber der bisherige Dialog bleibt erhalten bis zum nächsten Mal oder bis "Clear" benutzt wird.

178 Soll der Chatbot sich etwas für später aus einem Text merken, muss ihm das mitgeteilt werden. Er sieht zwar den jeweils aktuellen Text (d.h. das aktive Dokument), aber er sieht ihn nur, solange eine der beiden Checkboxen aktiviert ist, d.h. er wird ihm nicht im Chatverlauf mitgegeben. Damit der Chatbot sich sicher erinnern kann, wird er daher die Informationen wiederholen, wenn er dazu aufgefordert wird sich etwas zu merken. Um das Wechseln zwischen Dokumenten zu erleichtern, sieht der Chatbot übrigens auch jeweils den Namen des Dokuments. Auch hier gilt aber dasselbe: Wenn er sich diesen merken soll, muss ihm das gesagt werden ("Merke Dir die Stelle, in welches es um die Preisanpassung geht und den Namen des Dokuments"). Das kann z.B. praktisch sein, wenn zwei Dokumente parallel miteinander verglichen werden müssen (alternativ kann hierfür die Freestyle-Funktion mit dem Zusatz zum Einlesen eines Zweitdokuments verwendet werden).

179 Was diesen Chatbot von anderen Chatbots unterscheidet ist allerdings, dass er den aktuellen Text (oder eine Auswahl davon), den der Benutzer bearbeitet, nicht nur sehen, sondern auch verändern kann, wenn ihm dies erlaubt wird ("Grant write access") und der Benutzer das verlangt. Inky kann Textstellen suchen und markieren, es kann Teile löschen oder ersetzen, es kann Texte einfügen, auch vor und nach bestimmten Stellen im Text. Es ist also möglich, dem Chat bot beispielsweise den Auftrag zu geben "Markiere mir alle Stellen im Vertrag, in welchen es um Kosten geht.". Er gibt Antwort und wird den Befehl ausführen. Dies wird jeweils im Markup-Modus gemacht, so dass jede Anpassung auch wieder rückgängig gemacht werden kann (die Ausführung kann mit "Esc" abgebrochen werden). Hier wurde zum Beispiel der Befehl erteilt, alle eckigen Klammern aus dem Vertragsmuster zu entfernen, was er auch ausführte:



180 Das Ergebnis ist selbstverständlich zu prüfen – nebst inhaltlichen Fehlern auch, ob es z.B. zu Ersetzungsfehlern gekommen ist, z.B. indem derselbe Begriff mehrfach ersetzt worden ist. Die Programmierung von Inky ist unabhängig von den anderen Funktionen von Red

Ink. Es kann also sein, dass Inky etwas anders umsetzt als beispielsweise die Freestyle-Funktion von Red Ink.

- 181 Nebst diesem integrierten Chatbot Inky gibt es auch noch einen separaten Chatbot, der mit einem normalen Browser benutzt werden kann (dazu Rz. 88 ff.).

#### O. Weitere Tipps zur Handhabung von Red Ink in Word

- 182 Bei **längerem Texten** ist die Nutzung der Markup-Funktion oftmals nicht zielführend, weil die KI dafür zu lange braucht, tendenziell mehr Fehler macht oder Formatierungen nicht erhalten werden können. Dazu gibt es zwei Tipps:

- Insbesondere dort, wo ein grösseres Dokument selektiv kommentiert oder überarbeitet werden soll, hat sich in der Praxis die "**Bubbles:**"-Funktion sehr bewährt: Statt sich einen Markup vom Text machen zu lassen, wird Red Ink bzw. die KI gebeten, den Text mit Kommentaren zu versehen (in Freestyle z.B. "Bubbles: Gehe den Text durch und zeige mir alle Stellen die du sprachlich unklar findest und wie ich es besser machen könnte." eingeben). Die resultierenden Kommentare können dann über das Improve-Untermenü mit "**Apply comment**" automatisch umgesetzt werden. Der Vorteil ist, dass bei dieser Vorgehensweise das Tool nicht mehr den gesamten Text ausgeben muss, was es meistens nicht können wird (Sprachmodelle können nur einen Bruchteil der Textmenge an Output generieren als sie Input aufnehmen können). Eine weitere Strategie bei sehr grossen Dokumenten kann es sein, den Text portionenweise zu bearbeiten.
- Ist die direkte Überarbeitung eines Textes unumgänglich und genügt die "Apply comment"-Funktion nicht, wie zum Beispiel bei Übersetzungen, empfehlen wir, dass schrittweise vorgegangen wird. Es werden **jeweils nur einige Absätze** markiert und dann die betreffende Funktion angewählt. Zum Erhalt der Formatierungen empfehlen wir insbesondere dann, wenn mit Formatvorlagen gearbeitet wird, die Aktivierung der **Keep paragraph format** Option (siehe Rz. 23 oben). Der Einsatz von Formatvorlagen hat auch sonst den Vorteil, dass Dokumente besser und einheitlicher formatiert sind. Zwar ist es möglich, Red Ink zu bitten, sich auch die Formatierungen einzelner Zeichen zu merken (Keep format), aber diese Funktion verlangsamt die Bearbeitung massiv, weil der KI sehr viele Formatierungsangaben mitgegeben werden müssen, damit sie erhalten bleibt, während bei Keep paragraph format nur Absatzformatierungen gemerkt werden. Da ist es erfahrungsgemäß einfacher, absatzweise vorzugehen und einzelne Zeichen- und Wortformatierungen (z.B. Fettschrift) manuell nachzuführen. Wer jedoch ein längeres Dokument übersetzen muss, der verwendet – sofern datenrechtlich erlaubt – mit Vorzug ein anderes

Werkzeug wie "DeepL", welches speziell auf diese Aufgabe ausgelagert ist (es erhält Text als Datei ).

- 183 Beim Einsatz von Red Ink können auch **Tabellen erstellt** werden. Das ist manchmal von Vorteil, weil die Auswertung der Ergebnisse übersichtlicher ist. Hier ist im Befehl die KI anzuweisen, das Ergebnis in Form einer Tabelle darzustellen. Dieser wird zwar nur auf Textcodes mit einem von der KI gewählten Trennzeichen erstellt werden. Mit dem Word-Befehl "Text in Tabelle umwandeln ..." kann dieser Text aber einfach konvertiert werden (es muss dazu nur die Tabelle selektiert, der Befehl aufgerufen und das von der KI gewählte Trennzeichen eingegeben werden):



- 184 Red Ink selbst kann mit **Tabellen im Text** umgehen. Trifft er auf eine solche im selektierten Text, fragt er, ob er die einzelnen Textblöcke davor und danach und in den einzelnen Zellen separat durchgehen soll, damit die Tabelle nicht zerstört wird. Dies generiert dann entsprechend mehr Anfragen. Dies kann auch abgelehnt werden. Wird abgebrochen, geschieht nicht. Ist Markup aktiviert, wird auf die Diff-Methode (nicht DiffW) umgeschaltet, weil sie für diesen Zweck die effizienteste ist.
- 185 Wer **Tabellen von Red Ink überarbeiten lassen** will (z.B. um sie automatisch ausfüllen oder anpassen zu lassen), der sollte dies allerdings nicht in Word machen, sondern die Tabelle in Excel kopieren und Red Ink dort über die Freestyle-Funktion bitten, die Tabelle zu überarbeiten. Das funktioniert wesentlich besser und rascher als in Word.
- 186 Wenn in Word Red Ink ein Text übergeben werden soll, der aus **mehreren Teilen** besteht, die individuell angesteuert werden sollen (z.B. zwei Textstellen im selben Dokument, die verglichen werden sollen), dann hat es sich bewährt, diese mit HTML-artigen "Tags" zu versehen, um so die Inhalte einzugrenzen. Beispiel:

```
<FIRSTPART>
.... Hier kommt der Text des ersten Teils.
</FIRSTPART>
```

<SECONDPART>

*.... Hier kommt der Text des zweiten Teils.*

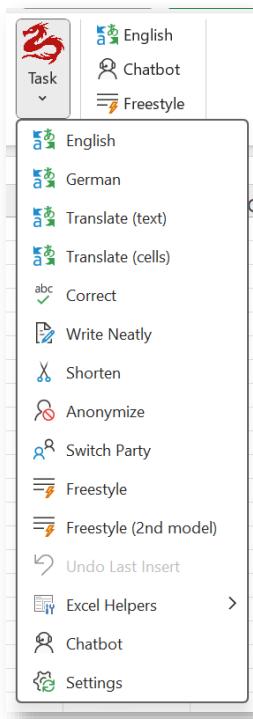
</SECONDPART>

Die verwendeten Bezeichnungen sollten logisch bzw. sinngebend sein. Im Prompt kann darauf verwiesen werden und ein fortschrittliches grosses Sprachmodell wird die Texte unterscheiden können. Das Add-in benutzt diese Technik intern auch (indem Texte, die an die KI übergeben werden mit <TEXTTOPROCESS> und </TEXTTOPROCESS> eingegrenzt werden).

- 187 Die Verwendung von **Absatzmarkierungen** ("\\n") im Prompt kann auch sinnvoll sein, um einzelne Elemente voneinander abzugrenzen. Das kann auch dem Sprachmodell helfen, den Prompt besser zu verstehen und umzusetzen. Es hat sich auch bewährt, **einzelne Schritte einer Instruktion** klar als solche zu kennzeichnen (z.B. "Mache (1) eine Analyse sämtlicher Änderungen im Text und (2) erstelle mir eine Zusammenfassung, der drei wichtigsten Änderungen").
- 188 Das Add-in für Word verarbeitet bei der Ausgabe auch diverse Formatiierungshinweise im **Markdown-Format** (".".md"), wie z.B. Fettdruck, Kursivschreibung, Titel oder Bullets. Diverse Sprachmodelle unterstützen dies. In Freestyle kann das Modell bei Bedarf speziell angewiesen werden, dieses Format bei der Ausgabe zu nutzen ("Output im Markdown-Format"), soweit es dies nicht sowieso von selbst tut. Nicht unterstützt werden hingegen Tabellen und Bilder. Für Tabellen ist wie in Rz. 182 oben vorzugehen.

#### P. **Funktionen von Red Ink in Excel**

- 189 In Excel funktioniert das Add-in wie in Word, allerdings sind die Funktionen leicht anders. Auch hier gibt es vordefinierte Funktionen sowie der Freestyle-Befehl. Der Zugang ist über ein Kontextmenü möglich (falls nicht ausgeschaltet) sowie über die Kacheln, auch Shortcuts können wie in Word definiert werden (siehe oben):



- 190 Da hier mit Zellen gearbeitet wird, kennt Red Ink zwei unterschiedliche Vorgehensweisen, mit denen Funktionen ausgeführt werden: Das Add-in kann angewiesen werden, **Zelle für Zelle** vorzugehen, was zum Beispiel bei Übersetzungen oder Anonymisierungen sinnvoll ist. Oder es wird angewiesen, den **selektierten Zellbereich als Ganzes** zu betrachten, zum Beispiel wenn eine Formel oder ein Textinhalt basierend auf dem bestehenden Inhalt des Arbeitsblattes eingefügt oder nach einer bestimmten Information gesucht werden soll ("Auf welcher Zeile befindet sich sowohl Sarah als auch Davis?", im nachfolgenden Beispiel wurde die Frage auf Englisch gestellt):

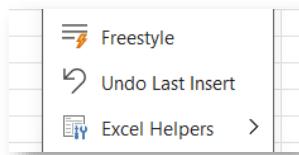
A	B	C	D	E	F	G
1						
2						
3	Sarah	Smith				
4	Mary	Jones				
5	David	Brown				
6	Sarah	Davis				
7	Michael	Wilson				
8	Jessica	Taylor				
9	John	Brown				
10	Ashley	Thomas				
11						
12						

**Red Ink**  
 The LLM has provided the following result (you can edit it):  
 Line 6 contains both Sarah and Davis.

- 191 Alle vordefinierten Funktionen (also z.B. die Übersetzungsfunktion) gehen jeweils Zelle für Zelle vor, weil das der übliche Anwendungsfall ist. Die Funktion Freestyle (Rz. 197 unten) hingegen betrachtet den selektierten Zellbereich standardmäßig als Ganzes, falls nicht bewusst ein Vorgehen Zelle für Zelle verlangt wird.
- 192 Beim Vorgehen Zelle für Zelle muss sichergestellt sein, dass die Zellen nicht **gesperrt** sind. Sind alle selektierten Zellen gesperrt, wird eine

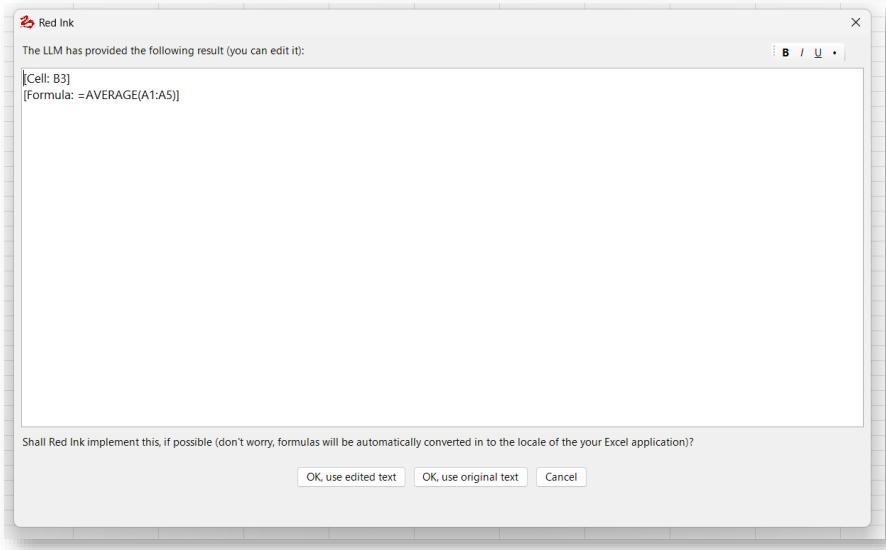
Fehlermeldung angezeigt. Ansonsten werden nur die nicht gesperrten und nicht leeren Zellen bearbeitet.

- 193 Achtung: **Excel ist nicht in der Lage**, die Anpassung der Zellen durch das Add-in **rückgängig zu machen**. Erstellen Sie also vor der Verwendung der KI ggf. eine Kopie der Inhalte (z.B. auf einem separaten Arbeitsblatt). Alternativ können Sie den Befehl **Undo Last Insert** verwenden, der über das Hauptmenü von Red Ink zugänglich ist (beachten Sie jedoch, dass Sie die von Red Ink eingefügten Inhalte nach Verwendung dieses Befehls nicht wiederherstellen können, falls Sie die KI-generierten Inhalte doch wieder haben wollen):



- 194 Im Unterschied zu Word werden in Excel vom Add-in etwaige **Formatierungen** innerhalb eines Textes einer Zelle nicht berücksichtigt, d.h. sie gehen verloren. Die Formatierung der gesamten Zelle wird jedoch nicht geändert.
- 195 Im Menü wird bei den Übersetzungs-Funktionen zwischen "**(text)**" und "**(cells)**" unterschieden. Der Unterschied betrifft die Frage, ob die Funktion nur Zellen bearbeitet, die Text (inkl. Zahlen) enthalten oder auch solche, die Formeln enthalten ("(cells)"). Sind auch Formeln enthalten, wird das Add-in dies berücksichtigen. Das Sprachmodell wird angewiesen, nur die Texte innerhalb von Formeln zu übersetzen, nicht auch die Wörter in den Formeln, da diese sonst nicht mehr funktionieren würden. Die KI kann mittels Freestyle allerdings auch gebeten werden, Formeln anzupassen. Bevor dies operativ eingesetzt wird, sollte mit dieser Funktion experimentiert werden, damit ermittelt werden kann, was am besten wie funktioniert für den konkreten Fall.
- 196 In Excel steht anstelle von "Improve" die ähnliche Funktion "**Write Neatly**" zur Verfügung. Sie unterscheidet sich durch einen leicht anderen internen Prompt, der darauf ausgelegt ist, Stichworte in ganze Sätze umzuwandeln und dabei ggf. einen Kontext zu berücksichtigen (der vorher abgefragt wird, aber optional ist). Die neu formulierten Sätze ersetzen den bisherigen Text in der Zelle.
- 197 Auch in Excel steht die Funktion von **Freestyle** zur Verfügung (siehe Rz. 33 oben). Die von Word bekannten Prefixe und Trigger funktionieren hier allerdings nicht, weil sie in Excel keinen Sinn machen oder normalerweise nicht benötigt werden. Grundsätzlich wird das Add-in beim Befehl Freestyle den Inhalt des gesamten selektierten Zellenbereichs inklusive Werte und Formeln an die KI übergeben, zusammen mit dem Befehl. Auf diese Weise kann die KI zum Beispiel gefragt werden, was der betreffende Bereich bedeutet, wie er funktioniert, oder ob er beispielsweise richtig rechnet. Die KI kann aber auch gebeten wer-

den, eine Formel für eine bestimmte Aufgabe zu entwerfen oder ein Formular auszufüllen ("Fülle mir C5:C11 analog zu D6:D11 aus, aber berücksichtige den Use Case in C3 und die sieben Fragestellungen links."). Das Ergebnis wird in einem Fenster angezeigt und kann bearbeitet werden. Werden Formel- oder Zellwerte zurückgegeben, kann das Add-in diese auch gleich zu implementieren versuchen (sie sind zu diesem Zweck speziell mit eckigen Klammern codiert). Wer gewisse der vorgeschlagenen Anpassungen nicht möchte, kann diese vor der Ausführung im Editor löschen oder manuell anpassen. Das Ergebnis der KI wird auch in die Zwischenablage kopiert. Anpassungen können mit diesem Befehl auch ausserhalb des selektierten Zellenbereichs vorgenommen werden, die KI sieht jedoch nur die selektierten Zellen. Soll die KI also etwas in ihre Überlegungen einbeziehen, muss dies selektiert werden. Wird der Trigger "**(color)**" hinzugefügt, werden auch bestehende Farbinformationen der KI mitgeteilt (z.B. "Fülle mir alle Zellen mit hellblauem Hintergrund mit X aus. (color)"). Es ist aber auch möglich, Freestyle ganz ohne Selektion zu benutzen, wenn es keine Rolle spielt, was das Arbeitsblatt bisher an Inhalten hat. Der Befehl "Berechne in B3 den Mittelwert von A1:A5." resultiert auch ohne Selektion in folgendem Ergebnis (wird danach "OK" gedrückt, wird Red Ink die betreffende Formel in B3 einfügen; sie wird nötigenfalls in das lokale Sprachformat überführt):

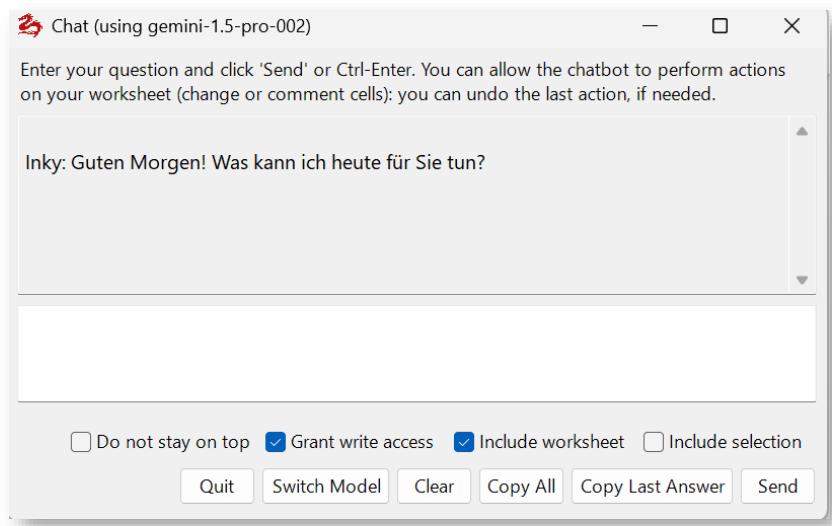


- 198 Soll Freestyle hingegen nur Zelle für Zelle vorgehen, muss dem Befehl der Prefix "**CellByCell:**" oder "**CBC:**" vorangestellt werden. Sollen zudem nur Zellen mit Textinhalt oder einem reinen Zahlenwert bearbeitet werden, dann muss der Prefix "**TextOnly:**" benutzt werden.
- 199 In Excel kann die Freestyle-Funktion analog zu Word die Ausgabe in einer Pane vornehmen. Dazu muss entweder "**Pane:**" vorangestellt werden, oder in der üblichen Ausgabe wird der Knopf "**Transfer to Pane**" verwendet, um den Inhalt in eine Pane zu verlagern. In der Pa-

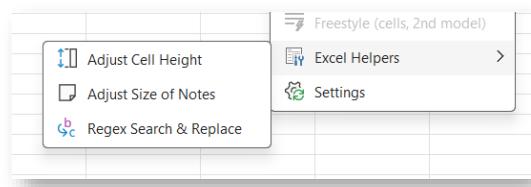
- ne ist es mit dem betreffenden Knopf auch möglich, nur die selektierten Inhalte mit den eckigen Klammern im Excel umsetzen zu lassen.
- 200 Wie in Word kann auch in Excel mit dem Prefix "**Bubbles:**" gearbeitet werden. Der Output wird dann in Form entsprechender Kommentare den betreffenden Zellen hinzugefügt. Ebenfalls wie in Word kann auch der Inhalt externer Dokumente in den Prompt eingelesen werden durch Verwendung des Platzhalters "**{doc}**" (z.B. "Fülle mir die Zellen A1:C20 mit dem Inhalt dieses Textes aus: {doc}"). Ebenfalls unterstützt werden die Trigger "**(file)**" und "**(clip)**" zum Einfügen einer Datei oder des Inhalts der Zwischenablage.
- 201 Freestyle greift normalerweise auf das aktuelle Arbeitsblatt zu. Sollen Daten auch anderer Arbeitsblätter einbezogen werden, ist der Trigger "**(addws)**" in den Prompt aufzunehmen (die Position spielt keine Rolle). Ist er angegeben, kann der Benutzer nach Eingabe des Prompts wählen, welches der derzeit offenen Arbeitsblätter er der KI zusätzlich übergeben will (es spielt keine Rolle, ob die Arbeitsblätter in unterschiedlichen Dateien sind). Wahlweise können auch alle weiteren, ebenfalls offenen Arbeitsblätter übergeben werden.
- 202 Wird dem Befehl "**Batch:**" vorangestellt, dann wird das Add-in den Befehl nacheinander auf alle Text-Dokumente in einem vom Benutzer angegebenen Verzeichnis ausführen. Dies kann benutzt werden, um Inhalte aus solchen Dokumenten (z.B. alle Unterlagen aus einem Fall) zu extrahieren und in einer Excel-Tabelle einzutragen ("Batch: Füge mir in Spalte A die Namen der beteiligten Personen, in Spalte B das Thema in drei Wörtern, und in Spalte C das Datum ein. In Spalte D kommt der Dateiname ohne Endung."). Der Benutzer wird zunächst gefragt, ab welcher Zeile im aktuellen Worksheet die Eintragungen beginnen sollen (die Zeile ist also nicht anzugeben im Prompt) und danach kann der Benutzer das Verzeichnis auswählen, das abgearbeitet wird (nicht jedoch Unterverzeichnisse). Es werden Word, Text und PDF-Dokumente unterstützt (bei PDF-Dokumenten kann auch eine Texterkennung erfolgen, falls das primäre Modell dies unterstützt und das Add-in erkennt, dass er den Text sonst nicht extrahieren kann). Der Durchlauf kann durch Drücken der "Cancel"-Taste unterbrochen werden.
- 203 Auch in Excel stehen für Freestyle natürlich die **Prompt Bibliothek** und der Befehl **Ctrl-P** zur Verfügung, um den zuletzt verwendeten Befehl wieder einzufügen.
- 204 Ein Praxishinweis: Sollen **mehrere Zellen übersetzt** oder sonst bearbeitet werden, kann alternativ auch die Freestyle-Funktion verwendet werden, und zwar so, dass alle Zellen markiert werden, die übersetzt werden sollen und Freestyle der Befehl gegeben wird "Translate to French" (als Beispiel). In diesem Fall werden alle Zellen zugleich an das Sprachmodell übermittelt und es liefert die Anpassungsanweisung in einem Rutsch, die dann umgesetzt werden können. Das funktioniert dann wesentlich schneller als wenn Red Ink jede Zelle einzeln über-

setzt. Allerdings funktioniert dies nicht bei verbundenen Zellen. In diesem Fall werden diese geleert (aufgrund der Art und Weise, wie Excel mit verbundenen Zellen umgeht); dies lässt sich immerhin dadurch verhindern, dass die entsprechenden Kommandos, welche leere Inhalte einfügen sollen, im Fenster zur Bearbeitung der Kommandos vor deren Umsetzung manuell gelöscht werden.

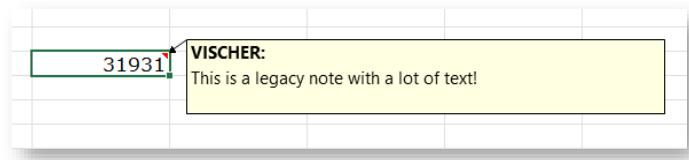
- 205 Zum Schutz vor zu langen Wartezeiten beschränkt Red Ink die Auswahl von Zeilen und Spalten jeweils automatisch auf jene Bereiche, die in Excel tatsächlich benutzt werden. Sollte ein Zellen-Durchlauf zu lange dauern, ist es zudem möglich, diesen mit "**Esc**" abzubrechen.
- 206 Weiter hat das Add-in eine **Chatbot**-Funktionalität, die analog zum Chatbot innerhalb Word arbeitet (Rz. 176 ff.) und entsprechend aufgerufen wird. Auch der Chatbot in Excel kann direkt Änderungen im aktuellen Arbeitsblatt vornehmen, falls dies erlaubt wird (mit der entsprechenden Checkbox "Grant write access"); er kann Zellen mit Inhalten und Formeln füllen und Kommentare einfügen. Je nachdem, ob dies angeklickt wird, wird bei jeder Frage und jedem Auftrag an den Chatbot auch das ganze aktuelle Arbeitsblatt oder die aktuelle Selektion übermittelt. Bei grösseren Arbeitsblättern empfiehlt es sich allenfalls, mit Selektionen zu arbeiten, weil die Übermittlung sonst lange dauert (es erfolgt auch eine Warnung). Bei Selektionen werden der KI automatisch auch die Farbinformationen der Zellen und Schrift mitgeteilt (so dass darauf Bezug genommen werden kann). Anders als beim Chatbot für Word müssen hier die Checkboxen "Include ..." müssen nicht angeklickt sein, damit der Chatbot auf das Arbeitsblatt schreibend zugreifen kann. Falls er keinen Zugriff hat, kann er allerdings beim Schreiben bestehende Zelleninhalte nicht berücksichtigen. Falls Sie eine Änderung des Chatbots rückgängig machen wollen, verwenden Sie die "Undo"-Funktion im Menü von Red Ink. Dies funktioniert nur mit der letzten Änderung und nicht bei Kommentaren. Sollen weitere Arbeitsblätter (nebst dem aktuellen) berücksichtigt werden, kann der aktuellen Eingabe der Trigger "**(addws)**" zugefügt werden. Red Ink wird dann die gegenwärtig ebenfalls noch offenen Arbeitsblätter anzeigen und es kann eins (oder alle) ausgewählt werden. Deren Inhalt wird dann bei dieser Anfrage mitgesandt und steht nur für diese zu Verfügung. Der Trigger muss also bei weiteren Anfragen ggf. wiederholt werden.



207 Das Add-in verfügt ferner über **drei Excel-Helper:**



- **Adjust Cell Height** passt die Zellenhöhe im selektierten Bereich dem Text auch dann an, wenn er über verbundene Zellen verfügt. Die Autofit-Funktion von Excel kann das nicht; bei verbundenen Zellen stellt sie die Zellenhöhe immer auf eine Höhe von einer Textzeile ein, weshalb der Text dann nicht vollständig sichtbar ist.
- **Adjust Size of Notes** passt die Grösse von bei den gewählten Zellen hinterlegten Notizen an. Dies kann manuell eine mühsame Sache sein. Es ist eine Mindestgrösse definiert, und die Zeichengrösse wird bis zu einem gewissen Grad berücksichtigt.

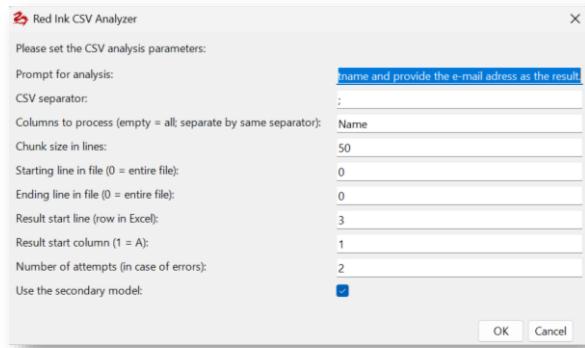


- **Regex Search & Replace:** Diese Funktion funktioniert gleich wie beim Word-Helper (siehe Rz. 175), jeweils Zelle pro Zelle im selektierten Bereich.

208 Über die **Settings** können diverse Konfigurationswerte eingestellt werden, analog zu Word (siehe Rz. 26 ff. oben). Werden sie gespeichert, wird eine lokale Konfigurationsdatei nur für Excel angelegt. Ansonsten wird eine allfällig zentral definierte Konfigurationsdatei oder jene des Add-ins von Word verwendet.

## Q. CSV Analyzer

- 209 Die normale Analyse von Spreadsheets mittels Chatbot oder Freestyle kann bei sehr grossen Excel-Arbeitsblättern an ihre Grenze kommen. Für gewisse dieser Fälle verfügt das Add-in über die Funktion "**Analyze CSV**". CSV ist ein Standardformat für strukturierte Daten ("Comma Separated Valued"). Excel-Arbeitsblätter können als CSV exportiert werden, aber auch Datenbanktabellen. Mit der Funktion können solche CSV-Dateien zeilenweise mittels einem grossen Sprachmodell verarbeitet werden. Das funktioniert dank schrittweise möglichem Abarbeiten auch bei grossen Dateien.
- 210 Wird die Funktion aufgerufen, muss zuerst die CSV-Datei mittels Drag-and-Drop übergeben werden. Danach wird nach dem Trennzeichen (z.B. Semikolon oder Komma) gefragt und die Datei analysiert. Dem Benutzer wird angezeigt, wieviele Zeilen sie hat und welches die Felder der Kopfzeile sind (es wird immer von einer Kopfzeile ausgegangen). Wird bestätigt, dass weitergemacht werden soll, können die Parameter für die Analyse erfasst werden (Red Ink merkt sie sich):



- 211 Erfasst werden können:

- **Prompt for analysis:** Der Prompt, mit dem das Modell die übergebenen Zeilen verarbeiten soll, also beispielsweise "Extract lines where Name indicates a male firstname and provide the e-mail address as the result." Was mit dem Resultat gemacht werden muss, das wird Red Ink dem Modell mitteilen. Das Modell erhält die Anweisung, das Resultat zeilenweise unter Angabe der Zeilennummer (in der CSV-Datei, beginnend mit 1 bei der Kopfzeile) auszugeben, im Beispielprompt also all jene Zeilen unter Angabe der E-Mail-Adresse (sie befindet sich im Feld "Name"), in welcher das Modell in der E-Mail-Adresse einen männlichen Vornamen erkennt. Reale Anwendungen können z.B. sein, dass mit dieser Analyse-Funktion Zeilen ermittelt werden, wo es sensible Daten hat, bestimmte Inhalte mit einer bestimmten Bedeutung vorkommen oder zur Qualitätskontrolle im Nachgang einer Anonymisierung einer Datenbank.

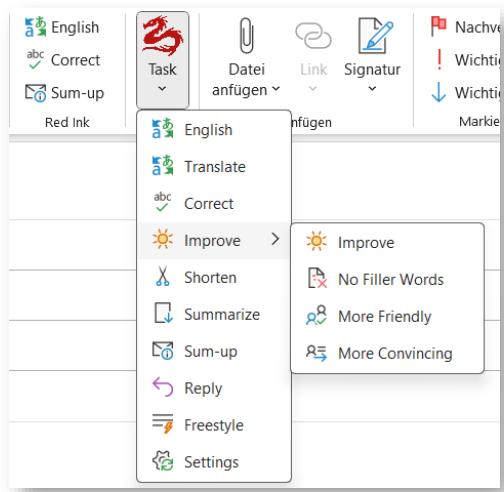
- **CSV separator:** Es ist das bereits erwähnte Trennzeichen, mit welchem die Werte in der CSV-Datei voneinander getrennt werden, also beispielsweise ein Komma oder Semikolon.
- **Columns to process:** Hier wird entweder nichts angegeben, dann werden die gesamten Zeilen aus der CSV-Datei verarbeitet oder es werden die Bezeichnung jener Spalten angegeben, die dem Modell übergeben werden sollen (im obigen Beispiel das Feld "Name"). Das kann bei grossen Tabellen zur Effizienz beitragen, indem nur jene Inhalte ausgelesen werden, die wirklich relevant sind. Werden mehrere Spalten angegeben, sind sie mit dem selben Trennzeichen zu trennen, das schon für die CSV-Datei selbst benutzt wird.
- **Chunksize:** Hier wird angegeben, wieviele Zeilen pro Durchgang an das Sprachmodell übergeben werden sollen. Es sollten nicht zu wenige sein, weil das sonst zu lange dauert, aber auch nicht zu grosse Brocken, weil dies das Modell sonst überfordern könnte.
- **Starting line, Ending line:** Mit Werten über 0 kann hier spezifiziert werden, falls nur ein Abschnitt aus der CSV-Datei ausgelesen werden soll. Das kann z.B. hilfreich sein, wenn eine frühere Analyse für einen Zeilenbereich einen Fehler ausgegeben hat.
- **Result start line, Result start column:** Die Analyse generiert auf dem aktuellen Arbeitsblatt eine Liste, die als Ergebnisbericht dient (siehe unten). Mit diesen beiden Werten kann festgelegt werden, wo auf dem Arbeitsblatt die linke obere Ecke ist (z.B. ganz oben links wäre 1, 1).
- **Number of attempts:** Bei Fehlern des Sprachmodells kann hier angegeben werden, wie oft es das Add-in nochmals versuchen soll.
- **Use a/the secondary model:** Hier kann gewählt werden, ob statt dem Hauptmodell allenfalls das sekundäre Modell oder – falls konfiguriert – eines der weiteren, alternativen Modelle (z.B. ein schnelles, einfacheres Modell) für die Analyse genutzt werden soll.

212 Danach wird Red Ink die CSV-Datei schrittweise (d.h. in Zeilenpaketen der angegebenen Chunk-Grösse) abarbeiten und das Ergebnis anzeigen (laufend). Jeder Chunk kann dabei in null oder einer Mehrzahl von Ergebniszellen mit dem Resultat der Auswertung resultieren. Im Ergebnis wird jeweils die Zeile der CSV-Datei angegeben, auf welche sich das betreffende Resultat passt. Kommt es trotz wiederholter Versuche zu einem Fehler, wird die angegeben. So sieht ein Bericht beispielsweise aus, basierend auf obigem Musterprompt (mit fiktiven Inhalten):

	A	B	C	D	E	F	G	H
1								
2								
3	<b>Analysis Report</b>							
4								
5	Filename:	20250301_ExportTableFromAllDatabases_E-Mail.csv						
6	Date:	11.10.2025 22:26:16						
7	Prompt:	Extract lines where Name indicates a male firstname and provide the e-mail address as the result.						
8	Model:	gemini-2.5-flash						
9								
10	<b>Line(s)</b>	<b>Result</b>						
11	3	schmidt.markus@webmail.net						
12	13	stefan.bauer@mailservice.org						
13	14	ivan.novak@emailprovider.com						
14	16	chris.wagner@mymail.de						
15	17	peter.huber@postbox.net						
16	18	antonio.becker@inbox.org						
17	19	peter.schulz@mail.com						
18	25	urs.hoffmann@email.de						
19	28	tiziano.schaefer@webspace.net						
20	31	miguel.koch@mailhost.org						
21	32	miguel.koch@mailhost.org						
22	33	thomas.klein@emailbox.com						

## R. Funktionen von Red Ink in Outlook

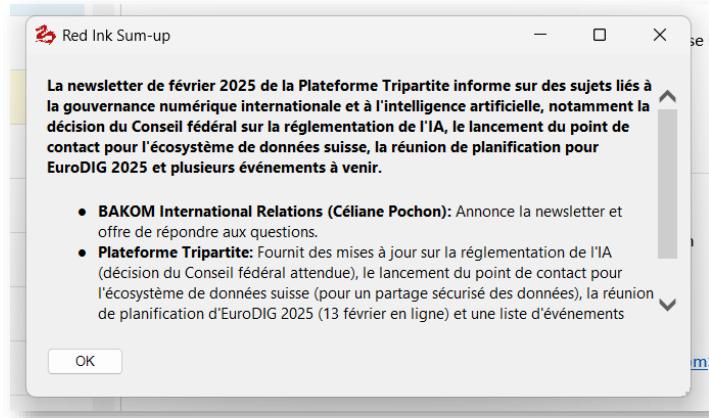
213 Auch in Outlook funktioniert das Add-in wie in Word, allerdings sind auch hier die Funktionen leicht anders. Der Zugang zu den Funktionen ist nur über die Kacheln möglich, weil Outlook keine Add-inspezifischen Kontextmenüs unterstützt. Es können daher auch keine Shortcuts wie in Word und Excel definiert werden. Die Kachel von Red Ink befindet sich sowohl im Hauptmenü als auch im Menü des Fenseters, das beim Verfassen einer E-Mail geöffnet wird (d.h. beim Entwerfen einer neuen E-Mail, beim Antworten auf eine E-Mail oder Weiterleiten einer E-Mail in einem eigenen Fenster). Wird eine E-Mail nur im Seitenfensterbereich von Outlook bearbeitet und ein Befehl von Red Ink angewählt, öffnet Red Ink die betreffende E-Mail in einem separaten Fenster und führt den Befehl erst dann aus. Zu beachten ist weiter, dass nur HTML- und RTF-Mails unterstützt werden, also Mails, welche Formatierungen enthalten können (das kann im Reiter "Text Formatieren" eingestellt werden). Dort erscheint dann die Kachel (sie ist etwas anders positioniert und die Quick Access Kachel ist vorangestellt, weil Outlook sie sonst ggf. zusammenklappt):



- 214 Es stehen beim Add-in für Outlook sinngemäss eine Auswahl derselben Funktionen zur Verfügung, die auch in Word zur Verfügung stehen (siehe Rz. 22 ff. oben), d.h. vorprogrammierte Funktionen sowie die Freestyle-Funktion, über welche ein beliebiger Prompt eingegeben werden kann, ob mit oder ohne selektierten Text (aber keine Helper).
- 215 Ein praktischer Hinweis für die Benutzung von **Translate**: Bei Zielsprachen, wo zwischen Duzen und der Höflichkeitsform unterschieden wird (wie z.B. im Deutschen), ist die KI angewiesen, anhand des Kontextes und der bereits verwendeten Wörter die richtige Wahl zu treffen. Wird eine Person mit Nachnamen angesprochen oder eine Grussformel mit vollständigem Namen verwendet, wird sie davon ausgehen, dass die Höflichkeitsformel nötig ist. Beim Übersetzen sollte also die Anrede oder die Grussformel mitselektiert werden, damit die KI das berücksichtigen kann. Sie sieht nur, was selektiert ist. Das funktioniert auch in Word so.
- 216 Wie in Word (dort unter den World Helpers) steht auch in Outlook die Funktion **Clipboard to Text** zur Verfügung, sofern das jeweils konfigurierte primäre Modell diese Funktion unterstützt. Sie funktioniert so, dass das Modell gebeten wird, den Inhalt der Zwischenablage in Text umzuwandeln (z.B. Text in einem Screenshot, Text in einer Sprachmitteilung). Wird gerade eine E-Mail verfasst, fügt die Funktion diesen Text dort ein. Ist das nicht der Fall, wird die Zwischenablage nach Aufruf dieser Funktion mit dem von der KI extrahierten oder generierten Text befüllt, und er kann so in eine beliebige Anwendung eingefügt werden.
- 217 Eine Outlook-spezifische Funktion ist **Reply**. Sie dient zur Vorbereitung von Antworten auf E-Mails. Hier müssen zuerst jene Teile der bisherigen E-Mail-Kette selektiert werden, auf die geantwortet werden soll, wobei die oberste Mail des angewählten Bereichs die E-Mail sein sollte, auf die unmittelbar geantwortet wird (wird nichts gewählt, wird die ganze Mail-Kette benutzt). Das Add-in wird diese Reihenfolge berücksichtigen. Es können über das sich öffnende Fenster konkrete Instruk-

tionen und Informationen für die Formulierung der Antwort eingegeben werden; werden keine Instruktionen eingegeben, dann wird die aus Sicht der KI wahrscheinlichste Antwort gegeben. Sind MyStyle-Prompts definiert (für den persönlichen Schreibstil, Rz. 48 ff.), wird der Benutzer daraufhin gefragt, ob und welchen er zusätzlich verwenden will. Die Antwort wird ganz oben eingefügt und kann frei bearbeitet werden.

- 218 Die Funktion **Sum-up** fasst die E-Mail-Kette zusammen (nach Personen). Die KI ist angewiesen, in der Sprache der E-Mail zu arbeiten. Um die Funktion zu nutzen, muss die betreffende E-Mail geöffnet und "Antworten" oder "Weiterleiten" geklickt werden. Dann die relevanten Teile selektieren (oder es wird automatisch die ganze Mail-Kette berücksichtigt) und die Funktion "Sum-up" wählen. Die KI wird oben in der Mail eine Zusammenfassung der Mail-Kette wiedergeben. Das kann bei längeren Mails nützlich sein, um einen raschen Überblick zu erhalten.
- 219 Im Gegensatz zu allen anderen Funktionen von Red Ink für Outlook können die Funktion Sum-up und Translate **auch bei nicht zum Schreiben geöffneten E-Mails** benutzt werden. Es genügt, in der Übersicht von Outlook eine E-Mail anzuwählen, damit sie im Feld rechts angezeigt wird. Wird dann Sum-up gedrückt, erstellt Red Ink eine Zusammenfassung bzw. erstellt eine Übersetzung (ggf. in der einzugebenden Zielsprache) und zeigt sie in einem separaten Fenster an:



- 220 Wenn **mehrere E-Mails** ausgewählt wurden (ohne dass eine geöffnet ist), dann wird der Sum-up-Befehl alle selektierten E-Mails lesen (wobei versucht wird, jeweils nur die neueste Mail jeder E-Mail-Kette zu extrahieren, um Zeit zu sparen) und einen Überblick über die wichtigsten und dringendsten E-Mails liefern. E-Mails, die bereits beantwortet wurden, werden nicht berücksichtigt.
- 221 Gewisse der Funktionen (bei Freestyle durch Zufügen von "Markup:" vor dem Prompt) arbeiten mit einer Markup-Funktion, obwohl Outlook eine solche eigentlich nicht anbietet. In diesem Fall wird zuerst der Text der KI angezeigt, gefolgt von "MARKUP:" und einem Markup, welcher farblich die Einfügungen und Löschungen der KI zeigt. Weil die Markups technisch innerhalb von Outlook nicht "angenommen" werden

können, erfolgte die Ausgabe lediglich zur Information. Sie muss vom Benutzer, sobald er mit seinem Text einverstanden ist, manuell gelöscht werden. Es kann in Settings wie in Word (Rz. 25) eingestellt werden, welche Markup-Methode verwendet werden soll, allerdings steht in Outlook nur "Word", "Diff" und "DiffW" zur Verfügung, weil nur diese hier technisch Sinn machen. Für "Diff" wird dieselbe Zeichenobergrenze benutzt wie in Word und gleich gehandhabt. Allerdings kann unabhängig von Word konfiguriert werden, ob bei den vorprogrammierten Funktionen, wo das Sinn macht (also z.B. bei "Correct" und "Shorten"), automatisch ein Markup erstellt wird. Auch hier ist es wie in Word möglich, die "Diff"-Markup-Ausgabe mit der Taste "**Esc**" abzubrechen, falls sie zu lange dauert. In der Praxis hat sich "DiffW" für die alltägliche Nutzung am besten bewährt.

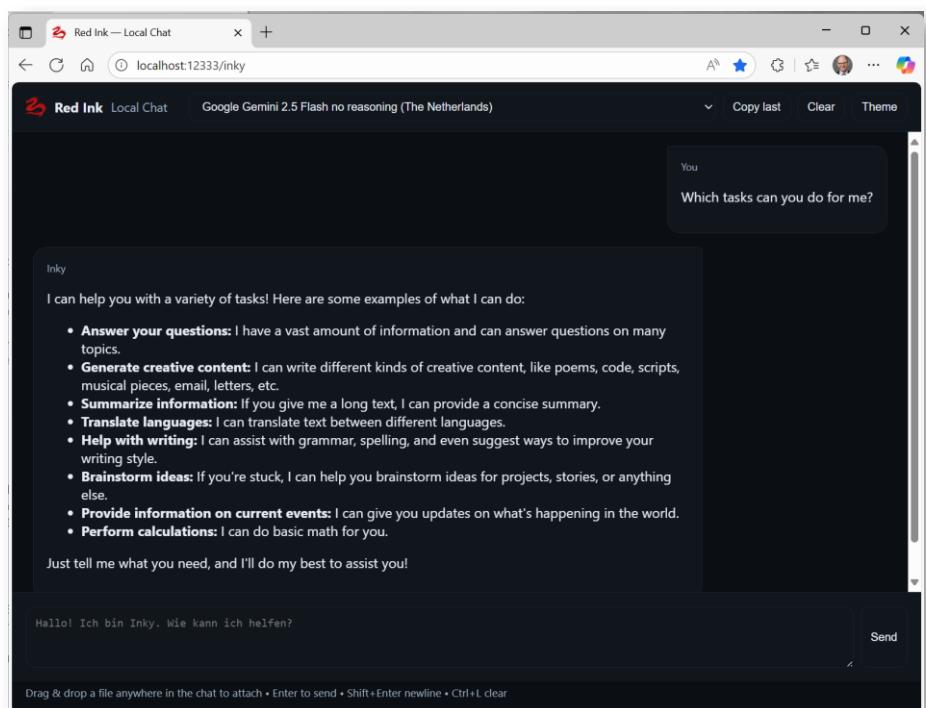
- 222 Wie in Word (Rz. 25) kann bei allen Funktionen ausser Freestyle, Reply und Sum-up ferner über die Konfiguration eingestellt werden, ob der von der KI generierte Text den **bisherigen Text ersetzt** oder ob er danach eingefügt wird.
- 223 Es kann weiter wie in Word (Rz. 25) konfiguriert werden, ob das Add-in bei der Übersetzungsfunktion, der Korrektur-, Verbesserungs- und der Kürzungsfunktion grundlegende **Formatierungen** (wie Schrift und Aufzählungen) zu erhalten versuchen soll oder ob er im reinen Textmodus arbeitet (was zur Folge hat, dass in der Ausgabe spezielle Formatierungen verloren gehen; Red Ink versucht jedoch einfache Formatierungen wie Fettschrift zu erhalten). Letzteres braucht mehr Zeit, weil der KI alle Formatierungen ebenfalls übergeben werden müssen (was im Format HTML geschieht). Um die Antwortzeiten im Rahmen zu halten, werden nur die wichtigsten Formatierungen erhalten. Bei der Wiedergabe ist es möglich, dass diese nicht exakt der bisherigen Darstellung entspricht. Die Funktion zur Begrenzung des Erhalts der Formatierungen auf eine bestimmte Zeichenzahl steht auch hier zur Verfügung und kann via Settings konfiguriert oder mit 0 ausgeschaltet werden.
- 224 Auch die **Freestyle**-Funktion steht in Outlook zur Verfügung, allerdings mit reduziertem Umfang gegenüber Word. Es kann allerdings auch hier wie bei Word mit dem Prefix "**Markup:**", "**MarkupWord:**", "**MarkupDiff:**" und "**MarkupDiffW:**" gearbeitet werden, nicht jedoch "MarkupRegex:", weil das für Outlook keinen Sinn macht (siehe dazu Rz. 25 und Rz. 40). Möglich ist ferner "**Replace:**", welches die Antwort des Sprachmodells an Stelle des selektierten Textes setzt (z.B. "Replace: Finde mir eine freundlichere Formulierung für diesen Satz").
- 225 Die in Word verfügbaren Trigger sind in Outlook nicht verfügbar, weil sie in Outlook normalerweise nicht gebraucht werden. Die Funktionen zum Erhalt der Formatierungen werden in Freestyle für Outlook (im Gegensatz zur Word-Version) aber nicht unterstützt. Verfügbar ist in Freestyle in Outlook jedoch auch die Prompt Bibliothek, und der letzte

Prompt, der im aktuellen Mail-Fenster verfasst wurde, kann via **Ctrl-P** eingefügt werden.

- 226 Für den Zugriff auf das **sekundäre** und etwaige **weitere Sprachmodelle** ist in Outlook kein eigener Freestyle-Befehl vorhanden. Stattdessen kann dieses andere Modell durch Zufügen des Triggers "**(2nd)**" angewählt werden, sofern es konfiguriert worden ist. Dann geht der Prompt an das sekundäre Sprachmodell. Welche das ist, kann über Settings (siehe Rz. 26 ff. oben) angeschaut werden oder wird angezeigt, wenn mit der Maus über das Red Ink Logo gefahren wird. Es ist in Settings auch möglich, die beiden KI auszutauschen. Sind alternative Modelle konfiguriert, kann der Benutzer auswählen, welches davon er nutzen möchte.
- 227 Wer die Einstellungen von Red Ink in Outlook ändern möchte, kann dies über die **Settings**-Funktion temporär oder dauerhaft tun (zu den einzelnen Werten siehe Rz. 26 ff. oben). Werden die Einstellungen einfach nur geändert, bleiben sie nur solange Outlook nicht geschlossen wird; danach kommen wieder die vorkonfigurierten Einstellungen zur Anwendung. Wer das nicht will, kann in Settings die Einstellungen in einer lokalen Kopie der Konfigurationsdatei speichern. Das erfolgt automatisch, sobald in Settings die Konfiguration gespeichert wird. In diesem Fall wird die "lokale" redink.ini-Datei (im Outlook-Verzeichnis und nur für Outlook) überschrieben (jedoch nicht eine allfällige zentrale Datei, wie sie über die Registry bereitgestellt werden kann, siehe Rz. 320 unten); die lokale Datei wird prioritär gelesen. Es ist auch möglich, über ein "Expert"-Fenster auf die weiteren Konfigurationsparameter zuzugreifen, aber wir empfehlen dies nicht (besser ist es, direkt in der Konfigurationsdatei Anpassungen vorzunehmen). Wird beim Addin für Outlook (oder Excel) selbst keine Konfigurationsdatei gefunden, wird jene von Word gesucht. Wer **für Outlook eine separate Konfiguration** verwenden möchte, speichert daher am besten eine separate Konfigurationsdatei für Outlook ab (durch Speichern der Konfiguration in Settings oder manuell).

#### S. **Separater Chatbot "Inky"**

- 228 Nebst den integrierten Chatbots in Word und Excel können das primäre und sekundäre Modell sowie alle alternativen KI-Modelle (soweit konfiguriert) auch über einen klassischen Chatbot in einem beliebigen Browser auf dem Computer des Benutzers verwendet werden. Er wird über das Red Ink Add-in von Outlook betrieben, d.h. dieses Add-in verfügt über einen kleinen Webserver mit dem Chatbot. Er kann über die Adresse **localhost:12333/inky** abgerufen werden (einfach so im Browser eingeben), sobald Outlook gestartet und initialisiert ist:



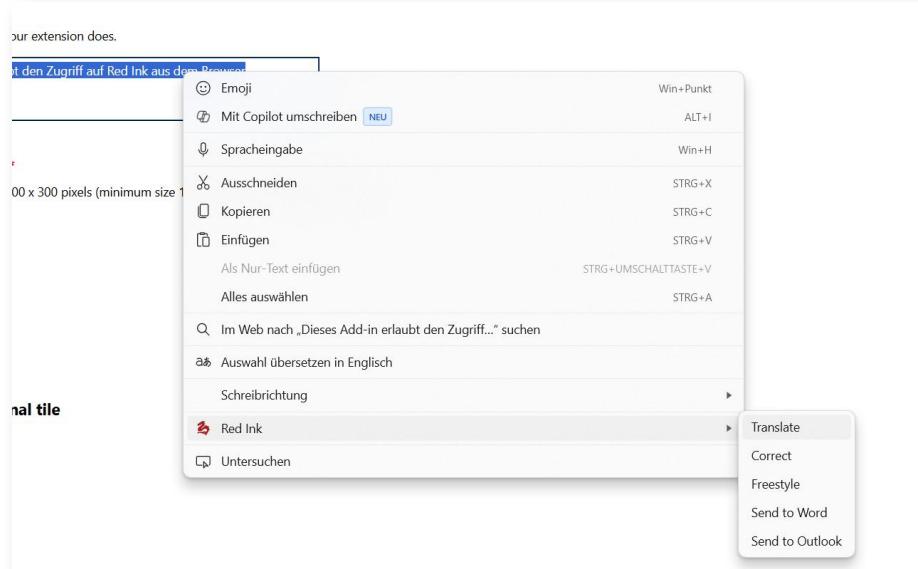
- 229 Das jeweilige Modell kann über das Drop-Down-Menü im Titel ausgewählt werden. Es kann mit einem Knopfdruck die letzte Antwort in die Zwischenablage kopiert und die Anzeige gelöscht werden. Der Verlauf wird automatisch zwischengespeichert. Dieser lokale Chatbot nutzt dieselbe Zeichenzahlbegrenzung wie die beiden integrierten Chatbots, hat aber einen eigenen System-Prompt.
- 230 Es können Word-, Powerpoint-, Excel- und normale Text-Dateien durch Drag & Drop an den Chatbot übergeben werden. Dateien in weiteren Formaten können ebenfalls auf dieselbe Weise übergeben werden, falls das Modell entsprechend konfiguriert ist (z.B. PDF, Bilder).
- 231 Liefert das Modell eine Bilddatei zurück, wird in der Antwort der Pfad angezeigt und die Datei auf dem Desktop des Benutzers gespeichert (oder es wird im Chat angezeigt, je nach Modell). Normale Textantworten können mit "Copy last" in den Zwischenspeicher übernommen werden. Wird Programmcode angezeigt, steht ein Knopf zum Kopieren nur des Programmcodes zur Verfügung.
- 232 Der Chatbot eignet sich gut für separate Brainstorming- und Recherche-Sitzungen mit der KI nach der Wahl des Benutzers, ohne dass

Word gestartet werden muss. Sind Modelle mit Internet-Suchfunktion konfiguriert, kann auf diese Weise auch recherchiert werden, ohne, dass der Benutzer sich in die diversen KI-Dienste einloggen muss (ein Login entfällt innerhalb des Red Ink Local Chat).

- 233 Im Chatbot können zwei getrennte Chats geführt werden. Es kann durch Anklicken des Knopfs mit der Zahl 1 und 2 oben rechts zwischen beiden hin- und hergeschaltet werden. Der Chat wird auch nach Schliessen des Browser-Fensters oder von Outlook gespeichert (mit "Clear" kann er gelöscht werden).
- 234 Für spezielle Anfragen bietet der Chatbot ferner den Knopf "Pure" an. Er wird anstelle von "Send" benutzt. In diesem Fall wird dem Chatbot nur gerade der Text übermittelt, den der Benutzer eingegeben hat, weder ein Systemprompt, noch weiterer Kontext. Dies kann nützlich sein, wenn über den Chatbot direkt ein Modell angesteuert werden soll, bei welchem sehr genau kontrolliert werden muss, was er erhält (z.B. ein Spezialmodell zur Bildgenerierung, wo weiterer Text die Erstellung des Bilds beeinflussen würde).
- 235 Auf Knopfdruck ("Theme") kann zwischen einer hellen und dunklen Anzeige umgestellt werden.

## T. Browser-Erweiterung

- 236 Red Ink verfügt auch über eine Erweiterung die für Chromium-basierte Browser (z.B. Edge, Chrome). Ist sie installiert, können Benutzer Text in ihrem Browser auswählen (z.B. beim Arbeiten auf einer Website) und Red Ink bestimmte Aktivitäten mit dem ausgewählten Text ausführen lassen:



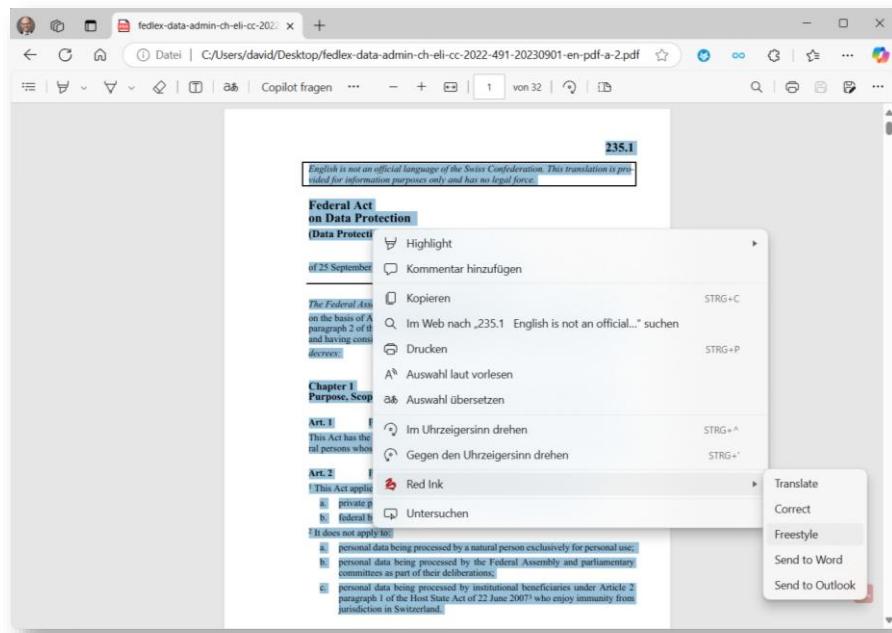
- 237 Die Befehle sind:

- **Translate:** Es geht ein Fenster auf und der Benutzer gibt die gewünschte Zielsprache an. Der übersetzte Text wird an Stelle des selektierten Textes ausgegeben.
- **Correct:** Der Text wird sprachlich korrigiert und ein Markup wird in einem Fenster angezeigt. Wird nicht "Esc" gedrückt, wird der selektierte Text durch den korrigierten Text (ohne Markup) ersetzt.
- **Freestyle:** Es kann ein beliebiger Prompt eingegeben werden (auch die Prompt-Bibliothek steht zur Verfügung). Allerdings funktionieren die Formatierungsbefehle und Funktionen wie "Bubbles" hier natürlich nicht. Standardmäßig werden die Antworten der KI in einem separaten Fenster angezeigt (und in die Zwischenablage abgelegt), so wie wenn in Word dem Prompt "Clipboard:" vorangestellt wird. Wer jedoch möchte, dass Red Ink den generierten Output an den Browser zurücksendet, damit er dort eingefügt wird, muss dem Prompt den Prefix "**Insert:**" voranstellen. Auch Markups sind möglich, analog zu Correct. Hierzu muss "**Markup:**" vorangestellt werden; dieser Prefix beinhaltet zugleich ein "Insert:", d.h. der Inhalt wird an den Browser zurückgesandt, falls nicht abgebrochen wird. Freestyle steht als einziger Befehl auch dann zur Verfügung, wenn im Browser kein Text selektiert ist.
- **Send to Word:** Der im Browser selektierte Text wird an der aktuellen Stelle im Dokument in Word eingefügt (ohne Copy & Paste).
- **Send to Outlook:** Der im Browser selektierte Text wird an der aktuellen Stelle im Dokument in Outlook eingefügt, sofern ein Fenster zum Verfassen einer Mail offen ist (ohne Copy & Paste).

- 238 Die Browser Extension funktioniert so, dass die dort installierte Software den markierten Text lediglich im Hintergrund an das Red Ink Add-in für Outlook sendet (bzw. an jenes für Word bei der "Send to Word"-Funktion) und der Text dann dort bearbeitet wird. Outlook muss daher mit Add-in "laufen", damit die Browser-Erweiterung funktioniert. Ist das nicht der Fall, geschieht einfach nichts. Je nach Situation kann es allerdings sein, dass das Fenster, welches sich beispielsweise bei Freestyle oder Translate in Outlook öffnet, auf den ersten Blick gar nicht bemerkt wird (Red Ink ist so programmiert, dass es sich in den Vordergrund schieben soll, aber das funktioniert möglicherweise nicht immer).
- 239 Die Antwort von Red Ink wird nach der Bearbeitung wieder an die Browser-Erweiterung zurückgesandt (im Falle von Freestyle nur beim Prefix "Markup:" oder "Insert:"), welche damit den selektierten Text ersetzt (wer das nicht möchte, muss vorher "Esc" drücken, wo ein Dialog innerhalb von Red Ink erfolgt). Es kann je nach Programmierung der Seite, welche im Browser angezeigt wird, allerdings geschehen,

dass der zurückgespielte Text an der falschen Stelle eingefügt wird (z.B. oberhalb des Eingabefelds). Gewisse Seiten wiederum sperren die Anzeige des Kontext-Menüs von Red Ink (z.B. wenn in Google Docs gearbeitet wird). Achtung: Der Browser wird nicht blockiert während Red Ink arbeitet. Wird im Browser inzwischen weitergearbeitet, kann es geschehen, dass der von Red Ink retournierte Text an der falschen Stelle eingefügt wird.

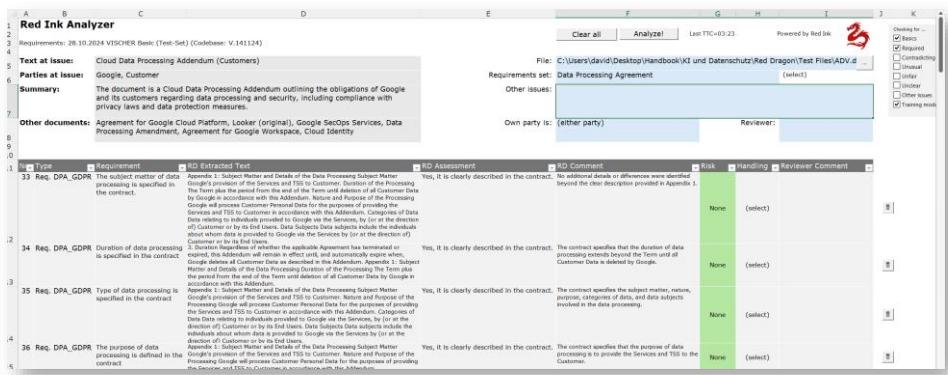
- 240 Im Alltag hat sich die Browser-Erweiterung vor allem bei der **Analyse von Inhalten von Webseiten**, aber auch **von PDF-Dokumenten** bewährt. Hier wird einfach die betreffende Webseite teilweise oder komplett markiert (z.B. mit Ctrl-A) und der Befehl Freestyle verwendet. Es kann dann eine Frage gestellt werden. Soll ein PDF-Dokument analysiert werden, so muss dieses im Browser statt im PDF-Reader geöffnet werden (Voraussetzung ist, dass im Browser kein PDF-Plugin verwendet wird, welches dazwischenfunkt). Es kann dann auch dort der gesamte Text selektiert und mittels Freestyle befragt werden (z.B. "Wo wird die Frage der Einwilligung behandelt?"):



## U. Verwendung von Red Ink durch andere Programme

- 241 Das Add-in von Excel kann mit seinem Helper-Programm auch von anderen Anwendungen benutzt werden, um auf Sprachmodelle zuzugreifen. Damit lassen sich in Excel sehr einfach Lösungen bauen, die ein Sprachmodell nutzen, weil sich der Benutzer, der sie programmiert, nicht mehr um die Schnittstelle kümmern muss. Das tut alles Red Ink, das im Hintergrund läuft, wann immer Excel benutzt wird. Ferner bietet das Add-in mit dem Helper noch die Möglichkeit, den Inhalt von PDF-Dateien einzulesen (wie der entsprechende Word-Helper). Das kann nützlich sein, weil Excel in VBA selbst diese Möglichkeit nicht bietet.

- 242 Ein Beispiel für eine solche Anwendung ist der **Red Ink Analyzer**, ein Werkzeug, das einerseits zur Analyse von juristischen Texten (z.B. Verträgen oder Datenschutzerklärungen) auf vordefinierte Anforderungen oder andere Probleme hin verwendet werden kann, und andererseits, um Dokumente systematisch mittels KI auszuwerten (z.B. Zusammenfassungen von Beweismitteln eines Verfahrens zu erstellen, aus einer Serie von Dokumenten spezifische Informationen extrahieren oder Dokumente einer internen Untersuchung auf bestimmte verdächtige Inhalte hin prüfen zu lassen). Dieses Tool bieten wir bei kommerziellem Einsatz gegen eine moderate Lizenzgebühr an.



- 243 Um aus einer Excel-Anwendung auf die LLM-Schnittstelle von Red Ink zuzugreifen, muss das Add-in für Excel sowie der Helper installiert und geladen sein (zum Helper siehe Rz. 291 unten).

- 244 Die **LLM-Schnittstelle** kann von anderen VBA-Modulen in Excel oder auch direkt aus einer Excel-Zelle benutzt werden. Der Aufruf hat folgende Syntax:

```
Antwort = LLM(SysPrompt, UserPrompt, Model, Temperatur,
 Timeout, SecondAPI, Hidesplash)
```

- 245 Die Parameter sind wie folgt:

Key	Typ	Bedeutung
Antwort	String	Der Output des Sprachmodells, wobei Escape-Zeichen bereits bereinigt sind
SysPrompt	String	Der System-Prompt, wobei der Prompt vor der Verwendung noch für JSON bereinigt wird
UserPrompt	String	Der User-Prompt, wobei der Prompt vor der Verwendung noch für JSON bereinigt wird
Model	String, optional	Model-Name (falls die Angabe für die API benötigt oder von ihr unterstützt wird); soll der Default-Wert verwendet werden, dann ist "" zu übergeben
Temperatur	String, optional	Temperatur (falls die Angabe für die API benötigt oder von ihr unterstützt wird); soll der Default-Wert verwendet werden, dann ist "" zu übergeben
Timeout	Long, optional	Timeout in Millisekunden; soll der

		Default-Wert verwendet werden, dann ist 0 zu übergeben
SecondAPI	Boolean, optional	True, falls das allfällige konfigurierte sekundäre Sprachmodell für die Abfrage benutzt werden soll, sonst False; Default-Wert ist False
Hidesplash	Boolean, optional	True, falls das Splash-Fenster mit dem Red Ink-Logo während einer Abfrage nicht angezeigt werden soll (erscheint bei grösseren Abfragen); Default-Wert ist False

- 246 In Excel kann die obige Funktion beispielsweise wie folgt aus einem anderen Modul aufgerufen werden:

```
result = Application.Run("redink_helper.xlam!LLM", SysPrompt,
UserPrompt)
```

Oder:

```
result = Application.Run("redink_helper.xlam!LLM", SysPrompt,
UserPrompt, "", "", 0, False, False)
```

- 247 Wer die LLM-Schnittstelle des Helpers testen will, kann die Prozedur "TestLLM" aufrufen. Nach kurzer Zeit sollte ein Fenster mit einer Antwort des jeweiligen Sprachmoduls erscheinen.

- 248 Die Funktion, um den **Text von Dateien einzulesen**, hat folgende Syntax:

```
Antwort = GetFileTextContent(Dateiname, FehlerInAntwort)
```

- 249 Die Parameter sind wie folgt:

Key	Typ	Bedeutung
Antwort	String	Der Textinhalt der Datei (oder ggf. den Fehlercode, beginnend mit "Error")
Dateiname	String	Der vollständige Dateiname mit Pfad, wobei Umgebungsvariablen automatisch ersetzt werden
FehlerInAntwort	Boolean, optional	True, wenn die Funktion im Falle eines Fehlers den Text "Error" plus eine Beschreibung des Fehlers liefern soll, ansonsten liefert sie einen leeren String

- 250 In einem Modul erfolgt der Aufruf analog dem obigen Beispiel für die Funktion LLM.

- 251 Schliesslich steht auch die Funktion **Adjust Cell Height** zur Verfügung, mit welcher die aktuell selektierten Zellen in ihrer Höhe auch dann automatisch angepasst werden an den Textinhalt, wenn sie verbunden sind (Excel kann das selbst nicht). Der Aufruf erfolgt über folgende Prozedur und in einem Modul analog dem obigen Beispiel für die Funktion LLM:

```
AdjustHeight()
```

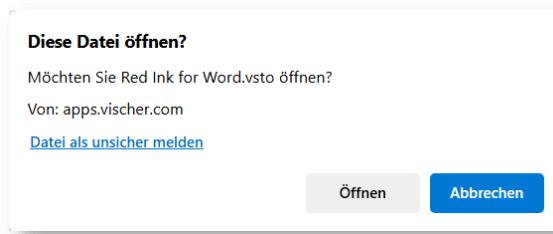
### III. INSTALLATION

#### A. Für Ungeduldige: Die Installation mit einem Klick

- 252 Die schnellste und einfachste Installation von Red Ink erfolgt über die Website <https://apps.vischer.com>, wo das jeweilige Add-in mit einem Klick auf den entsprechenden Button installiert werden kann:



- 253 Achtung: Diese Art der Installation mit einem Klick funktioniert derzeit **nur mit dem Browser Edge**, aber nicht mit Chrome, Firefox oder einem anderen Browser.
- 254 Der Browser wird typischerweise warnen, weil er dieses Programm nicht kennt. Es muss in diesem Beispiel "Öffnen" angeklickt werden, um die Installation auszuführen:



- 255 Die Installation erfordert keine weiteren Eingaben. Ist sie fertig, müssen Word, Excel oder Outlook neu gestartet werden und Red Ink steht zur Verfügung. Beim ersten Start werden die Minimalangaben konfiguriert, d.h. es muss der geheime Zugangscode (i.d.R. der sog. API-Key) zum Sprachmodell der Wahl eingegeben werden. Wer ein Abo für "ChatGPT" hat, kann sich diesen von OpenAI geben lassen. Dieser wird in die Maske eingefügt. Dies genügt normalerweise und es kann mit Red Ink gearbeitet werden. Über Settings können dann weitere Einstellungen vorgenommen werden. Mehr dazu in den nachfolgenden detaillierten Ausführungen, auch zur Installation der Helper-Funktionen. Updates können durch erneutes Klicken der Buttons installiert werden, werden aber nach einer gewissen Zeit auch automatisch vorgeschla-

gen. Wer Red Ink im Unternehmen installieren will, braucht unter Umständen eine entsprechende Freigabe, weil die Sicherheitsfilter eine Installation blockieren können.

## B. Die Installation im Detail

256 Es gibt derzeit drei Methoden, Red Ink zu installieren:

**Methode 1:** Installation direkt über einen Installer im Internet

Dies erfolgt wie vorstehend beschrieben über unseren Deployment-Server <https://apps.vischer.com> und ist die einfachste Methode. Damit werden allerdings nicht beide Helper-Dateien installiert, falls diese gewünscht werden; dies kann aber nachgeholt werden. Der Nachteil dieser Methode ist, dass sie je nach Sicherheitseinstellungen blockiert wird, vor allem in Betrieben. Diese Methode eignet sich daher eher für **Privatpersonen** und **kleine Betriebe**.

- a) **Methode 2:** Download des Installationspaketes und vollständig lokal ausgeführte Installation

Es werden die Dateien zuerst heruntergeladen und aus einer lokalen Quelle installiert bzw. die zwei Helper-Dateien – sofern erwünscht – in bestimmte Verzeichnisse kopiert (was auch nachträglich noch möglich ist). Diese Methode eignet sich eher für **größere Betriebe**, die Red Ink ihren Mitarbeitenden zur Verfügung stellen aber nicht den Zugriff auf die Dateien im Rahmen von Methode 1 freigeben wollen.

- b) **Methode 3:** Installation via Image- oder Softwareverteilung

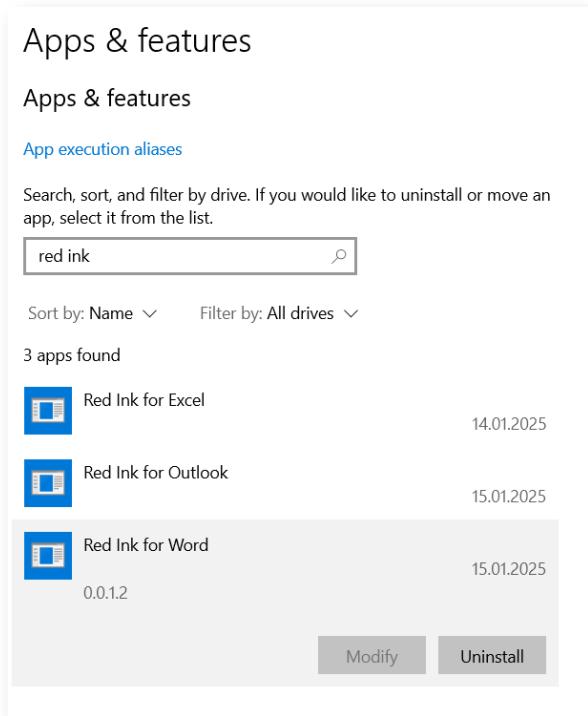
Sie ist für Betriebe gedacht, die Red Ink intern noch stärker kontrolliert bzw. vorinstalliert verteilen möchten. Auf diese gehen wir hier nicht näher ein. Red Ink hat allerdings auch Funktionen, damit es in **Betrieben mit zahlreichen Benutzern** eingesetzt werden (siehe u.a. Rz. 309 ff. unten).

257 Vor allem in Betrieben kann der Zugriff auf die Installationsdateien sowohl von Methode 1 als auch Methode 2 **aus Sicherheitsgründen blockiert** sein. In diesen Fällen kann nur der Administrator der IT-Umgebung weiterhelfen und die Dateien (z.B. basierend auf ihrer digitalen Signatur von uns, VISCHER AG, freigeben).

258 Es kann auch sein, dass **Windows Defender** (oder auch ein anderes Antiviren-Programm) einzelne Installationsdateien (insbesondere der Excel-Helper) fälschlicherweise als Bedrohung (namentlich den Trojaner "O97M/Sadoca.C!ml" aus dem Jahre 2020) erkennt und in die Quarantäne verschiebt. Dasselbe gilt für die Helper-Dateien. Ein solcher Fehlalarm kann je nach den Sicherheitseinstellungen in Windows übersteuert werden (die Dateien können auf eine "weisse Liste" gesetzt werden). Manchmal hilft es auch, den Computer neu zu starten und etwas zu warten. Wir haben solche Fehlalarme vor allem dann beo-

- bachtet, wenn Red Ink mehrfach hintereinander installiert wird. Wir haben das Problem Microsoft gemeldet.
- 259 Wird Red Ink zum ersten Mal installiert, erfolgt eine **minimale Konfiguration**. Ansonsten kommen die Standardwerte zum Einsatz. Über die manuelle Bearbeitung der Konfigurationsdatei ist auch eine sehr viel differenzierte Konfiguration möglich. Red Ink kann in Unternehmen auch automatisch verteilt und hat Funktionen, damit es für mehrere Benutzer gleichzeitig eingesetzt werden kann
- 260 Zur Installation der nötigen Dateien für den **Transcriptor** (die Sprachmodelle und ggf. Programmbibliotheken) siehe Rz. 97 ff. oben.
- 261 Red Ink hat auch eine **Update-Funktion**. Sie hängt davon ab, ob Methode 1 oder Methode 2 verwendet worden ist:
- Bei Methode 1 erfolgt das Update durch Anklicken der Installationslinks oder durch die eingebaute Update-Funktion (im Settings-Menü). Die Add-ins sind zudem so konfiguriert, dass sie alle drei Tage nach Updates prüfen und ein solches selbst installieren können (das kann über die Konfigurationsdatei verändert werden). Updates werden entweder durch einen interne Aufruf von <https://apps.vischer.com> ausgeführt (der Benutzer muss nur die Vornahme des Updates bestätigen, sieht sonst aber nichts) oder so durchgeführt, dass die Add-ins den Microsoft-Edge-Browser und dort den auf <https://apps.vischer.com> hinterlegten VSTO-Installer aufrufen (d.h. es geschieht dasselbe wie Anklicken der roten Knöpfe auf derselben Seite). Das Ausführen des Installers muss im Browser manuell bestätigt werden.
  - Bei Methode 2 erfolgt das Update, indem eine neue Version des Installationspaketes heruntergeladen wird und über die bestehenden Installer-Verzeichnisse ("word", "excel" und "outlook") kopiert werden. Danach kann die Installation wie bei der Initialinstallation nochmals ausgeführt werden, oder es wird die eingebaute Update-Funktion (im Settings-Menü) benutzt, welche im Grunde dasselbe tut. Die Add-ins sind zudem so konfiguriert, dass sie alle drei Tage nach Updates prüfen (das kann über die Konfigurationsdatei verändert werden). Updates aus den Add-ins direkt setzen allerdings erstens voraus, dass der Update-Pfad in der Konfigurationsdatei hinterlegt ist ("UpdatePath = "). Zweitens muss dieser Update-Pfad derselbe sein, aus welchem die Add-ins ursprünglich installiert worden sind. Sonst muss der Benutzer sein bisheriges Add-in zuerst deinstallieren, bevor er die neue Version installieren kann.
- Die Updates sind bei laufendem Word, Excel und Outlook möglich, wirken aber erst beim nächsten Start der Anwendung.
- 262 Die Updatefunktion generiert eine lokale Log-Datei. Sie wird unter %AppData%\redink abgelegt (update.log) und kann im Falle von Fehlern bei der Lokalisierung der Fehlerquelle helfen.

- 263 Ob ein Update erfolgreich war, kann das Add-in bisher nicht selbst prüfen; es kann den Prozess nur anstoßen. Der Erfolg ist ggf. am Installer zu sehen, dessen Fenster erscheint und er kann manuell geprüft werden. Im Windows-Menü zum "Programme hinzufügen und entfernen" wird jedes Add-in angezeigt, zusammen mit einer Versionsnummer der Programmdateien (hier: 0.0.1.2). Nach einem Update muss diese Nummer höher sein:



- 264 Im Zweifel das Add-in deinstallieren und neu installieren. Die Konfiguration bleibt erhalten.
- 265 Die **Deinstallation** von Red Ink erfolgt über die vorstehend beschriebene Windows-Funktion (den Knopf "Uninstall" im obigen Screenshot drücken). Die Helper-Dateien können entweder über die Settings-Funktion oder manuell gelöscht und sind damit deinstalliert.
- 266 Wir empfehlen allen, sich zudem in den **Red-Ink-Mail-Verteiler** auf <https://vischer.com/redink> einzutragen, um bezüglich unsere Tools auf dem Laufenden zu bleiben.

### C. Vorbereitung: API-Zugang

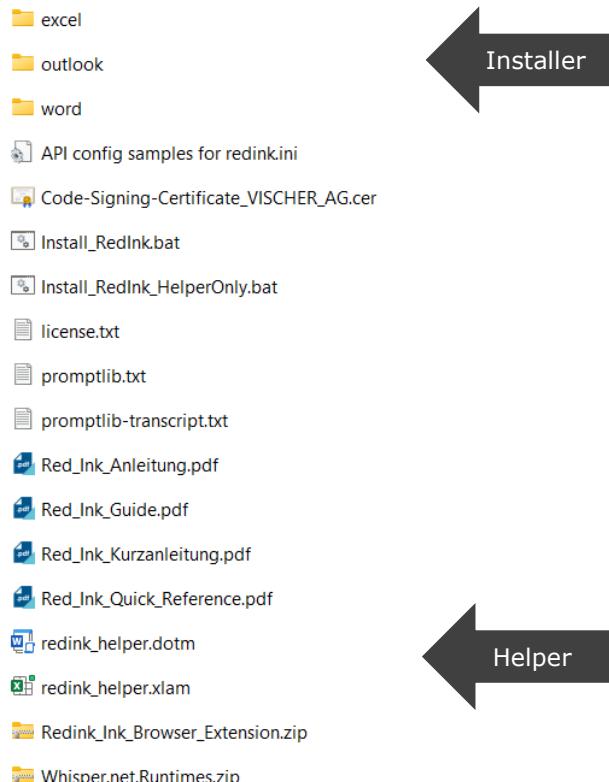
- 267 Damit Red Ink benutzt werden kann, braucht es einen Zugang zu einem geeigneten Sprachmodell einer neuen Generation (wie "gpt-5" von OpenAI oder Microsoft oder "Gemini 2.5 Pro" von Google). Ein normaler Zugang zu "ChatGPT" oder "Copilot" genügt nicht. Benötigt wird ein sog. API-Zugang. API steht für "Application Programming Interface" (zu Deutsch "Anwendungsprogrammierschnittstelle") und bedeutet hier eine über Internet oder im lokalen Netzwerk zugängliche Schnittstelle, an welche Red Ink (oder sonst eine Software) direkt An-

- fragen für das Sprachmodell senden kann. In der Fachsprache ist teilweise auch von "Endpoint" die Rede.
- 268 Einen API-Zugang zu einem Sprachmodell zu erhalten ist nicht schwer. Viele Unternehmen haben schon einen solchen für andere Anwendungen in Betrieb (z.B. über die "Azure OpenAI Services", wenn sie Microsofts Online-Services einsetzen) und können diesen auch für Red Ink nutzen. Wer keinen solchen API-Zugang hat, kann diesen für sehr wenig Geld zum Beispiel bei OpenAI (<https://openai.com/api/>) oder Google (<https://ai.google.dev/gemini-api/docs/api-key>) abonnieren. Wer ein "ChatGPT"-Konto hat, kann dies über dieses Konto machen (dies ist auch mit der kostenlosen Version von ChatGPT möglich, allerdings muss vor dem Generieren eines API-Key zwingend eine Kreditkarte hinterlegt und ein Betrag definiert werden, sonst gibt es später die Fehlermeldung "429"). Im Unterschied zu Services wie "ChatGPT" oder "Copilot für M365" wird die Nutzung eines Sprachmodells via API in der Regel nach Verbrauch bezahlt, wobei die Kosten für viele Benutzer erfahrungsgemäß tiefer sein werden als mit einem Abo für einen der Chat-Services. Wenn Sie mehr wissen wollen, bitten Sie einen guten KI-Chatbot um Hilfe.
- 269 Red Ink funktioniert grundsätzlich mit allen Sprachmodellen. Für rasche Antworten und komplexere Aufgaben sollte allerdings ein fortgeschrittenes Modell genutzt werden. Die leistungsfähigsten Modelle bieten Cloud-Anbieter wie Google, Microsoft, OpenAI, Anthropic oder Perplexity an. Es gibt daneben aber viele lokale Provider, die ebenfalls API-Zugänge anbieten und hierfür bekannte Open-Source-Modelle wie "Qwen3" und "Deepseek-R1" verwenden. Für manche Aufgaben genügen diese ebenfalls, auch wenn sie nicht an die Leistung der grossen Cloud-Modelle heranreichen. Lokale Anbieter können zudem aus der Perspektive des Datenschutzes bzw. des Geheimnisschutzes und weiterer Zusatzservices Vorteile mit sich bringen.
- 270 Beim Einsatz von Red Ink mit personenbezogenen Daten ist auf jeden Fall darauf zu achten, dass ein KI-Service gebucht wird, der nicht nur von einem seriösen Anbieter stammt, sondern auch über einen Auftragsbearbeitungsvertrag und – falls Daten in ein Land ohne angemessenen Datenschutz übermittelt werden – auch entsprechende Absicherungen bietet. Diese Voraussetzungen erfüllen normalerweise nur KI-Services für geschäftliche Nutzer. Wir empfehlen im beruflichen Umfeld keine für Verbraucher bestimmte KI-Services zu nutzen; diese bieten oft nicht die erforderliche Kontrolle über die eigenen Daten.
- 271 Wer mit Red Ink Daten bearbeiten will, die einem Berufs- oder Amtsgeheimnis unterliegen, sollte sich einen API-Zugang beschaffen, der die entsprechenden Anforderungen erfüllt. Wir als Kanzlei benutzen beispielsweise den Zugang über die Vertex API von Google mit einem speziellen Vertrag. Es gibt aber auch andere Anbieter. Wir haben eine Liste solcher Anbieter auf <https://vischer.com/redink-de> publiziert, zusammen mit weiterführenden Angaben.

- 272 Bei (Schweizer) Rechtsfragen diesbezüglich beraten wir gerne.
- 273 Wer einen API-Zugang hat, kann sich einen entsprechenden "API-Key" generieren lassen, d.h. eine geheime Zeichensequenz, die als Zugangsschlüssel dient und in Red Ink hinterlegt werden muss, damit die Abfragen des Tools vom API beantwortet werden. Bei Google Vertex und gewissen anderen Anbietern ist ein spezielles Authentifikationsverfahren nötig, wo noch weitere Angaben benötigt werden. Den API-Key oder die weiteren Angaben sollten für die erste Benutzung von Red Ink griffbereit sein.

#### D. **Schritt 1: Installer bzw. Installationspaket herunterladen**

- 274 Für **Methode 1** ist das Vorgehen oben (Rz. 252 ff.) beschrieben.
- 275 Für **Methode 2** kann von derselben Adresse <https://apps.vischer.com> das Installationspaket "redink.zip" heruntergeladen (ebenfalls in der jeweils neusten Version). Es wird als "Local Installation Package" bezeichnet. Es handelt sich um eine ZIP-Datei. Deren Inhalt muss in ein temporäres Verzeichnis (z.B. auf dem Desktop oder in ein neu geschaffenes Verzeichnis für "Red Ink") entpackt (siehe aber Rz. 279 unten). Es enthält unter anderem folgende Dateien:



- 276 In den drei Verzeichnissen befindet sich der Installer für die drei Add-ins. Diese braucht es im Minimum. Der Rest ist optional.
- 277 Im Installationspaket hat es ferner in der Regel:
- Diese Dokumentation (in Deutsch und Englisch)

- Eine Kurzanleitung (in Deutsch und Englisch)
- Das digitale Code-Signing-Zertifikat von VISCHER AG
- Ein Muster der Konfigurationsdatei (redink.ini)
- Ein Muster der Prompt-Bibliothek
- Ein Muster der Prompt-Bibliothek für den Transcriptor
- Die optionalen Batch-Dateien "Install\_RedInk.bat" und "Install\_RedInk\_HelperOnly.bat" zur einfacheren Installation
- Eine Datei mit der Angabe zur Lizenz (auch mit Angaben zu den Lizenzen der verwendeten Programm-Bibliotheken)
- Die beiden optionalen Red Ink Helper für Word und Excel
- Eine Datei mit Muster-Einträgen für die API-Konfiguration für OpenAI, Azure und Google Vertex
- Eine Datei mit Muster-Konfigurationen für diverse "Special Services"
- Die Lizenzbestimmungen

278 Mit der **Batch-Datei** (".bat") können die nachfolgenden Schritte 2 und 4 mit einem Klick ausgeführt werden. Sie kopiert die beiden Helper-Dateien wie in Schritt 4 angegeben und führt dann die drei Installer, wie in Schritt 2 angegeben, aus. Wer will, kann diese anklicken und hat damit, wenn alles funktioniert, Schritt 2 und 4 ausgeführt. Es wird jeweils zuerst gefragt, ob fortgesetzt werden soll. Kommt es zu einem Fehler, wird es angezeigt (z.B. falls Word noch läuft, während der Helper zu kopieren versucht wird, kann das u.U. blockieren). Allenfalls fragt Windows, ob den Batch-Dateien vertraut wird. Dies ist entsprechend zu bestätigen.

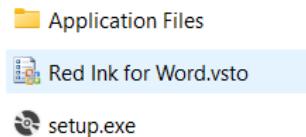
279 Damit spätere Updates ab künftigen Versionen des Installationspaketes möglich sind, muss die Installation immer aus demselben Dateipfad heraus erfolgen. Es lohnt sich daher, die Dateien aus dem Installationspaket in ein fixes, für Red Ink geschaffenes Verzeichnis auf dem lokalen Computer oder im Netzwerk des Betriebs abzuspeichern, von wo aus dann die Installation durchgeführt wird.

280 **Hinweis für Benutzer der Methode 1:** Auch sie können das Installationspaket herunterladen, um an die weiteren Dateien darin zu gelangen oder sie können diverse der Dateien (z.B. die Muster der Prompt Bibliothek) direkt von <https://apps.vischer.com> herunterladen.

## E. **Schritt 2: Installer ausführen**

281 Bei **Methode 1** wird der Installer bereits mit Schritt 1 mit der Bestätigung, dass Sie der Datei vertrauen, ausgeführt. Mehr ist hier nicht zu machen.

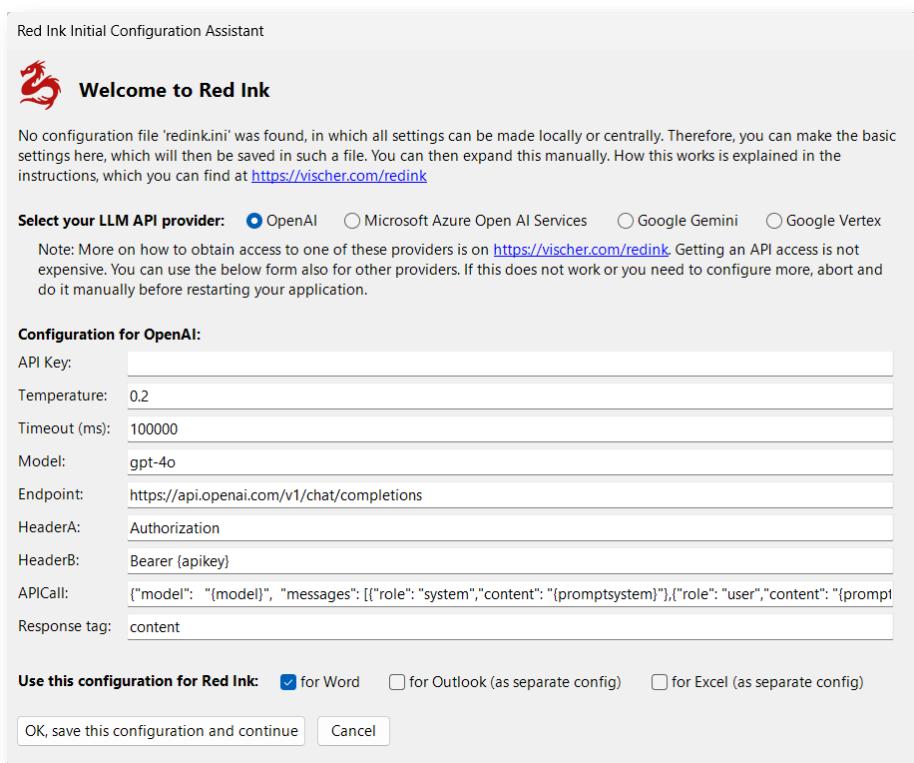
- 282 Bei **Methode 2** befindet sich in den Verzeichnissen "word", "excel" und "outlook" jeweils eine Datei mit der Endung ".vsto". Das ist das Add-in. Das sieht dann ungefähr so aus:



- 283 Zur Installation sollte die Datei ".vsto" ausgeführt werden ("setup.exe" geht auch, wird aber eher durch Sicherheitseinstellungen gesperrt). Die Installation sollte nach einigen Sekunden abgeschlossen sein. Es muss nichts eingegeben werden.
- 284 Für **beide Methoden** gilt: Sollte bei einer Anwendung das Add-in in einer früheren Version schon installiert sein, dann wird die Installation nicht funktionieren, falls diese von einer anderen Quelle oder auf eine andere Weise installiert worden ist. Es muss das frühere Add-in dann zuerst deinstalliert werden. Das geht in Windows mit "Programme hinzufügen oder entfernen" ebenfalls relativ einfach (vgl. Rz. 263 oben).
- 285 Die Konfigurationsdatei wird bei der Installation nicht überschrieben und bei der Deinstallation auch nicht entfernt.

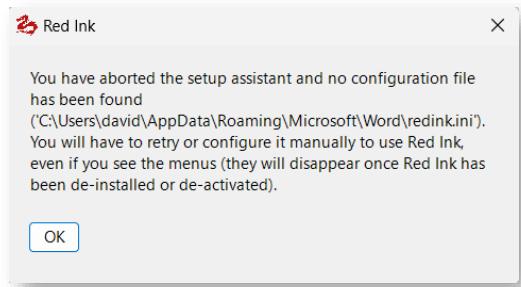
#### F. **Schritt 3: Erste Konfiguration mittels Assistent**

- 286 Wird Red Ink zum ersten Mal aufgestartet und liegt noch keine Konfigurationsdatei vor, erscheint folgendes Fenster, mit welcher eine Minimalkonfiguration erstellt werden kann (es kann in der aktuellen Version jeweils etwas anders aussehen; bei Word erscheint es erst, wenn ein Dokument geöffnet oder erstellt wird):



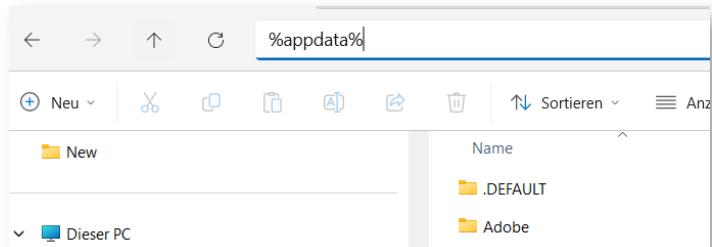
- 287 Zuerst ist einer der Provider auszuwählen. Es wurden die häufigsten Provider mit ihren typischen Angaben vorprogrammiert, soweit das allgemeingültig geht. Wir können diese Angaben aber nicht ständig nachführen, d.h. sie stimmen unter Umständen nicht mit den neusten Vorgaben überein. Wer hier nicht weiterkommt kann auch einen guten KI-Chatbot um Hilfe bitten. Die Nutzung von API-Zugängen ist an sich nicht für Endverbraucher gedacht und daher unter Umständen etwas kompliziert. Ist der eigene Provider hier nicht aufgeführt, können die Felder auch mit Angaben eines anderen Providers ausgefüllt werden (die Wahl über die Radio Buttons ist nur für die Vorauswahl der Werte relevant, sie wird nicht weiterverwendet). Der Inhalt der Parameter entspricht den Parametern der Konfigurationsdatei (diese sind in Rz. 309 ff. unten ausführlich beschrieben).
- 288 Wer beispielsweise einen API-Key von OpenAI hat, braucht diesen im optimalen Fall nur im betreffenden Feld einzugeben und kann loslegen.
- 289 Die Angaben müssen gespeichert werden, bevor sie benutzt werden können. Dabei wird eine lokale Konfigurationsdatei für Red Ink geschrieben. Es kann gewählt werden, ob sie nur für das aktuelle Add-in oder auch die Add-ins der beiden anderen Office-Produkte geschrieben werden soll.
- 290 Geschieht dies nicht bzw. wird abgebrochen, ist dies nicht weiter schlimm. Die Office-Anwendung kann trotzdem benutzt werden, aber der Assistent wird jedes Mal erscheinen, bis die Konfiguration gesetzt ist, die Konfigurationsdatei im relevanten Verzeichnis hinterlegt wird oder das Add-in deinstalliert (über "Programme entfernen ..." von

Windows) oder deaktiviert (innerhalb des Office-Produkts) worden ist. Nach dem Abbruch erscheint zudem folgende Fehlermeldung:



## G. Schritt 4: Helper installieren (Optional, kann später erfolgen)

- 291 Die Helper braucht es nicht unbedingt, aber wer das Kontextmenü in Word und Excel für Red Ink oder die Tastenkombinationen nutzen will, oder wer aus Excel gewisse Funktionen für eigene Excel-Anwendungen nutzen will (siehe Rz. 241 oben), braucht sie. Es handelt um digital signierte Programmdateien für Word und Excel, welche Makros in Visual Basic for Applications (VBA) enthält.
- 292 Es gibt **zwei Methoden**, die Helper zu installieren. Die einfachste Methode ist die Installation über die Settings-Funktion in Word und Excel. Dort erscheint ein Knopf "**Install Helper**". Wird er gedrückt, wird die aktuellste Helper-Version heruntergeladen und in das betreffende Verzeichnis kopiert. Der Nachteil dieser Methode ist, dass sie durch Sicherheitsfunktionen blockiert werden kann. Wenn dem so ist, muss der Helper manuell installiert werden. Dies ist nachfolgend beschrieben.
- 293 Für die manuelle Installation sind die Helper auch im Installationspaket enthalten. Zur Installation müssen sie lediglich in das passende Verzeichnis von Office kopiert werden:
- 294 Der Word-Helper "redink\_helper.dotm" gehört hierhin:  
C:\Users\vorname.name\AppData\Roaming\Microsoft\Word\Startup
- 295 Der Excel-Helper "redink\_helper.xlam" gehört hierhin:  
C:\Users\vorname.name\AppData\Roaming\Microsoft\Excel\XLSTART
- 296 Der rote Text "**vorname.name**" ist durch den eigenen Benutzernamen zu ersetzen. Am einfachsten wird das Verzeichnis angezeigt, wenn im Windows Explorer die Zeichenfolge "%appdata%" eingegeben und mit der Eingabetaste bestätigt wird:



Es erscheint automatisch ein Verzeichnis mit Zugang zu den Daten aller Anwendungen. Dort ist dann "Microsoft" und schliesslich Word oder Excel und die obigen Verzeichnisse "Startup" bzw. "XLSTART" zu wählen. Es kann sein, dass eines oder beide dieser Verzeichnisse fehlen; in diesem Fall sind sie einfach zu erstellen (es sind die Verzeichnisse, in welchem VBA-Dateien abgelegt werden, damit sie beim Start von Word bzw. Excel ausgeführt werden).

- 297 **Wer das nicht von Hand machen möchte**, kann die Batch-Datei "Install\_RedInk\_HelperOnly.bat" im Installationspaket ausführen. Sie kopiert die beiden Dateien an die richtige Stelle.
- 298 Die jeweilige Datei wird automatisch beim Start von Word und Excel geladen und aktiviert. Sie kann durch Löschen deinstalliert werden. Es kann allerdings sein, dass die Ausführung der Dateien durch die lokalen Sicherheitseinstellungen blockiert werden, obwohl die Daten von VISCHER digital signiert worden sind. In den meisten Betrieben wird das sogar normal sein. In diesem Fall müssen die Sicherheitseinstellungen um eine entsprechende Einzelfreigabe ergänzt werden (am einfachsten für Programmdateien, die von uns digital signiert sind). Allenfalls muss der Benutzer in Word und Excel auch angeben, dass der Inhalt "aktiviert" werden muss.
- 299 Die Installation der Helper kann auch später vorgenommen werden. Bis dahin wird allerdings das Kontextmenü nicht sichtbar sein. Die Add-ins prüfen jeweils, ob das entsprechende Helper-Programm läuft und aktivieren es dann automatisch, falls das Kontextmenü nicht über die Konfigurationsdatei global deaktiviert ist.

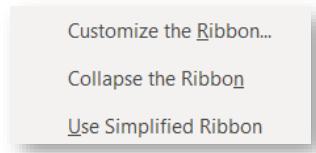
#### H. **Schritt 5: Add-ins benutzen**

- 300 Die Add-ins können nun benutzt werden. In Word und Excel sollten die Kacheln sofort sichtbar sein, wenn ein Dokument oder Arbeitsblatt geöffnet ist. In Outlook erscheinen die Kacheln nur, wenn ein Fenster zum Verfassen einer Mail (neue Mail, Antwort, Weiterleitung) offen ist.
- 301 Im **Anhang** hat es einige Vorschläge, wie Sie Red Ink besser kennenlernen können.
- 302 Hat es zu viele Kacheln auf dem Menüband ("Ribbon") gemessen an der Breite des Fensters, versuchen die Office-Anwendungen diese zu verdichten. Das sieht dann so aus:



- 303 Dies kann programmiertechnisch nicht verhindert werden. Jeder Benutzer kann aber nicht benötigte Kacheln durch Anpassen des Menübands entfernen und so Platz schaffen, damit es nicht zur Verdichtung

kommt (Zugang zu "Menüband anpassen..." über einen rechten Mausklick auf dem Menüband); dort kann auch die Position der Kacheln festgelegt werden (je weiter links sie zu liegen kommt, desto weniger wahrscheinlich wird sie verkleinert):



- 304 In Outlook platziert das Add-in seine Kacheln deshalb bewusst weit links. Wird Red Ink in einem Betrieb eingesetzt, kann es allerdings sein, dass beim nächsten Start der Anwendung die Position wieder zurückgesetzt ist, weil die globalen Einstellungen dies verlangen.

#### **I. Schritt 6: Bei Bedarf weitere Anpassungen vornehmen**

- 305 Red Ink kann auf sehr vielfältige Weise konfiguriert und angepasst werden. Das geht so weit, dass sogar die intern verwendeten Prompts verändert werden können. Ein Teil dieser Einstellungen ist über die Settings-Funktion verfügbar; dort sind allerdings nur jene Einstellungen, die "normale" Benutzer typischerweise benötigen. Auf die weiteren Einstellungen kann über die "Expert Config" zugegriffen werden oder noch einfach über die Konfigurationsdatei "redink.ini". Es handelt sich dabei um eine einfache Textdatei, in welcher alle Konfigurationen in Klartext abgelegt sind. Alles weitere ist im nächsten Kapitel im Detail beschrieben, auch der Speicherort der Konfigurationsdatei.

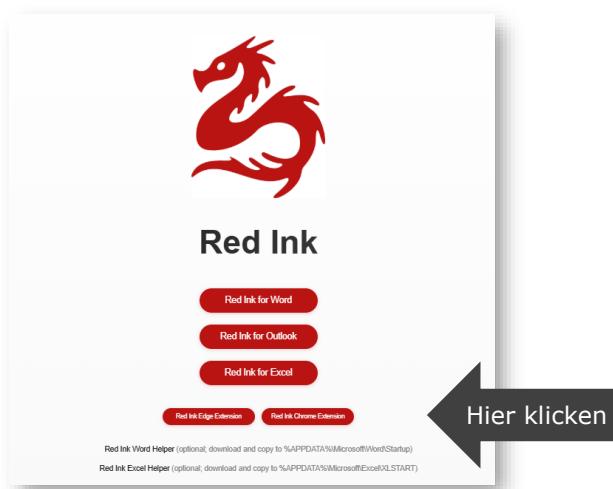
#### **J. Installation der Browser-Erweiterung**

- 306 Die Browser-Erweiterung wird von Chromium-basierten Browsern wie Edge oder Chrome unterstützt. Die Installation erfolgt entweder über den Add-on-Store oder manuell. Für die automatische Installation, Edge oder Chrome öffnen und zum jeweiligen Add-on-Store gehen:

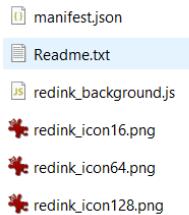
Edge: <https://microsoftedge.microsoft.com/addons/detail/red-ink-browser-extension/dflpmohocianaolidmcphfcpcognni>

Chrome: <https://chromewebstore.google.com/detail/red-ink-browser-extension/doagmfoemngdlbghobfkbobehbodgdoa>

oder <https://apps.vischer.com> aufrufen:



307 Für Letzteres sind die Dateien im Installationspaket enthalten:



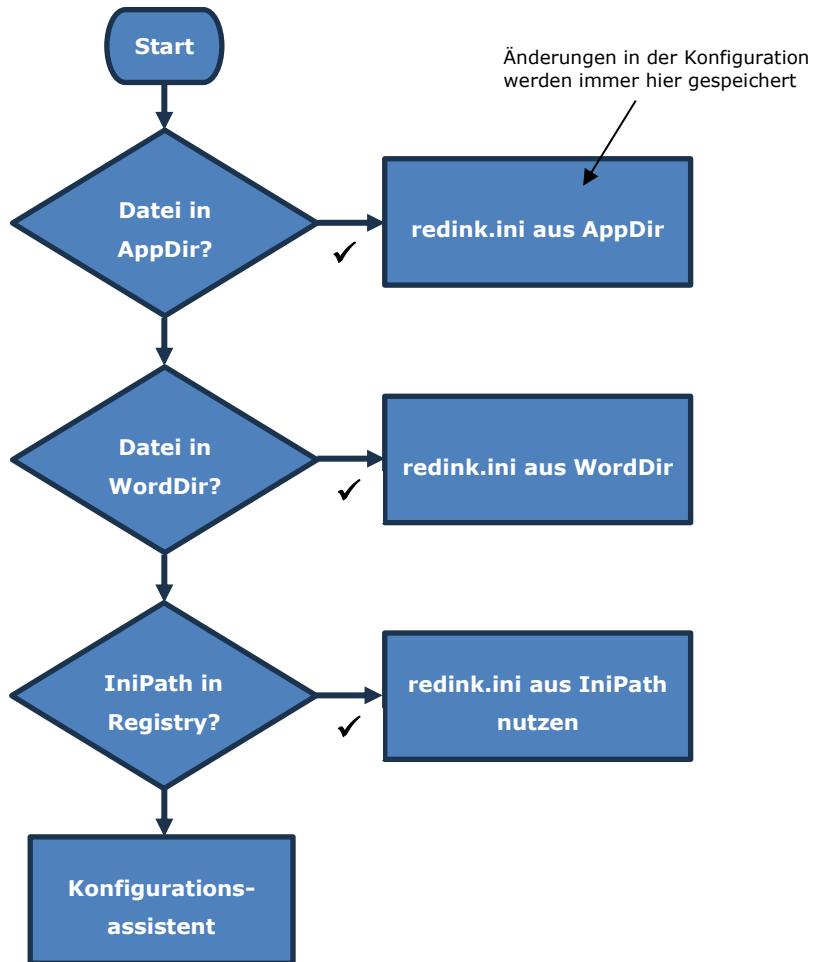
308 Sie müssen in ein dauerhaftes Verzeichnis kopiert werden (ausser Readme.txt) und können dann im Browser eingelesen werden, bei Edge z.B. auf der internen Seite "edge://extensions" (oder "chrome://extensions" bei Chrome), wobei hierfür der Entwickler-Modus aktiviert werden muss. Die Erweiterung kommuniziert über eine lokale http-Verbindung mit Red Ink, und zwar über die Ports 12333 und 12334 von IP-Adresse 127.0.0.1. In gewissen Unternehmen wird diese Kommunikation aus Sicherheitsgründen allerdings unterbunden sein.

#### **IV. KONFIGURATION (FÜR FORTGESCHRITTENE)**

##### **A. Konfigurationsdatei "redink.ini"**

309 Die Konfiguration von Red Ink erfolgt über die Parameter-Datei "redink.ini". Es handelt sich dabei um eine mit jedem Texteditor bearbeitbare Text-Datei, in welcher die Parameter für den Betrieb von Red Ink manuell angepasst werden können. Fehlt sie, startet das jeweilige Add-in einen Assistenten, mit welchem die minimalen Parameter erfasst werden, die es für die Benutzung braucht und wendet ansonsten die Standard-Parameter an (siehe Rz. 286 ff.). Es wird dann eine "redink.ini"-Datei automatisch generiert. Für eine spezifischere Konfiguration oder den Einsatz im Unternehmen empfiehlt es sich, die Konfigurationsdatei manuell vorzubereiten und vor dem Start der Add-ins in das betreffende Verzeichnis zu kopieren bzw. zentral in einem über das Registry allen Arbeitsstationen mitgeteilte Verzeichnis bereitzustellen. So kann Red Ink auch im Netzwerk verteilt werden.

- 310 Bezuglich des Standorts der Konfigurationsdatei ist Red Ink flexibel. Es gilt folgendes Konzept:



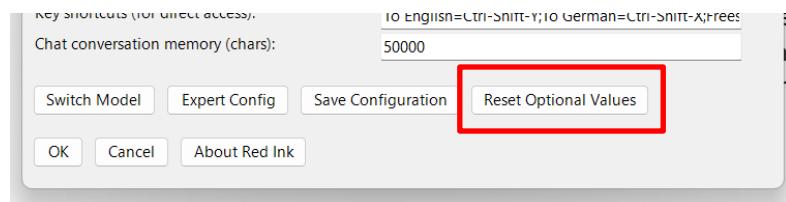
Dabei befinden sich die Verzeichnisse an folgenden Orten:

Verzeichnis	Ort
AppDir von Word	%AppData%\Microsoft\Word\
AppDir von Excel	%AppData%\Microsoft\Excel\
AppDir von Outlook	%AppData%\Microsoft\Outlook\
WordDir	Das AppDir von Word
IniPath	Der in der Registry unter dem Schlüssel "IniPath" hinterlegte Pfad (siehe dazu Rz. 320)

- 311 Die Priorität der Konfigurationsdatei im "lokalen" AppDir-Verzeichnis ist in den Add-ins fest einprogrammiert, kann im Quellcode über eine Konstante aber leicht geändert werden (erfordert ein erneutes Komplizieren der Anwendungen). In diesem Fall werden Änderungen trotzdem in das lokale AppDir geschrieben, aber nicht beachtet.
- 312 Die "redink.ini"- Datei wird bei jeder Benutzung eingelesen. Die verwendeten Parameter sind zwar über alle Add-ins dieselben, doch gewisse Parameter finden nur in einem oder zwei Add-ins Anwendung.

Sie stören die anderen Add-ins aber nicht. Durch separate Speicherorte für die Konfigurationsdatei ist es möglich, die drei Add-ins unterschiedlich zu konfigurieren. Sie können aber auch eine gemeinsame lokale Konfigurationsdatei teilen (im WordDir). Sobald aber in Outlook oder Excel eine Konfiguration abgespeichert wird (was jeder Benutzer kann), wird eine lokale, eigene Konfigurationsdatei abgelegt und benutzt.

- 313 Passt der Benutzer seine eigenen Konfigurationseinstellungen via Settings an und speichert er sie ab, erfolgt diese Speicherung in einer lokalen Kopie der Konfigurationsdatei, die gemäss obigem Schema prioritär eingelesen wird. Will er diese Änderungen wieder verwerfen und gewissermassen zur zentralen Konfigurationsdatei zurückkehren, kann er dies über einen Knopf unten im Settings-Dialog tun. Ist keine zentrale Konfigurationsdatei via Registry verfügbar, kann der Benutzer stattdessen die von ihm konfigurierten optionalen Einstellungen wieder auf die Standardwerte zurücksetzen (nicht tangiert werden die Zugangsdaten zur API):



- 314 Die Parameter der Konfigurationsdatei "redink.ini" sind wie folgt (der Key muss jeweils so wie angegeben geschrieben werden, inkl. Gross- und Kleinschreibung):

Key	Beispielwert	Bedeutung
APIKey	= sk-proj-XXXX	Der geheime API-Key für den Zugriff auf das LLM API, entweder im Klartext oder verschlüsselt; wird das OAuth 2.0-Verfahren benutzt, dann wird hier der Private Key des Service Accounts abgelegt, der für Red Ink benutzt wird (im Klartext oder verschlüsselt), und zwar der nackte Base64-Teil (d.h. ohne die Pre- und Suffix-Texte des PEM-Formats wie "---- BEGIN PRIVATE KEY----\n"), wobei Absatzeichen ("\\n") ignoriert bzw. vor der Verwendung herausgefiltert werden
APIKeyEncrypted	= Yes	"Yes", falls der API-Key (oder bei OAuth 2.0 der Private Key) verschlüsselt ist, sonst "No"; Zur Verschlüsselung siehe Abschnitt D
APIKeyPrefix	= sk-proj-	Der Prefix, der bei jedem API-Key des betreffenden Anbieters vorkommt (falls ein solcher verwendet wird; das dient der

			Sicherheit, weil der Prefix bei der Verschlüsselung nicht codiert wird). Bei der Verwendung von OAuth 2.0 wird der Prefix nicht beachtet
Endpoint	=	<a href="https://api.openai.com/v1/chat/completions">https://api.openai.com/v1/chat/completions</a>	Die URL, an die die API-Abfrage mittels POST-Methode gesendet wird; wenn https verwendet wird, wird die Übermittlung verschlüsselt; wird der Url ein "GET:" vorangestellt, erfolgt die Abfrage mittels GET-Methode; es können im Endpoint-Parameter auch Platzhalter verwendet werden, und es ist möglich, eine Doppelabfrage zu konfigurieren, d.h. dass zuerst ein POST-Befehl erfolgt und in einem zweiten Durchgang (mit Ergebnissen des ersten Befehls) eine zweite GET-Abfrage; die beiden Endpoints werden dann durch das Zeichen ":" getrennt; dies ist vor allem für Special Services relevant; Details zu dieser Konfiguration und der Verwendung von Platzhaltern sind in Rz. 340 ff. beschrieben
HeaderA	=	Authorization	Der erste Teil des http-Headers; der API-Key kann als Platzhalter vorgesehen werden (wie angegeben beim nächsten Beispiel angegeben); es ist dies der Feldname des Headers (ohne Doppelpunkt); das Feld darf leer sein
HeaderB	=	Bearer {apikey}	Der zweite Teil des http-Headers; der API-Key kann als Platzhalter vorgesehen werden (wie angegeben); es ist dies der Wert des Felds, das in HeaderA angegeben ist; der Parameter darf leer sein, falls HeaderA es auch ist
Response	=	content	Das JSON-Feld, mit welchem bei der API-Antwort die Antwort des LLM gekennzeichnet ist, die weiter verarbeitet werden soll vom Add-in; die Bezeichnung des Felds können noch folgende Parameter angehängt werden: (rkmode_all), (rkmode_longest) und (rkmode_first) – gibt an, wie beim Vorkommen mehrerer Felder mit demselben Namen umgegangen werden soll (es werden alle kombiniert, es wird der längste Inhalt oder es wird der erste Inhalt genommen; Standard ist der längste); (nothink), falls gesetzt, filtert das Add-in jeden Text vor dem Tag </think> aus der Antwort des LLM aus; das kann nützlich sein, wenn das LLM seine Denkschritte mit der Antwort ausgibt, diese aber stören. Bei Bedarf (insb. bei der Ver-

			wendung von "Special Service") sind anstelle einer einfachen Feld-Bezeichnung hier Templates für komplexere Auswertungen definiert werden; Details sind in Rz. 340 ff. beschrieben; erfordert ein Zugriff zwei Aufrufe von Endpoints, können hier zwei Antwort-Templates, getrennt mit ";", erfasst werden; dies muss dann mit dem Parameter "APICall" und "Endpoint" abgestimmt sein; wird "Response = JSON" angegeben, wird die Antwort des Modells als solche ausgegeben (für Debugging-Zwecke); unabhängig vom Parameter "Response" extrahiert Red Ink bei der Antwort des Sprachmodells binäre Bildobjekte (und speichert diese ab) und Quellenzitate (d.h. Hyperlinks, die dann angefügt werden)
APICall	=	{"model": "{model}", "messages": [{"role": "system", "content": "{promptsystem}"}, {"role": "user", "content": "{promptuser}"}], "temperature": {temperature}}	Die Syntax, in welchem die API des LLM die Abfrage übermittelt wird, wobei in geschwungenen Klammern die entsprechenden Platzhalter für den Systemprompt, den Benutzerprompt, das Modell und die Temperatur angegeben werden können; wird nicht zwischen System- und Benutzerprompt unterschieden, sollten beide hintereinander angegeben werden; neben {promptuser} und {promptsystem} kann auch der Platzhalter {userinstruction} benutzt werden, der den Prompt beinhaltet, den der Benutzer in Freestyle eingibt (was je nach Modell praktisch sein kann, weil er auf diese Weise zusätzlich im User-Prompt platziert werden kann, obwohl er normalerweise im System-Prompt ist; der Platzhalter {objectcall} dient schliesslich dazu, die Call-Sequenz für das Übermitteln von Dateiinhalten aufzunehmen (siehe "APICall_Object"); bei Bedarf (insb. bei der Verwendung von "Special Service") können hier auch zwei API-Call-Strings konfiguriert werden (getrennt mit " ", was dazu führt, dass zuerst ein POST-Befehl, dann ein GET-Befehl ausgeführt wird; Details sind in Rz. 340 ff. beschrieben; es müssen dann auch die Parameter "Endpoint" und "Response" erweitert werden)
APICall_Object	=	, {"inlineData": {"mimeType": "{mimetype}", "data": "{encodeddata}"}}	Die Sequenz, die an der richtigen Stelle im APICall integriert wird (dort, wo "APICall" den Platzhalter {objectcall} enthält) und dazu dient, dem Modell den Inhalt einer Datei zu übermit-

		oder multi-part:model:{model };prompt:{prompt system} {prompt user};filefield:image []	teln; der Platzhalter {mimetype} wird vom Add-in automatisch mit dem MIME-Typ ersetzt (z.B. "image/jpeg"), der Platzhalter {encodeddata} wird automatisch ausgefüllt mit dem Base64-Code der Datei; alternativ zur JSON-Codierung wird auch die Multipart-Codierung unterstützt; in diesem Fall wird APICall nicht gebraucht (es kann ein beliebiger Wert enthalten sein), und APICall_Object muss mit "multipart:" beginnen, danach folgen getrennt durch ; die Felder und nach einem Doppelpunkt ihre Platzhalter (wird ":" benötigt ist ";" das Escape Zeichen); nur wenn dieser Parameter definiert ist, steht in Freestyle der Trigger "(file)" zur Verfügung
Timeout	=	200000	Der Timeout für eine Anfrage bei der API in Millisekunden
Temperature	=	0.2	Die Temperatur, soweit sie angegeben werden kann (wird oben eingesetzt)
Model	=	gpt-4o	Der Modellname (wird oben eingesetzt, nötigenfalls auch im Endpoint, wenn dort ein Platzhalter ist)
MaxOutputToken	=	8192	Hier kann eingestellt werden, wieviele Tokens ein Output des Sprachmodells maximal haben kann; das Add-in wird in Word und Outlook den Benutzer warnen, wenn ein Text bearbeitet werden soll, der aus mutmasslich mehr als dieser Anzahl Tokens besteht, was zur Folge haben kann, dass der Output gekürzt wird vom Sprachmodell, was wiederum bedeuten kann, dass er nicht vollständig ist; die meisten Modelle können viel weniger langen Output generieren als sie Input aufnehmen können; bei 0 wird die Prüfung nicht vorgenommen
Anon	=	askshow; 4	Mit diesem Parameter kann die integrierte Anonymisierungsfunktion für das betreffende Modell konfiguriert werden; es ist der Modus (none, silent, ask, askshow, show) und der Typ (0-4) der Anonymisierung anzugeben (Details siehe Rz. 149 ff.); wird der Parameter nicht gesetzt, findet keine Anonymisierung statt; der Benutzer kann diesen Parameter mit einer lokalen Datei mit dem Namen redink-anon.txt auf seinem Desktop übersteuern
TokenCount	=	candidatesTokenCount; thoughtsTokenCount; 5; CHF	Angaben zur Freestyle-Funktion, welche die jeweils verbrauchten Tokens protokol-

			liert und mit einem Währungsbetrag multipliziert; es sind zuerst die JSON-Segmente aufzuzählen, welche die gewünschten Tokenwerte enthalten (eins oder mehrere), dann optional der Multiplikator und dann die Angabe der Währung oder einer anderen Bezeichnung, die dann in der Datei "redink-cost.txt" auf dem Desktop erscheint
OAuth2	=	Yes	"Yes", falls statt eines normalen API-Key das OAuth 2.0-Verfahren für die Authentisierung verwendet werden soll; in diesem Fall enthält der Key "APIKey" den Private Key des für Red Ink benutzten Service Accounts, mit welchem der nötige Access Token angefordert wird; es sind dann auch die weiteren Keys nachfolgend zu konfigurieren
OAuth2ClientMail	=	red-dink@earnest-racecars-212313-x4.iam.gserviceaccount.com	Der "client_email"-Parameter, der die E-Mail-Adresse des Service Accounts enthält, welcher für Red Ink benutzt wird und der dem Authentication-Server nach dem OAuth 2.0-Verfahren gesendet werden muss, um einen Access Token anzufordern
OAuth2Scopes	=	https://www.googleapis.com/auth/cloud-platform	Der "scopes"-Parameter, der dem Authentication-Server nach dem OAuth 2.0-Verfahren gesendet werden muss, um einen Access Token anzufordern; er gibt an, für welche Ressourcen der Access Token verlangt wird
OAuth2Endpoint	=	https://oauth2.googleapis.com/token	Die Adresse des OAuth 2.0-Servers, der die Authentisierung durchführt
OAuth2ATExpiry	=	3600	Die Lebensdauer eines Access Token in Sekunden (kurz vor Ablauf dieser Zeit wird ein neuer Access Token angefordert); wird nichts konfiguriert, wird der Wert 3600 angenommen
DoubleS	=	Yes	Falls das "scharfe S" standardmäßig mit einem Doppel-S ersetzt werden soll; Standard ist "No"
Clean	=	No	Falls gesetzt, werden unsichtbare Leerzeichen und doppelte Leerzeichen im Output des LLM automatisch entfernt; solche Codes können als Wasserzeichen benutzt werden – mit diesem Befehl werden sie dann entfernt
PreCorrection	=	As an additional instruction but only if you generate text in German language, replace any appearance of 'per-	Ein optionaler Befehl, der bei jeder API-Abfrage mitgegeben wird (auf Englisch), z.B. um standardmässige Korrekturen vornehmen zu können; dieser Befehl wird bei der ersten LLM-

		sonenbezogene Daten' with 'Personendaten'	Abfrage mitgegeben
PostCorrection	=	All references in the text provided to you for processing that refer to Vischer have to be in all-caps, like VISCHER.	Ein optionaler Befehl, der nach jeder API-Abfrage automatisch auf der Antwort durchgeführt wird; wird dieser Befehl eingegeben, kommt es zu doppelt sovielen Abfragen, was höhere Kosten zur Folge hat und mehr Zeit braucht; daher ist vorzugsweise "PreCorrection" zu verwenden; hier ist der komplette (System-)Prompt einzugeben, während der Text, um den es geht, der KI als (User-) Prompt eingefasst durch die beiden Tags "<TEXTTOPROCESS>" und "</TEXTTOPROCESS>" übergeben wird
APIDebug	=	False	Wenn auf True gesetzt, wird die vollständige Antwort der LLM (oder des Special Service) in eine JSON-Datei auf dem Desktop des Benutzers geschrieben (sie wird mit jeder Anfrage überschrieben); Standard ist False.
Language1	=	English	Dies ist die erste Standardsprache, in welcher die Übersetzungsfunktion von Red Ink im Direktzugriff zur Verfügung steht. Der Text sollte auf Englisch sein; ist nichts angegeben, ist dies "English"
Language2	=	German	Dies ist die zweite Standardsprache, in welcher die Übersetzungsfunktion von Red Ink im Direktzugriff zur Verfügung steht. Der Text sollte auf Englisch sein; ist nichts angegeben, ist dies "German"
ShortcutsWordExcel	=	To English=Ctrl-Shift-E;To German=Ctrl-Shift-D;Freestyle=Ctrl-Shift-P;Self-Compare Selection=Ctrl-Alt-C	Damit können für Word und Excel für die einzelnen Menüs Tasten-Shortcuts definiert werden, damit die Funktionen auch ohne Rechtsklick aufgerufen werden können (geht schneller); es muss im Parameter der Text des Menüs genauso angegeben werden, wie er erscheint (bei Freestyle in Word mit Angabe des Modells, wo dieses im Kontextmenu erscheint), danach die Tastenkombination; unterstützt werden Ctrl, Alt, Shift, alle Buchstaben, Zahlen, F-Tasten und diverse Navigations- und sonstige Tasten (aber nicht vom Zahlenblock); wer AltGR will, sollte Ctrl-Alt schreiben; die Kombination wird im Menü als Tooltip angezeigt; Achtung: Red Ink vermag gewisse voreingestellte Zuweisungen nicht zu überschreiben, d.h.

			sie funktionieren dann einfach nicht; ferner funktionieren die Tastenkombinationen nur, wenn der Helper für Word bzw. Excel läuft (siehe Rz. 291 ff. oben)
UpdateCheckInterval	=	-1	Die Anzahl Tage nach dem letzten Update, die die Add-ins warten sollen, um nach neuen Updates zu prüfen oder solche zu versuchen; eine "stille" Prüfung funktioniert nur, wenn eine Installation direkt über das Internet erfolgt ist (sog. ClickOnce-Installation = Methode 1, siehe Rz. 274); ist ein Update-Pfad für lokale Updates konfiguriert, wird auch an Updates erinnert, aber der Benutzer wird jedes Mal gefragt, ob ein Update versucht werden soll (die Online-Update-Funktion wird übersteuert, wenn ein solcher Pfad besteht); Standard ist 7; ist der Wert auf 0 gesetzt, wird nie geprüft, ist er auf -1 gesetzt, wird bei jedem Start geprüft (empfiehlt sich nur für ClickOnce-Installationen); ist kein Update-Pfad vorhanden, erfolgt keine Aufforderung zu lokalen Updates
UpdatePath	=	X:\Updates\RedInk\	Wer lokale Updates direkt aus den Add-ins ermöglichen will, muss hier den Pfad zur Ablage der Installationsdateien von Red Ink angeben, wie sie über das Installationspaket bezogen werden können (Rz. 274); es wird in diesem Verzeichnis der Unterordner "word", "excel" und "outlook" erwartet; die Add-ins müssen aus demselben Pfad installiert worden sein, da das Update sonst nicht funktioniert
AlternateModelPath	=	X:\Configuration\allmodels.ini	Hier kann optional eine Konfigurationsdatei mit den Angaben für weitere Sprachmodelle konfiguriert werden, die in Word beim Aufrufen von "Freestyle (2nd)" als Alternative zum sekundären Sprachmodell gewählt werden können; dieser Parameter muss den vollen Pfad inkl. Dateiname der Konfigurationsdatei dieser alternativen Sprachmodelle enthalten; ihr Format und Inhalt ist in Rz. 331 ff. beschrieben
SpecialServicePath	=	X:\Configuration\specialservices.ini	Hier können Dienstleistungen konfiguriert werden, die auf die in Word über den Menüpunkt Analyse zugegriffen werden kann; das können Rechtsinformationssysteme, aber auch spezialisierte KI-Modelle oder interne Vektordatenbanken sein; die Funktionsweise und Konfiguration ist in Rz. 68 ff.

			(beschrieben)
SpeechModelPath	=	X:\Speech\	Der Pfad, in welchem die Vosk- und Whisper-Speech-to-Text- Modelle abgelegt sind, falls der Transcriptor benutzt werden soll; die Vosk-Modelle sind über <a href="https://alphacepheli.com/vosk/models">https://alphacepheli.com/vosk/models</a> zu beziehen (als ZIP) und für Whisper unter <a href="https://huggingface.co/ggerganov/whisper.cpp/tree/main">https://huggingface.co/ggerganov/whisper.cpp/tree/main</a> ; Red Ink erwartet in diesem Verzeichnis für jedes Modell ein Unterverzeichnis mit dessen Namen, das mit "vosk-model" beginnt; der Wert ist optional; die Prompt-Bibliothek für Transkripte kann auch in diesem Verzeichnis abgelegt werden
LocalModelPath	=	X:\Models\	Der Pfad, an welchem von Red Ink benutzte Modelle lokal (bzw. in einem Netzwerklauwerk) abgelegt sind; dies kann z.B. das Embedding-Modell für "Context Search" oder ein Anonymisierungsmodell sein; die Modelle werden jeweils in einem von Red Ink festgelegten Unterverzeichnis abgelegt, bei "Context Search" heisst es z.B. "embed" (siehe dort)
TTSEndpoint	=	https://texttospeech.googleapis.com/v1/ https://api.openai.com/v1/audio/speech	Die Internet-Adresse des Text-to-Speech-API von Google und OpenAI, falls die Funktion Create Podcast oder Create Audio benutzt werden soll; ihre Funktion setzt bei Google und OpenAI voraus, dass das primäre oder sekundäre Modell für die Vertex API von Google bzw. OpenAI konfiguriert worden ist; sind sowohl Google als auch OpenAI vorhanden, sind die Adressen durch " " zu trennen
ContextMenu	=	Yes	Ob – bei verfügbaren Helpern – das Kontextmenü in Word und Excel angezeigt werden soll; Standard ist "Yes"
UsageRestrictions	=	You may use Red Ink for all kinds of data, including professional secrecy data.	Dies ist ein beliebiger Text, der im Add-in erscheint, wenn der Benutzer mit dem Mauszeiger über das Red Ink-Logo fährt; er kann benutzt werden, um den Benutzer auf Verwendungsbeschränkungen hinzuweisen
SecondAPI	=	Yes	"Yes" falls ein zweites Modell konfiguriert und via Freestyle zur Verfügung gestellt werden soll; dies ist optional; es kann dasselbe oder ein anderes API konfiguriert werden wie beim Haupt-Modell; wird kein zweites Modell benötigt, dann auf "No" setzen
APIKey_2	=	sk-proj-XXXX	Wie oben, für das zweite Modell

APIKeyEncrypted_2	=	Yes	Wie oben, für das zweite Modell
APIKeyPrefix_2	=	sk-proj-	Wie oben, für das zweite Modell
Endpoint_2	=	https://api.openai.com/v1/chat/completions	Wie oben, für das zweite Modell
HeaderA_2	=	Authorization	Wie oben, für das zweite Modell
HeaderB_2	=	Bearer {apikey}	Wie oben, für das zweite Modell
Response_2	=	content	Wie oben, für das zweite Modell
APICall_2	=	{"model": "{model}", "messages": [{"role": "user", "content": "{promptsystem}{promptuser}"}], "temperature": 1.0}	Wie oben, für das zweite Modell
APICall_Object_2	=	, {"inlineData": {"mimeType": "{mimetype}", "data": "{encodeddata}"}}	Wie oben, für das zweite Modell
Timeout_2	=	200000	Wie oben, für das zweite Modell
Temperature_2	=	1.0	Wie oben, für das zweite Modell
Model_2	=	o1-preview	Wie oben, für das zweite Modell
MaxOutputToken_2	=	8192	Wie oben, für das zweite Modell
Anon_2	=	none; 0	Wie oben, für das zweite Modell
TokenCount_2	=	candidatesTokenCount; thoughtsTokenCount; 5; CHF	Wie oben, für das zweite Modell
OAuth2_2	=	No	Wie oben, für das zweite Modell
OAuth2ClientMail_2	=	red-ink@earnest-racecars-212313-x4.iam.gserviceaccount.com	Wie oben, für das zweite Modell
OAuth2Scopes_2	=	https://www.googleapis.com/auth/cloud-platform	Wie oben, für das zweite Modell
OAuth2Endpoint_2	=	https://oauth2.googleapis.com/token	Wie oben, für das zweite Modell
OAuth2ATExpiry_2	=	50	Wie oben, für das zweite Modell
MarkupMethodHelper	=	2	Gibt an, welche Methode zur Erstellung eines Markups beim Einsatz des Word Helpers Self-Compare Selection verwendet werden soll: 1 = Compare-Funktion von Word; dabei ist es unvermeidlich, dass Fenster vorübergehend aufgehen, und andere Add-ins (wie z.B. iManage) stören den Vorgang, weil sie nicht konform programmiert sind 2 = einfacher Diff-Algorithmus; er ist bei kurzen Texten unproblematischer und schneller als Word, aber nicht so zuverlässig 3 = derselbe Diff-Algorithmus, allerdings wird das Ergebnis in

			<p>einem Fenster angezeigt, was die schnellste Methode ist Standard ist 3 Für weitere Ausführungen siehe Rz. 25 oben</p>
MarkupMethodWord	=	1	<p>Gibt an, welche Methode zur Erstellung eines Markups beim Einsatz der diversen vorprogrammierten Funktionen von Word verwendet werden soll (falls eins erstellt wird):          1 = Compare-Funktion von Word; dabei ist es unvermeidlich, dass Fenster vorübergehend aufgehen, und andere Add-ins (wie z.B. iManage) stören den Vorgang, weil sie nicht konform programmiert sind (= Standard)          2 = einfacher Diff-Algorithmus; er ist bei kurzen Texten unproblematischer und schneller als Word, aber nicht so zuverlässig          3 = derselbe Diff-Algorithmus, allerdings wird das Ergebnis in einem Fenster angezeigt, was die schnellste Methode ist          4 = ein LLM- und Regex-basiertes Verfahren, welches je nach LLM für grössere Texte als Diff funktioniert und vor allem Formatierungen belässt; es ist aber inhaltlich weniger zuverlässig          Standard ist 3          Für weitere Ausführungen siehe Rz. 25 oben</p>
MarkupMethodOutlook	=	2	<p>Gibt an, welche Methode zur Erstellung eines Markups beim Einsatz der diversen vorprogrammierten Funktionen von Outlook verwendet werden soll (falls eins erstellt wird):          1 = Compare-Funktion von Word; weil Outlook keine Überarbeitungsmarkierungen kennt, wird der Markup farblich angezeigt.          2 = einfacher Diff-Algorithmus; er ist bei kurzen Texten unproblematischer und schneller als Word, aber nicht so zuverlässig          3 = derselbe Diff-Algorithmus, allerdings wird das Ergebnis in einem Fenster angezeigt, was die schnellste Methode ist          Standard ist 3          Für weitere Ausführungen siehe Rz. 25 oben</p>
MarkupDiffCap	=	20000	<p>Gibt die maximale Textlänge an, für welche die Diff-Markup-Methode verwendet werden soll (falls nicht DiffW benutzt wird) und für die automatische Markdown-Konvertierung für selektierte Texte, weil er für sehr lange Texte nicht geeignet ist; es kann im Einzelfall aber entschieden werden, ihn trotzdem</p>

			zu benutzen (sonst DiffW); Standard ist 20'000; für weitere Ausführungen siehe Rz. 25 oben
MarkupRegexCap	=	30000	Gibt die maximale Textlänge an, für welche der Regex-Markup-Methode verwendet werden soll, weil er bei längeren Texten viel Zeit beanspruchen kann und weniger zuverlässig ist; es kann im Einzelfall aber entschieden werden, ihn trotzdem zu benutzen (sonst Word Compare); Standard ist 30'000; für weitere Ausführungen siehe Rz. 25 oben
MarkdownConvert	=	Yes	Gibt an, ob bei Texten in Word vorab diverse Formatierungen wie Fett, Kursiv oder Unterstrichen in Markdown konvertiert werden sollen, damit sie nachher wieder in Formatierungen umgewandelt werden können, da die meisten LLM dies unterstützen und Red Ink Markdown beim Einfügen von Text unterstützt; dies ist standardmäßig eingeschaltet, aber auf die Anzahl Zeichen von MarkupDiffCap begrenzt (weil es sonst zu viel Zeit braucht)
KeepFormat1	=	No	Gibt an, ob in Outlook und Word bei der Übersetzungsfunktion versucht werden soll, die grundlegenden Formatierungen (Zeichen, Aufzählungen) zu erhalten oder ob in reinem Text gearbeitet wird (weshalb Formatierungen verloren gehen können); wird die Formatierung erhalten, braucht die Verarbeitung deutlich mehr Zeit, weil die Formatierung in den Text codiert werden muss, die von der KI bearbeitet wird (und sie dafür mehr Zeit braucht); Standard ist "Nein"; für weitere Ausführungen siehe Rz. 25 oben
KeepFormat2	=	Yes	Gibt an, ob in Outlook und Word bei der Korrektur-, Verbesserungs- und der Kürzungsfunktion versucht werden soll, die grundlegenden Formatierungen (Zeichen, Aufzählungen) zu erhalten oder ob in reinem Text gearbeitet wird (weshalb Formatierungen verloren gehen können); wird die Formatierung erhalten, braucht die Verarbeitung deutlich mehr Zeit, weil die Formatierung in den Text codiert werden muss, die von der KI bearbeitet wird (und sie dafür mehr Zeit braucht); Standard ist "Nein"; für weitere Ausführungen siehe Rz. 25 oben
KeepParaFormatInLine	=	Yes	Gibt an, ob in Word beim Bearbeiten von selektiertem Text

			versucht werden soll, die Formatierungen der Absätze des Original-Textes in den Text der Absätze einzucodieren, damit das Add-in versuchen kann, mindestens die Absatzformatierungen beim Ergebnis der KI wiederherzustellen; dies geht weniger weit als KeepFormat, braucht aber auch weniger Zeit und Daten; auch ohne diese Einstellung wird das Add-in versuchen, sich die Absatzformatierungen wo es passt zu merken; Standard ist "No"; für weitere Ausführungen siehe Rz. 25 oben
KeepFormatCap	=	5000	Die Funktionen KeepFormat und KeepParaFormatInLine können auf diese Weise bei längeren Texten automatisch ausgeschaltet werden, damit nicht zu viel Zeit darauf aufgewendet wird und Red Ink deswegen hängen bleibt; angegeben wird die Zeichenzahl, über welcher die automatische Abschaltung erfolgt (bei einem Wert von 0 erfolgt keine Prüfung, bei einem Wert von 1 wird immer auf die Speicherung des Formats im Text selbst verzichtet); Standard ist 5000; für weitere Ausführungen siehe Rz. 25 oben
ReplaceText1	=	Yes	Gibt an, ob bei der Übersetzungsfunktion in Outlook und Word der zu übersetzende Text durch die Übersetzung ersetzt werden soll oder ob die Übersetzung danach eingefügt wird; Standard ist "Ja"; für weitere Ausführungen siehe Rz. 25 oben
ReplaceText2	=	No	Gibt bei einigen der anderen, vorprogrammierten Funktionen (wie z.B. Korrekturen, Kürzungen) in Outlook und Word an, ob der angewählte Text durch die Antwort der KI ersetzt werden soll oder ob die Antwort danach eingefügt wird; Standard ist "Nein"; für weitere Ausführungen siehe Rz. 25 oben
DoMarkupWord	=	Yes	Gibt an, ob in Word bei gewissen der vordefinierten Funktionen (wie z.B. Korrekturen, Kürzungen) automatisch auch ein Markup des geänderten Textes angezeigt werden soll (was eine gewisse Zeit dauern kann); Standard ist "Ja"; für weitere Ausführungen siehe Rz. 25 oben
DoMarkupOutlook	=	Yes	Gibt an, ob in Outlook bei gewissen der vordefinierten Funktionen (wie z.B. Korrekturen, Kürzungen) automatisch auch ein Markup des geänderten Textes angezeigt werden soll (was

			eine gewisse Zeit dauern kann); Standard ist "Ja"; für weitere Ausführungen siehe Rz. 25 oben
PromptLib	=	%APPDATA%\Microsoft\Word\promptlib.txt	Falls der Wert angegeben wird, wird das Add-in bei Verwendung der Freestyle-Funktion aus der angegebenen Datei die darin gespeicherten Prompts laden und sie dem Benutzer zur Auswahl anbieten, falls er keinen Prompt angibt (zum Format der Datei siehe unten); es muss der Dateipfad mit Name der Datei angegeben werden (sie muss im txt-Format sein); für weitere Ausführungen siehe Rz. 324 unten
PromptLib_Transcript	=	X:\Speech\promptlib_transcript.txt	Der Pfad zur Prompt-Bibliothek für die Process-Funktion im Transcriptor; die Text-Datei folgt derselben Syntax wie die "normale" Prompt Bibliothek auf der vorstehenden Zeile; zum Transcriptor siehe Rz. 83 ff. oben
MyStylePath	=	%APPDATA%\Microsoft\Word\mystyle.txt	Falls der Wert angegeben wird, steht in Word und Outlook die MyStyle-Funktion zur Verfügung; für weitere Ausführungen siehe Rz. 48 ff. und Rz. 324 unten
DocCheckPath	=	N:\AI\Library	Falls der Wert angegeben wird, wird die Funktion Document Check versuchen, hier Dateien mit der Signatur "redink-dc-* .txt" zu lesen und Rule Sets extrahieren (dieser Wert kann für die zentrale Speicherung in einem Netzwerk benutzt werden)
DocCheckPathLocal	=	%APPDATA%\Microsoft\Word	Falls der Wert angegeben wird, wird die Funktion Document Check versuchen, hier Dateien mit der Signatur "redink-dc-* .txt" zu lesen und Rule Sets extrahieren (dieser Wert kann für die lokale Speicherung benutzt werden, wenn nur der Benutzer das Rule Set nutzen können soll)
FindClausePath	=	N:\AI\Library	Falls der Wert angegeben wird, wird die Funktion Find Clause versuchen, hier Dateien mit der Signatur "redink-dc-* .txt" zu lesen und eine darin enthaltene Klauseldatenbank zu extrahieren (dieser Wert kann für die zentrale Speicherung in einem Netzwerk benutzt werden)
FindClausePathLocal	=	%APPDATA%\Microsoft\Word	Falls der Wert angegeben wird, wird die Funktion Find Clause versuchen, hier Dateien mit der Signatur "redink-lib-* .txt" zu lesen und eine darin enthaltene Klauseldatenbank zu extrahieren (dieser Wert kann für die

			lokale Speicherung benutzt werden, wenn nur der Benutzer das Rule Set nutzen können soll)
WebAgentPath	=	N:\AI\Library	Falls der Wert angegeben wird, wird die Funktion WebAgent versuchen, hier Dateien mit der Signatur "redink-ag-* .json" zu lesen und ihr benötigtes Script zu extrahieren (dieser Wert kann für die zentrale Speicherung in einem Netzwerk benutzt werden)
WebAgentPathLocal	=	%AP-PDATA%\Microsoft\Word	Falls der Wert angegeben wird, wird die Funktion WebAgent versuchen, hier Dateien mit der Signatur "redink-ag-* .json" zu lesen und ihr benötigtes Script zu extrahieren (dieser Wert kann für die lokale Speicherung benutzt werden, wenn nur der Benutzer das Rule Set nutzen können soll)
ISearch	=	Yes	"Yes", falls in der Funktion Freestyle zusätzlich eine Internet-Abfrage möglich sein soll; Standard ist "Ja"
ISearch_URL	=	https://duckduckgo.com/html/?q=	Die URL, die für die Internet-Abfrage benutzt werden soll; Standard ist DuckDuckGo
ISearch_ResponseMask1	=	duckduck-go.com/l/?uddg=	Dieser Wert gibt an, welche Zeichen sich bei den gefundenen Treffern im HTML-Code der Suchmaschinentrefferseite links von der URL des Treffers finden (der Wert wird benutzt, um die Treffer zu identifizieren); der Standard ist wie links angegeben
ISearch_ResponseMask2	=	&	Dieser Wert gibt an, welche Zeichen sich bei den gefundenen Treffern im HTML-Code der Suchmaschinentrefferseite rechts von der URL des Treffers finden (der Wert wird benutzt, um die Treffer zu identifizieren); der Standard ist wie links angegeben
ISearch_Name	=	DuckDuckGo	Der Name der Suchmaschine (wird teilweise dem Benutzer angezeigt); der Standard ist wie links angegeben
ISearch_Tries	=	10	Wieviele Suchtreffer sich die KI (von oben nach unten) anschauen soll, wenn die Internet-Funktion benutzt wird, bis sie die unten definierte Anzahl an Inhalten hat; Standard ist 10, Maximum ist 30
ISearch_MaxDepth	=	2	Wie tief das Add-in in eine Website, die als Treffer besucht wird, hineingehen soll, weil bei vielen der komplexeren Websites die Informationen sich erst auf Unterseiten finden; Stan-

			dard ist 2, Maximum ist 10
ISearch_Timeout	= 3		Wie lange das Add-in sich einem Suchtreffer widmen soll (in Sekunden); Standard ist 3, Maximum ist 60
ISearch_Results	= 2		Nach wievieleen Suchtreffern, die Inhalte geliefert haben, soll die Suche aufhören; Standard ist 4, Maximum ist 15
ISearch_Approve	= No		Bevor etwas an die Suchmaschine übergeben wird, kann das Add-in fragen, ob es das auch tun darf (und die Suchanfrage anzeigen); dies kann der Wahrung der Vertraulichkeit dienen; Standard ist "Nein"
ISearch_SearchTerm_SP	= You are a ...		Prompt, der benutzt wird, um die passenden Internet-Suchbegriffe zu ermitteln; es können Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion {CurrentDate} = aktuelles Datum
ISearch_Apply_SP	= You are a ...		Prompt, der benutzt wird, um den Freestyle-Befehl mit den Internet-Suchergebnissen umzusetzen; es können Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion {SearchResult} = Suchergebnis
ISearch_Apply_SP_Markup	= You are a ...		Prompt, der benutzt wird, um den Freestyle-Befehl mit den Internet-Suchergebnissen umzusetzen, wenn ein Text selektiert worden ist; es können Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion {SearchResult} = Suchergebnis
Lib	= Yes		"Yes" falls die Bibliothekssuche aktiviert sein soll; Standard ist "Nein"
Lib_File	= C:\users\username\Documents\library.txt		Der Dateiname mit vollem Pfad, in dem sich die Bibliotheksdatei befindet (im Format txt, doc, docx, oder rtf)
Lib_Timeout	= 30000		Timeout des LLM, wo dieses für die Bibliotheksfunktion benutzt wird (in Millisekunden); Standard ist 60000
Lib_Find_SP	= You are a ...		Prompt, der benutzt wird, um für die Bibliotheksfunktion aus der Bibliothek die relevanten Informationen zu extrahieren; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion {LibraryText} = Inhalt der Bibliothek
Lib_Apply_SP	= You are a ...		Prompt um die gefundenen In-

			halte auf die Anfrage des Benutzers anzuwenden; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion
Lib_Apply_SP_Markup	=	You are a ...	Prompt um die gefundenen Inhalte auf die Anfrage des Benutzers auf einen bestehenden Text als Markup anzuwenden; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion
LicensedTill	=	31.12.2025	Hier kann das Enddatum der eigenen Lizenz für Red Ink hinterlegt werden; danach wird Red Ink mit dieser Einstellung nicht mehr funktionieren (wobei ab sieben Tagen vorher eine Erinnerung erscheint); wird nichts angegeben, wird die im Code hinterlegte maximale Lebensdauer der Anwendung als Standardwert genommen; das Datum wird im "About Red Ink"-Fenster, wie es unter Settings aufgerufen werden kann, angezeigt; das Tool kontrolliert nicht, ob das Datum plausibel ist – der Parameter dient somit der eigenen Erinnerung
SP_Translate	=	You are a ...	Prompt für Übersetzungen; es kann folgender Platzhalter verwendet werden (siehe auch unten): {TranslateLanguage} = Sprache
SP_Correct	=	You are a ...	Prompt für Korrekturen (siehe auch unten)
SP_Improve	=	You are a ...	Prompt für sprachliche Verbesserungen (siehe auch unten)
SP_MyStyle_Apply	=	You are a ...	Prompt für sprachliche Anpassung basierend auf einem MyStyle Prompt (Rz. 48 ff.)
SP_NoFillers	=	You are a ...	Prompt für das Entfernen von Füllwörtern und Redundanzen (siehe auch unten)
SP_Convincing	=	You are a ...	Prompt für überzeugendere Formulierungen (siehe auch unten)
SP_Friendly	=	You are a ...	Prompt für freundlichere Formulierungen (siehe auch unten)
SP_Shorten	=	You are a ...	Prompt für Kürzungen; es kann folgender Platzhalter verwendet werden (siehe auch unten): {ShortenLength} = Outputlänge in Worten
SP_Summarize	=	You are a ...	Prompt für Zusammenfassungen; es kann folgender Platzhalter verwendet werden (siehe auch unten): {SummaryLength} = Outputlänge in Worten

SP_Explain	=	You are a ...	Prompt zur Analyse eines Textes (siehe auch unten)
SP_SuggestTitles	=	You are a ...	Prompt für Vorschläge verschiedener Titel (siehe auch unten)
SP_SwitchParty	=	You are a ...	Prompt für die Switch Party Funktion; es können folgender Platzhalter verwendet werden (siehe auch unten): {OldParty} = bisherige Partei {NewParty} = neue Partei
SP_Anonymize	=	You are a ...	Prompt zum Anonymisieren (siehe auch unten)
SP_WriteNeatly	=	You are a ...	Prompt für Vervollständigungen in Excel; es kann folgender Platzhalter verwendet werden (siehe auch unten): {Context} = erfasster Kontext
SP_FreestyleText	=	You are a ...	Prompt für die Freestyle-Funktion falls Text selektiert wurde (ohne die Zusätze, siehe unten); es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion
SP_FreestyleNoText	=	You are a ...	Prompt für die Freestyle-Funktion falls kein Text selektiert wurde (ohne die Zusätze, siehe unten); es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion
SP_ContextSearch	=	You are a ...	Prompt für die Context-Search-Funktion; es kann folgender Platzhalter verwendet werden (siehe auch unten): {SearchContext} = Kontext
SP_ContextSearchMulti	=	You are a ...	Prompt für die Context-Search-Funktion für den Fall, dass alle Treffer in einem Textteil gefunden werden sollen; es kann folgender Platzhalter verwendet werden (siehe auch unten): {SearchContext} = Kontext
SP_RangeOfCells	=	You are a ...	Prompt für die Range-Funktion in Excel; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion
SP_ParseFile	=	You are a ...	Prompt für den CSV-Analyzer in Excel; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Instruktion {Separator} = CSV Trennzeichen
SP_MailReply	=	You are a ...	Prompt für das Verfassen einer E-Mail-Antwort in Outlook; es kann folgender Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = Hinweise
SP_MailSumup	=	You are a ...	Prompt für das Zusammenfassen einer Mailkette (siehe auch unten)

			unten)
SP_MailSumup2	=	You are a ...	Prompt für das Zusammenfassen von mehreren selektierten E-Mails (siehe auch unten)
SP_Add_KeepFormulasIn tact	=	Beware, the text contains ...	Zusatzprompt, mit dem das Sprachmodell angewiesen wird, in Excel in den Zellen enthaltene Formeln beizubehalten
SP_Add-KeepHTMLIntact	=	When completing your task, leave any HTML tags ...	Zusatzprompt, mit dem das Sprachmodell angewiesen wird, HTML-codes nicht zu entfernen (sie werden für die Speicherung von Formatierungen gebraucht)
SP_Add_KeepInlineIntact	=	Do not remove any coding with ...	Zusatzprompt, mit dem das Sprachmodell angewiesen wird, andere im Text eincodierte Formatierungscodes nicht zu entfernen (sie werden für die Speicherung von Formatierungen gebraucht)
SP_Add_Bubbles	=	Provide your response to ...	Zusatzprompt, mit dem das Sprachmodell für Freestyle in Word angewiesen wird, die Ausgabe so zu formatieren, dass sie in Kommentare zum Text eingefügt werden können; das Add-in geht diese dann einzeln durch und setzt die Kommentare
SP_Add_Slides	=	You shall provide your output ...	Zusatzprompt, mit dem das Sprachmodell für Freestyle in Word angewiesen wird, die Ausgabe in Form von zusätzlichen Seiten für eine Powerpoint-Datei zu formatieren; sie liefert die Informationen für die Erstellung der Folienseiten in Form eines JSON-Strings, den Red Ink dann umsetzen kann
SP_Add_Batch	=	The main content ...	Zusatzprompt, mit dem das Sprachmodell für Freestyle in Excel angewiesen wird, den Inhalt einer Datei (die dem Modell jeweils einzeln übergeben wird) zu analysieren und das Ergebnis in einer bestimmten Zeile ("{LineNumber}") einzutragen
SP_MarkupRegex	=	You are an expert text comparison ...	Zusatzprompt, mit dem das Sprachmodell für Freestyle in Word angewiesen wird, zwei Texte zu vergleichen und die Differenzen als Regex-Suchmuster auszuweisen, damit in Freestyle von Word die Markup-Methode "Regex" umgesetzt werden kann
SP_Revisions	=	When you are asked to ...	Zusatzprompt, mit dem dem Sprachmodell für Freestyle in Word erklärt wird, wie es Markups in einem Text erkennen kann, falls die betreffende Funktion aktiviert worden ist ("(rev)").
SP_ChatWord	=	You are a helpful ...	Basisprompt für den Chatbot

			innerhalb von Word
SP_Add_ChatWord_Commands	=	You help the user ...	Zusatzprompt, um dem Chatbot zu erklären, wie er welche Kommandos zur Gestaltung von Texten anwendet
SP_ChatExcel	=	You are a helpful ...	Basisprompt für den Chatbot innerhalb von Excel
SP_Add_ChatExcel_Commands	=	You help the user ...	Zusatzprompt, um dem Chatbot in Excel zu erklären, wie er welche Kommandos zur Gestaltung von Arbeitsblättern anwendet
SP_Chat	=	You are an AI assistant ...	System-Prompt des Red Ink Local Chat (der via Browser benutzt werden kann, wenn Outlook läuft)
SP_Podcast	=	You are a professional podcaster ...	Prompt zur Erstellung von Podcast-Scripten basierend auf einem Text; es können folgende Platzhalter verwendet werden (siehe auch unten): {Language} = Sprache {TargetAudience} = Zielpublikum {DialogueContext} = Kontext {Duration} = Dauer {ExtraInstructions} = zusätzliche Vorgaben/Prompts
SP_MyStyle_Word	=	Read and deeply analyze all sample documents ...	Prompt zur Schreibstil-Analyse innerhalb von Word; es können folgende Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = weitere Instruktionen Dokumente werden zwischen den Tags <DOCUMENTnn>...</DOCUMENTnn> übergeben. Siehe auch Rz. 48 ff.
SP_MyStyle_Outlook	=	Read and deeply analyze all Outlook mails ...	Prompt zur Schreibstil-Analyse innerhalb von Outlook; es können folgende Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = weitere Instruktionen {Username} = Name des Benutzers Mails werden zwischen den Tags <MAILnn>...</MAILnn> übergeben. Siehe auch Rz. 48 ff.
ChatCap	=	50'000	Wieviele Zeichen aus dem bisherigen Dialog dem Chatbot bei jeder neuen Frage mitgegeben wird (zusätzlich zur Frage, den Basis- und Zusatz-prompts und dem aktuellen Text); Standardwert ist 50'000
SP_MergePrompt	=	You will be provided a text to insert into another text ...	Der Standard-Prompt, der in Word für das LLM-basierte Zusammenführen des in einer Pane selektierten Textes und dem selektierten Text im aktuellen Dokument von Word benutzt wird; er wird dem Benutzer in einem Eingabefenster angezeigt

			und kann von ihm verändert werden; es genügt, auf den einzufügenden und den anderen Text zu referenzieren; der Zusatzprompt SP_Add_MergePrompt liefert dem Modell dann die konkreten Hinweise
SP_MergePrompt2	=	You will be provided a text to insert into another text ...	Dito, für "Apply comment"
SP_Add_MergePrompt	=	The text to insert or merge ...	Der Prompt-Zusatz, der jeweils an den Prompt für das Zusammenführen von Texten angehängt wird, damit alles funktioniert. Er weist das Modell darauf hin, dass der Text in der Pane von Red Ink zwischen den Tags "<INSERT>" und "</INSERT>" gesendet wird, und der Text im Dokument zwischen den Tags "<TEXTTOPROCESS>" und "</TEXTTOPROCESS>"; der Benutzer muss dies also nicht angeben
SP_InsertClipboard	=	You will receive a binary object ...	Der Prompt, der beim Word-Helper "Clipboard to Text" benutzt wird, um den Inhalt der Zwischenablage in Text zu konvertieren
SP_DocCheck_Clause	=	You are to review ...	Der Prompt, der bei DocCheck in Word benutzt wird, um eine markierte Textstelle in einem Zug gegen alle Kriterien im gewählten Rule Set zu prüfen; das Rule Set wird zwischen den Tags "<RULESET>" und "</RULESET>" geliefert, weitere Dokumente zwischen "<DOCUMENTnn>" und "</DOCUMENTnn>" und der zu prüfende Text (d.h. was selektiert wurde) zwischen den Tags "<TEXTTOANALYZE>" und "</TEXTTOANALYZE>"; es können folgende Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = weitere Instruktionen Dieser Prompt kann im jeweiligen Rule Set durch einen eigenen Prompt ersetzt werden; wird er dort auf X gesetzt, steht die Funktion für das betreffende RuleSet nicht zur Verfügung
SP_DocCheck_MultiClause	=	You are to review ...	Der Prompt, der bei DocCheck in Word benutzt wird, um eine markierte Textstelle gegen ein einzelnes Kriterien-Set aus dem gewählten Rule Set zu prüfen; das Kriterium aus dem Rule Set wird zwischen den Tags "<RULESET>" und "</RULESET>" geliefert, weitere Dokumente zwischen "<DOCUMENTnn>" und

			"</DOCUMENTnn>" und der zu prüfende Text (d.h. was selektiert wurde) zwischen den Tags "<TEXTTOANALYZE>" und "</TEXTTOANALYZE>"; es können folgende Platzhalter verwendet werden (siehe auch unten): {OtherPrompt} = weitere Instruktionen Dieser Prompt kann im jeweiligen Rule Set durch einen eigenen Prompt ersetzt werden; wird er dort auf X gesetzt, steht die Funktion für das betreffende RuleSet nicht zur Verfügung
SP_DocCheck_MultiClaus_Sum	=	You are a well ...	Der Prompt, der benutzt wird, um die Ergebnisse der Prüfung bei der schrittweisen Prüfung eines Textes zu einem Bericht zusammenzufügen; dieser Prompt kann im jeweiligen Rule Set durch einen eigenen Prompt ersetzt werden
SP_DocCheck_MultiClaus_Sum_Bubbles	=	You are a well ...	Der Prompt, der benutzt wird, um die Ergebnisse der Prüfung bei der schrittweisen Prüfung eines Textes bei Verwendung von Word-Kommentaren zusammenzufassen; die Zusammenfassung wird dann am Anfang des selektierten Textes als separater Kommentar angezeigt; dieser Prompt kann im jeweiligen Rule Set durch einen eigenen Prompt ersetzt werden
SP_FindClause	=	Act as a clause finder ...	Der Prompt, der benutzt wird, um eine Klauseldatenbank bei Verwendung der Funktion Find Clause nach passenden Klauseln abzusuchen; die Datenbank wird (als JSON-Struktur) zwischen den Tags "<LIBRARY>" und "</LIBRARY>" angehängt, ein etwaiger Suchbegriff zwischen den Tags "<SEARCHQUERY>" und "</SEARCHQUERY>" sowie ein Text als Suchkontext zwischen den Tags "<TEXTFORSEARCH>" und "</TEXTFORSEARCH>"; dieser Prompt kann in der jeweiligen Klauseldatenbank durch einen eigenen Prompt ersetzt werden
SP_FindClause_Clean	=	You are a careful copy editor ...	Der Prompt, der benutzt wird, um einen selektierten Text, der einer Klauseldatenbank hinzugefügt werden soll ("Add Clause") vorher zu anonymisieren und sonst zu bereinigen

- 315 Bei den **Schalterwerten** sind ein "Ja", "Yes" und "True" untereinander gleichwertig, ein "Nein", "No" und "False" untereinander ebenfalls.
- 316 Die **Prompts**, die Red Ink für seine grundlegenden Funktionen benötigt, müssen normalerweise nicht konfiguriert werden. Sie können über

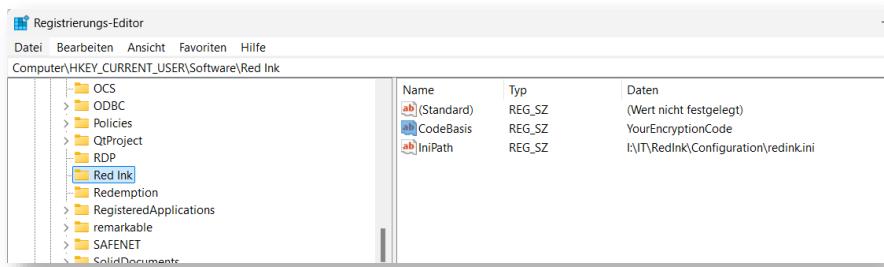
die Parameter aber geändert werden, wenn sie für das jeweils benutzte Modell nicht passen bzw. dieses sie nicht hinreichend befolgt. Der initiale Zugang dazu erfolgt über die "Expert Config"-Funktion in den Settings, wo sie manuell herauskopiert werden können (beim Speichern der Konfiguration werden sie aus verschiedenen Gründen nicht in die Konfigurationsdatei geschrieben, sofern sie dem aktuellen Standard der jeweiligen Version entsprechen). Den Prompts ist zu entnehmen, dass sie jeweils als System-Prompt aufgebaut sind und der selektierte Text jeweils als User-Prompt übergeben wird (sofern das Modell überhaupt unterscheidet). Der User-Prompt enthält den Text normalerweise zwischen den Tags "<TEXTTOPROCESS>" und "/<TEXTTOPROCESS>". Für gewisse andere Funktionen werden andere Tags verwendet, wie aus den Standard-Systemprompts zu entnehmen ist. Diese Tags sollten auch bei Änderungen benutzt werden, damit die Prompts optimal funktionieren. Die meisten der Systemprompts enthalten auch noch den Standard-Platzhalter "{INI\_PreCorrection}"; an dessen Stelle wird der Wert des entsprechenden Parameters eingetragen, der in der Konfigurationsdatei ebenfalls abgelegt werden kann. Der Benutzer kann ihn auch über Settings definieren.

- 317 Der **Schlüssel** für die Verschlüsselung des API-Key bzw. Private Key und der IniPath für die zentrale Konfigurationsdatei befinden sich nicht in der Konfigurationsdatei (siehe dazu Rz. 345 ff. und Rz. 320 ff. unten). Nicht in der Konfigurationsdatei (sondern in der lokalen Installation) gespeichert werden auch einzelne lokale Parameter wie etwa **der letzte Freestyle-Prompt** oder in Word die jeweils letztgültigen Parameter für den Chatbot.
- 318 In der Konfigurationsdatei werden Zeilen, die mit einem **Semikolon** beginnen, nicht berücksichtigt. Falls zwingende Werte (Keys) fehlen, fordert das jeweilige Add-in den Benutzer auf, sie einzugeben. Fehlt die Datei, wird der Setup-Assistent gestartet (siehe Rz. 286 oben).
- 319 Dateipfade können folgende **Platzhalter** verwenden (damit können die obigen Variablen für Dateien flexibler gestaltet werden):

%APPSTARTUPPATH%	Das Verzeichnis, aus welchem die jeweilige Anwendung gestartet wurde
%APPDATA%	Der Applications-Data-Ordner des jeweiligen Benutzers, z.B. C:\Users\Benutzername\AppData\Roaming; dieser Platzhalter eignet sich am besten für Red Ink, weil damit der Pfad zur lokalen Kopie der Prompt-Bibliothek gebaut werden kann; im Roaming-Verzeichnis befindet sich das Verzeichnis "Microsoft" und darunter die Verzeichnisse für Word, Excel und Outlook, wo sich auch eine etwaige lokale Kopie von "redink.ini" befindet
%USERPROFILE%	Das Profile-Verzeichnis des jeweiligen Benutzers, z.B. C:\Users\Benutzername
%WINDIR%	Das Windows-Verzeichnis, typischerweise C:\Windows

%TEMP%	Das Verzeichnis für temporäre Dateien
%HOMEPATH%	Das Verzeichnis des Benutzer-Profil, aber ohne Laufwerks-Angabe
%DESKTOP%	Das Desktop-Verzeichnis des jeweiligen Benutzers

- 320 Es ist wie oben angegeben möglich, einen **zentralen Pfad für die Konfigurationsdatei** "redink.ini" für alle Add-ins zu definieren, sodass die Konfigurationsdatei von einer zentralen Stelle geladen wird (und damit für alle Benutzerinnen und Benutzer einheitlich verwaltet werden kann). Hierzu muss auf allen Geräten im Betrieb ein Eintrag im Registry von Windows vorgenommen werden, und zwar wie folgt unter dem Schlüssel "IniPath" des Eintrags "Red Ink":



- 321 Der Registry-Pfad ist im Code der Add-ins (d.h. in deren geteilten Bibliothek) selbst bereits als Konstante hinterlegt, ebenso ob der Eintrag im Registry Vorrang haben soll oder eine im AppDir-Verzeichnis für die Konfigurationsdatei allenfalls vorhandene "redink.ini"-Datei. Standardmäßig ist, wie gezeigt (siehe Rz. 310 oben), letzterer Fall programmiert, d.h. falls im (lokalen) Standard-Verzeichnis jedes Add-in eine Konfigurationsdatei vorhanden ist, wird diese verwendet, ansonsten wird auf diejenige zurückgegriffen, die im WordDir ist und ansonsten auf jene im Pfad, der in der Registry verzeichnet ist (typischerweise die zentrale Kopie von "redink.ini"):

```
Public Const RegPath_Base As String = "HKEY_CURRENT_USER\Software\" & AN3 & "\"
Public Const RegPath_CodeBasis As String = "CodeBasis"
Public Const RegPath_IniPath As String = "IniPath"
Public Const RegPath_IniPrio As Boolean = False ' True if the registry path shall have priority
```

- 322 Ein Betrieb, welcher **Red Ink verteilen** will, sollte daher die Benutzer nur die drei Add-ins installieren lassen (und ggf. die Helper) und im Registry der Benutzer den entsprechenden Eintrag auf ein zentrales Verzeichnis vornehmen, wo sich eine Kopie von "redink.ini" für alle befindet. Will ein Benutzer oder eine Benutzerin eine abweichende Konfiguration, kann er oder sie dies durch Speichern der Konfiguration in Settings tun, womit er sich von der zentralen Konfigurationsdatei loskoppelt (er kann über Settings allerdings seine lokale Konfiguration auch jederzeit wieder "aufgeben" und zur zentralen Konfiguration zurückkehren).

323 Innerhalb des Add-in für Word kann der **Registry-Eintrag** lokal geschrieben werden, indem der Pfad in einem Word-Dokument aufgeschrieben wird (ohne "redink.ini"), er selektiert und Freestyle aufgerufen wird. Anstelle des Prompts ist "inipath" als Kommandozeilen-Befehl einzugeben (siehe Rz. 47 oben). Dies wird den selektierten Wert in die entsprechende Stelle im Registry schreiben, soweit dies zugelassen ist.

#### **B. Prompt-Bibliothek**

324 Das Word-Add-in unterstützt die Benutzung einer Prompt-Bibliothek, falls der entsprechende Parameter ("PromptLib") auf einen gültigen Pfad mit einer entsprechenden Text-Datei zeigt.

325 Die Text-Datei muss als reine Textdatei gespeichert sein (kein Word-file; innerhalb von Word als ".txt"-Datei speichern) und muss folgenden Inhalt aufweisen:

- In jeder Zeile ist zunächst der Titel oder ein Kurzbeschrieb des Prompts aufzuführen (z.B. "Vertragsvergleich"). Danach folgt ohne Abstand das Zeichen "|" (AltGr-7) und dann der Prompt (er darf das Trennzeichen enthalten);
- Für jeden Prompt ist eine neue Zeile zu verwenden;
- Leerzeilen und Zeilen, die mit ";" beginnen, werden ignoriert.

326 Die Prompt-Bibliothek kann auf dem eigenen Rechner gespeichert sein, im selben Verzeichnis wie die Konfigurationsdatei oder in einem geteilten Verzeichnis im Netzwerk. Findet das Add-in die Datei nicht, wird bei Anwendung der Funktion ein Fehler gemeldet (nicht schon beim Aufstarten des Add-ins).

327 Werden Änderungen an der Datei vorgenommen (was auch innerhalb des jeweiligen Add-in über den "Edit"-Knopf in der Auswahl der Prompt-Bibliothek möglich ist), werden diese beim nächsten Aufruf der Funktion berücksichtigt, weil sie jedes Mal neu geladen wird. So kann mit den Prompts experimentiert und sie können gleich angepasst werden, wenn sie noch nicht stimmen.

328 Die Prompts sind so zu formulieren, wie sie auch sonst in Freestyle benutzt werden können, d.h. auch die Prefixe und Trigger können benutzt werden. Im Installationspaket sind einige Musterprompts in einer Prompt-Bibliotheksdatei als Beispiel vorhanden – Verwendung auf eigene Gefahr. Wer einen guten Prompt hat, soll ihn uns mitteilen, damit wir ihn ggf. in die Sammlung aufnehmen. Wir haben schon den einen oder anderen interessanten Prompt in der Muster-Prompt-Bibliothek aufgenommen. Die neuste Fassung kann jeweils über <https://apps.vischer.com> bezogen werden.

329 Es ist möglich, in den Prompts der Prompt Bibliothek auch mehrere **Benutzerparameter** so einzubauen, dass der Benutzer vor dem Ausführen des Prompts aufgefordert wird, die entsprechenden Parameter einzugeben, die dann anstelle entsprechender Platzhalter im Prompt

automatisch eingefügt werden. Es wird hierfür dieselbe Syntax verwendet wie für die Parameter der Special Services und für die Skripte der WebAgent-Funktion, also z.B. "{Parameter3 = Which parts of the contract to analyze; String; Full contract; Full contract <the entire contract and provide a full report ("auto" mode)>, Selected sub phase <a selected Sub Phase ("step-by-step" mode)>}" (siehe dazu Rz. 76 und Anhang 3).

- 330 Dasselbe Dateiformat wird auch für die Prompt Bibliothek des Transcribers benutzt.

### C. Weitere alternative Sprachmodelle

- 331 In Word ist es möglich, bei Aufruf von "Freestyle (2nd)" statt dem ggf. konfigurierten sekundären Sprachmodell auch noch zwischen weiteren, alternativen Sprachmodellen zu wählen (Rz. 39). Hierzu muss eine Konfigurationsdatei definiert werden, in welcher die diversen alternativen Sprachmodelle mit ihrer Konfiguration hinterlegt sind, und der Pfad für diese Datei muss mit dem Parameter "AlternateModelPath =" in "redink.ini" festgelegt worden sein.
- 332 Format und Inhalt der Konfigurationsdatei für die alternativen Sprachmodelle entspricht derjenigen von "redink.ini" (dazu Rz. 309). Für jedes Modell sind dort die nötigen Angaben in derselben Weise zu hinterlegen wie für das primäre Modell (also z.B. mit dem Parameter "APIKey = ...." Oder "Endpoint = ..."). Der einzige Unterschied besteht darin, dass jede Modellkonfiguration mit einer Zeile eingeleitet werden muss, die in eckigen Klammern die Beschreibung des Modells enthält (dieser Titel dient gleichzeitig als Trenner für die einzelnen Segmente, die die Modellparameter des jeweiligen Modells enthalten):

```
[Perplexity Sonar Pro: Will also search the Internet (3.3 Min. Timeout, USA)]
APIKey = pplx-xxxxxx
APIKeyPrefix = pplx-
APIKeyEncrypted = True
Model = sonar-pro
Endpoint = https://api.perplexity.ai/chat/completions
HeaderA = Authorization
HeaderB = Bearer {apikey}
Response = content
APICall = {"model": "{model}", "messages": [{"role": "system", "content": "Follow the user's instructions as if they are drafted like a system prompt."}, {"role": "user", "content": "[promptsystem]\n[promptuser]"}], "temperature": {temperature}, "top_p": 0.9, "search_domain_filter": null, "return_includes_rerank": false, "return_related_questions": false, "top_k": 0, "stream": false, "presence_penalty": 0, "frequency_penalty": 0, "timeout": 200000, "temperature": 0.2}

[Perplexity Sonar Reasoning Pro: Will reason and search the Internet (6.6 Min. Timeout, USA)]
APIKey = pplx-xxxxxx
APIKeyPrefix = pplx-
APIKeyEncrypted = True
Model = sonar-reasoning-pro
Endpoint = https://api.perplexity.ai/chat/completions
```

- 333 Diese Beschreibung wird dann dem Benutzer angezeigt, wenn er sein Modell wählen soll, wenn er den "Freestyle (2nd)" aufruft (falls ein Pfad zur Konfigurationsdatei in "redink.ini" festgelegt worden ist).
- 334 Wählt der Benutzer ein solches alternatives Modell, wird es nach der Ausführung von Freestyle wieder zurückgesetzt auf das ursprünglich definierte sekundäre Modell. Diese Rücksetzung kann abgewählt werden.

den mit einer Checkbox in der Modellauswahl. Es bleibt dann bis zum Neustart von Red Ink oder dem erneuten Laden der "redink.ini" das sekundäre Modell. Wird die Konfigurationsdatei gespeichert, wird das bisherige sekundäre Modell überschrieben damit.

335 Achtung: Bei der Konfiguration von Reasoning-Modellen oder Modellen zur Bild-Generierung ist darauf zu achten, dass ein hinreichender Timeout konfiguriert wird und Red Ink den "Denkvorgang" nicht anzeigen. Der Benutzer muss also geduldig sein.

336 Einzelne Funktionen unterstützen die Zuweisung von spezifischen Modellen aus der obigen Liste. In diesen Fällen wird die Funktion jeweils im betreffenden Segment durch eine zusätzliche Parameterzeile aktiviert, jeweils mit Zuweisung des Werts "True":

- **FindClause = True:** Das Modell wird zum Durchsuchen der Klauseldatenbank der Funktion "Find Clause" benutzt. Damit kann ein kleineres und schnelleres Modell für diese Funktion standardmäßig gewählt werden. Sie wird dadurch schneller.
- **WebAgent = True:** Das Modell wird für das Ausführen des WebAgent verwendet. Es sollte ein Modell sein, welches gut darin ist, den HTML-Code von Webseiten zu analysieren, aber auch Inhalte von Websites zusammenzufassen.

#### D. OAuth2.0 (z.B. Google Vertex API)

337 Wer auf einen Endpoint eines Sprachmodells zugreifen will, muss sich bei diesem in der Regel mit Hilfe eines API-Keys ausweisen (so z.B. bei OpenAI und Azure OpenAI Services oder bei den freien Google-KI-Angeboten). Bei Endpoints wie Googles Vertex KI-API hingegen kommt das OAuth 2.0 Verfahren zum Einsatz, das mehr Sicherheit bietet, es aber auch komplizierter macht. Red Ink unterstützt es auch.

338 Bei diesem Verfahren muss auf dem Server zunächst ein sog. Dienstkonto (Service Account) eröffnet werden, also ein Konto speziell für Red Ink. Hierfür ist ein Zugriff auf die Administratoren-Konsole erforderlich. Ein solches Konto erlaubt den Zugang auch für Benutzer ohne eigenes Benutzerkonto. Für dieses Konto müssen dann ein Public und Private Key generiert werden. Er sollte in Form einer JSON-Datei exportierbar sein, weil wir den Private Key brauchen, um ihn in der Konfigurationsdatei von Red Ink abzulegen. Diese JSON-Datei sieht beispielsweise so aus:

```
{
 "type": "service_account",
 "project_id": "earnest-

```

- 339 Von dieser JSON-Datei übernehmen wir die Werte "private\_key" (ohne den Pre- und Suffix, nur den nackten Schlüssel; es kommt nicht darauf an, ob der Schlüssel selbst Zeilenumbrüche im Format '\n' enthält; sie sind zur Erhöhung der Sicherheit der Verschlüsselung zu entfernen), "client\_email" und "auth\_uri" sowie den passenden Scopes-Wert (im Falle von Vertex ist dies "https://www.googleapis.com/auth/cloud-platform"). Diese werden in der Konfigurationsdatei abgelegt, wobei der Private Key verschlüsselt abgelegt werden sollte (siehe nachfolgend, wie er mit Hilfe von Red Ink in Word verschlüsselt werden kann). Mit diesen Angaben kann Red Ink sich einen sog. Access Token beschaffen, der dann bis zu dessen Verfall (i.d.R. nach 3600 Sekunden, was auch konfiguriert wird) als eine Art API-Key benutzt werden kann. Danach beschafft sich Red Ink automatisch einen neuen Access Token. Der Private Key ist geheim zu halten. Wird er kompromittiert, so kann er allerdings über die Konsole des Anbieters des Endpoints neu gesetzt werden. Red Ink bietet mit der Möglichkeit einer leichten Verschlüsselung einen gewissen Schutz (siehe sogleich).

## E. Konfiguration erweiterter API-Calls

- 340 Für ein normales LLM genügt es, einen einfachen API-Call-Text zu definieren ("APICall") und anzugeben, in welchem Feld des zurückgegebenen JSON-Strings die Antwort des LLM enthalten sein wird ("Response").
- 341 Für die Special Service Dienste sind jedoch meist komplexere Programmierungen nötig, da Red Ink in diesen Fällen die Antworten teils in zwei Schritten einholen, teils auch aus den retournierten JSON-Strings in mehreren Schritten auslesen muss. Hier ein Beispiel einer komplexeren Programmierung (am Beispiel von LexiFind):

```

APIKey =
APIKeyPrefix =
APIKeyEncrypted = False
Model = LexFind
Endpoint = https://www.lexfind.ch/api/fe/de/fulltext-
search|https://www.lexfind.ch/api/fe/de/fulltext-search/{id}?session_id={session_id}&page_no=1
&results_per_page=25
HeaderA =
HeaderB =
Response = id;session_id|[**Lexfind.ch ({results[*].number_of_results|/} Treffer)**]
(https://www.lexfind.ch/fe/de/search/{id}/{session_id}/de) (hier max. 25):\n\n% for
texts_of_law_with_matches %|[**{systematic_number} - {matches[0].title}**]
(https://www.lexfind.ch/fe/de/{dta_urls[0].url}))\n\n{matches[0].keywords}\nStatus:
{matches[0].info_badge|removed=Entfernt;abrogated=Ausser Kraft;current=Aktuell;not_current=Nicht
Aktuell} - {matches[0].version_active_since} - {dta_urls[0].language} - [{entity.name}]
({dta_urls[0].original_url})\n\nFundstelle: {htmlnocr:matches[0].snippet}\n\n% endfor %
APICall =
{"search_text":{promptuser}","active_only":true,"search_in_systematic_number":{parameter3},"sear
ch_in_title":{parameter2},"search_in_keywords":{parameter4},"search_in_content":{parameter2},"ent
ity_filter":{parameter1},"systematic_filter":[],"category_filter":[],"use_global_systematics":tru
e,"direct_search":false}
Timeout = 200000
Parameter1 = Gemeinwesen; String; Bund (CH); Bund (CH)<[27]>, Bund und Kantone (alle)<[]>, Aargau
(AG)<[1]>, Appenzell Ausserrhoden (AR)<[3]>, Appenzell Innerrhoden (AI)<[2]>, Basel-Landschaft
(BL)<[5]>, Basel-Stadt (BS)<[6]>, Bern (BE)<[4]>, Freiburg (FR)<[7]>, Genf (GE)<[8]>, Glarus (GL)
<[9]>, Graubünden (GR)<[10]>, Intlex (Intlex)<[28]>, Jura (JU)<[11]>, Luzern (LU)<[12]>,
Neuenburg (NE)<[13]>, Nidwalden (NW)<[14]>, Obwalden (OW)<[15]>, Schaffhausen (SH)<[17]>, Schwyz
(SZ)<[19]>, Solothurn (SO)<[18]>, St. Gallen (SG)<[16]>, Tessin (TI)<[21]>, Thurgau (TG)<[20]>,
Uri (UR)<[22]>, Waadt (VD)<[23]>, Wallis (VS)<[24]>, Zug (ZG)<[25]>, Zürich (ZH)<[26]>
Parameter2 = Suche im Erlasstext; Boolean; True
Parameter3 = Suche im Titel/SR; Boolean; True
Parameter4 = Suche in Stichworten; Boolean; True

```

## 342 Red Ink unterstützt folgende Funktionen:

- Im Parameter "Endpoint" können auch zwei Endpoints definiert werden, getrennt durch "|". Der erste Endpoint wird normalerweise mit einem POST-Befehl angesprochen. Ist ein zweiter Endpoint definiert, wird dieser mit einem GET-Befehl angesprochen, wobei Ergebnisse aus dem ersten Befehl an deren Stelle des Platzhalters eingesetzt werden. Im obigen Beispiel sind dies die Werte {id} und {session-id}, die direkt aus der JSON-Antwort des POST-Befehls extrahiert werden. Diese beiden Werte sind in der linken Teil des "Response"-Parameters hinterlegt, d.h. dem Teil links von "|" (er dient zur Auswertung der vom Endpoint gelieferten Antworten). Die Antwort des zweiten Endpoints (d.h. des GET-Requests) werden vom rechten Teil ausgewertet (dazu so-gleich).
- Im Parameter "Endpoint" kann auch der erste Endpoint als GET-Befehl statt POST ausgeführt werden. Hierzu muss der URL der Prefix "GET:" vorangestellt werden.
- In beiden Fällen können in den Parametern "Endpoint" und "API-Call" diverse Werte mittels Platzhaltern eingefügt werden, die dann vor dem Aufruf eingefügt werden (bei "Endpoint" werden Leerzeichen werden mit "+" ersetzt):
  - {model} = Modellname
  - {apikey} = API-Key
  - {ownsessionid} = eine von Red Ink selbst generierte ID
  - {promptsystem} = der Befehl an das LLM

- {promptuser} = normalerweise der vom Benutzer selektierte Text (ohne "<TEXTTOPPROCESS>" -Tags)
- {userinstruction} = die nackte, vom Benutzer in Freestyle eingegebene Anweisung
- {temperature} = Temperatur (nur bei "APICall")
- Im Falle von Special Service: {parameter1}, {parameter2}, {parameter3}, {parameter4}
- Der Parameter "Response" kann auf drei Arten gesetzt werden:
  - Eine einzelne Bezeichnung wie "text" oder "response": In diesem Fall wird der erste entsprechende Wert aus dem JSON extrahiert;
  - Die Bezeichnung "JSON": In diesem Fall wird der JSON-String wie empfangen ausgegeben; das dient dem Debugging oder der Programmierung komplexerer Templates (dazu sogleich);
  - Ein komplexes Template, wie im obigen Beispiel: In diesem Fall wird Red Ink das Template abarbeiten und einen entsprechenden Text-String aufbauen. Da Red Ink Markdown-formatierte Strings ausgeben bzw. in der Pane anzeigen kann, kann das Template benutzt werden, um entsprechende Formatierungen einzubauen. Im Wesentlichen besteht ein Template somit aus Textelementen mit Formatierungen, aus Platzhaltern für aus der JSON-Antwort extrahierten Werten und aus einer Schleifenfunktion, um mehrere JSON-Elemente zu extrahieren. Nachstehend sind weitere Details enthalten.

Sind bei "Endpoint" zwei Endpoints definiert, müssen in "Response" auch zwei Elemente definiert werden, getrennt durch "|".

- 343 Eine ausführliche Anleitung, wie "Response"-Templates programmiert werden, findet sich im Anhang 2. Wem das zu kompliziert ist, dem bietet Red Ink eine automatische Programmierung an. Hierzu muss in Word ein Beispiel-Output des Endpoints angegeben werden (hierzu den "Response"-Wert "JSON" benutzen) und darunter beschrieben werden, wie der Output des Templates aussehen soll. Beides wird selektiert und Freestyle angewählt. Dort "generateresponsetemplate" eingeben. Red Ink wird dem aktuell konfigurierte Modell die nötigen Anweisungen geben, um das Template zu programmieren. Es erscheint kurz danach in Word.
- 344 Im Installationspaket sind etliche Konfigurationsbeispiele für diverse Special Service-Angebote enthalten, inklusive den Templates.

## F. Sicherheitsfunktionen

- 345 Von Red Ink ist der Quellcode offen einsehbar. Auch die Konfigurations-Datei ist offen einsehbar. Trotzdem kann der API-Key (oder der

Refresh Token und das ClientSecret bei der Verwendung von OAuth 2.0), der Zugang zum API des LLM gewährt, verschlüsselt abgespeichert werden. Auch ist es möglich, Red Ink so zu konfigurieren, dass die jeweilige Kopie nur in einem bestimmten Netzwerk läuft (was in gewissen Fällen nötig sein kann).

346 Die **Verschlüsselung** des API-Key, Refresh-Token und ClientSecret ("XOR") ist technisch nicht stark. Sie sollte aber genügen, um nicht spezialisierte Angriffe abzuwehren. Hinzu kommt, dass diese Angriffe typischerweise darauf aufbauen, dass Teile des verschlüsselten Klartextes bekannt sind, was hier nicht der Fall ist. Aus diesem Grund wird der bei API-Keys mitunter übliche Prefix maskiert (beim Private Key im OAuth 2.0-Verfahren wird aus diesem Grund das Pre- und Suffix weggelassen).

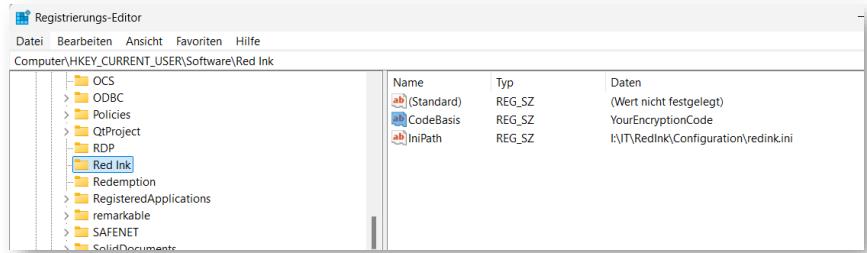
347 Der verwendete Schlüssel ist ein Text, der frei gestaltet werden kann (standardmäßig ist er nicht gesetzt, in den Beispielen unten hat er jedoch den Wert "SecretValue"). Er muss jedoch vor Zugriffen geschützt aufbewahrt, aber trotzdem für die Software zugänglich sein. Das ist bei quelloffener Software eine Herausforderung, wenn kein aufwändigeres Authentisierungsverfahren verwendet werden soll. Zur Aufbewahrung des Schlüssels für die Ver- und Entschlüsselung haben wir daher folgende zwei Methoden entwickelt:

- **Methode 1 (hardcoded):** Der Schlüssel für die Verschlüsselung befindet sich im Code des Add-in selbst, und zwar jeweils zu Beginn definiert als "private" Konstante "Int\_CodeBasis" ("private", damit es nicht über ein anderes Modul ausgelesen werden kann). Im Quellcode der Bibliothek der Add-ins (SharedLibrary) würde das dann so aussehen, wobei "YourEncryptionCode" für den eigenen Schlüssel stehen würde:

```
' Amend the following two values to hard code the encryption key and permitted domains
Private Const Int_CodeBasis As String = "YourEncryptionCode"
Public Const alloweddomains As String = "int.company.com, int2.company.com"
```

So kann ein Betrieb seinen eigenen Schlüssel wählen, ihn im VB.net-Code angeben, den Code neu kompilieren und die fertige ausführbaren Dateien installieren, wo der Code grundsätzlich nicht mehr einsehbar ist. Der Schlüssel bleibt also verborgen und ist nur für das Add-in selbst verfügbar. Der Nachteil dieser Methode ist, dass die Kompilierung des Codes zwar ohne Weiteres mit Entwicklungsumgebungen wie dem kostenlosen "Visual Studio 2022" mit kostenlosen Bibliotheken möglich ist, aber gewisse Fachkenntnisse voraussetzt. Außerdem muss die Prozedur der Code-Anpassung und Installation eines Passwortschutzes bei jeder neuen Version erneut vorgenommen werden. Gegen eine entsprechende Gebühr können wir jedoch entsprechend mit einem eigenen Schlüssel versehene Versionen der Add-ins liefern.

- **Methode 2 (registry):** Der Schlüssel für die Verschlüsselung wird nicht im Code selbst gespeichert, sondern in der Registry von Windows. Das ist der Ort, an dem die meisten Programme ihre Konfiguration ablegen. Dort wird ein eigener Eintrag für Red Ink erstellt und der Code im Klartext gespeichert.



Ist im Code der Wert *Int\_CodeBasis* nicht gesetzt (wie in Methode 1 vorgesehen), wird das Add-in immer dann, wenn die Verschlüsselung aktiv ist, versuchen in der Registry an der oben angegebenen Stelle den Schlüssel abzurufen. Der Pfad ist im Code ebenfalls als Konstante gesetzt und könnte dort bei Bedarf geändert werden. Der Vorteil dieser Methode ist, dass der Code nicht geändert werden muss. Der Nachteil dieser Methode ist, dass der Zugang zur Registry gesperrt werden muss für jene Benutzer, die den Schlüssel nicht zu Gesicht bekommen sollen (z.B. Zugriff auf "regedit" sperren). Das wird in vielen Unternehmen allerdings so oder so getan. Zu bedenken ist aber, dass jede Anwendung den Wert herauslesen kann. Wenn ein Unternehmen seinen Mitarbeitern beispielsweise erlaubt, selbst Makros in VBA zu schreiben, können diese ein kleines Skript erstellen, welches den Wert ausliest. Um den Wert in die Registry zu schreiben, kann der RegEdit-Befehl verwendet werden, es ist aber auch möglich, über das Word-Add-in den Eintrag vorzunehmen. Hierzu ist in einem Word-Dokument der gewünschte Schlüssel (z.B. "YourEncryptionCode") ohne Anführungszeichen einzutippen, ihn zu selektieren, die Freestyle-Funktion anzuwählen und statt des Prompts "codebasis" einzugeben (siehe dazu Rz. 47 oben). Das Add-in wird den Wert in die Registry schreiben (und einen bestehenden Wert überschreiben), wenn dies technisch möglich ist. Danach muss das Add-in neu gestartet werden, um den neuen Schlüssel zu laden.

- 348 Standardmäßig wird Methode 2 verwendet, wobei bei der ersten Installation kein Wert in der Registry gespeichert ist. Auch der Installer wird diesen Wert nicht setzen. Das muss also manuell gemacht werden.
- 349 Um den eigenen API-Key und den allfälligen Private Key damit zu verschlüsseln, wird Red Ink für Word installiert und dort der API-Key etc. in ein leeres Dokument kopiert, selektiert und die Funktion Freestyle aufgerufen. Dort wird als Befehl "encode" eingegeben (siehe dazu Rz.

47 oben). Als nächstes muss der geheime Code eingegeben werden. Red Ink wird den verschlüsselten API-Key in einer neuen Zeile in das-selbe Dokument schreiben. Achtung: Die automatische Silbentrennung ist auszuschalten (das Add-in versucht dies zu machen). Sonst wird der verschlüsselte Text mit zusätzlichen Trennstrichen kopiert und kann nicht mehr korrekt entschlüsselt werden. Ein verschlüsselter Text kann umgekehrt mit dem Befehl "decode" wieder entschlüsselt werden. Wichtig ist, dass ein etwaiger API-Prefix in der Konfigurationsdatei kor-rekt konfiguriert wurde (dies betrifft nur den API-Key, nicht den Private Key). Ist der API-Key bzw. der Private Key verschlüsselt, kann er in der Konfigurationsdatei eingetragen und der dortige Wert "APIKeyEncr-ypted" auf "Yes" gesetzt werden (im OAuth 2.0-Verfahren gilt diese Konfiguration jeweils für den Private Key). Dasselbe gilt für das zweite Modell. Vor der Verwendung des Private Key werden etwaige Zei-lenumbrüche bzw. "\n" aus dem entschlüsselten Text automatisch ent-fernt. Es spielt daher keine Rolle, ob diese einen solchen enthalten.

- 350 Die zweite Sicherheitsfunktion, **Run-only-at-home**, ist nur nötig, wenn oben die Methode 1 verwendet wird. Die Sicherheitsfunktion sorgt dafür, dass die betreffende Kopie der Add-ins nur jeweils in den programmierten Domänen läuft. Dies wirkt der Verwendung des Add-ins mitsamt API-Key etc. in anderen Unternehmen oder auf privaten Computern und so dem Missbrauch des eigenen API-Key etc. entgegen. Die eigene Domäne wird ebenfalls als Konstante "AllowedDo-mains" zu Beginn des Code-Moduls der gemeinsamen Bibliothek der Add-ins erfasst (im Beispiel hier die Domänen "int.company.com" und "int2.company.com"):

```
' Amend the following two values to hard code the encryption key and permitted domains
Private Const Int_CodeBasis As String = "YourEncryptionCode"
Public Const alloweddomains As String = "int.company.com, int2.company.com"
```

- 351 Um herauszufinden, welche die eigene Domäne ist und in welchen Do-mains Red Ink laufen kann, kann im Add-in für Word über die Funktion Freestyle der Kommandozeilen-Befehl "domain" eingegeben werden (siehe dazu Rz. 47 oben). Er zeigt beides an.

## V. HÄUFIGE FRAGEN (FAQ)

### 1. Wie unterscheidet Red Ink sich von anderen KI-Tools?

Das muss jeder selbst beurteilen. Aus unserer Sicht lässt sich damit viel besser als bei gewissen anderen Dienstleistungen und Produkten kontrollieren, was mit den eigenen Daten geschieht, wir haben wesent-lich mehr Funktionalitäten direkt im Programm integriert, und die Kos-ten zum KI-Einsatz sind viel tiefer als bei manchen KI-Services, wo je-des Abo pro Mitarbeiter und Monat 15-30 CHF oder EUR kostet. Der Einsatz von Red Ink kostet normalerweise ein Bruchteil davon. Schliesslich ist Red Ink für uns die Basis für weitere KI-Anwendungen,

die auf Red Ink als Schnittstelle zu den grossen Sprachmodellen dienen – zum Beispiel für andere Tools, die wir für die Analyse von juristischen Texten einsetzen. Für manche Dinge setzen wir in der Kanzlei weiterhin auch andere KI-Werkzeuge ein.

## **2. Wie gut funktioniert Red Ink wirklich?**

Das hängt wesentlich Teil vom jeweils verwendeten Sprachmodell ab. Denn Red Ink ist im Grunde nur eine Schnittstelle, um möglichst einfach und vielfältig auf ein Sprachmodell zuzugreifen. Wie gut ein Text korrigiert oder zusammengefasst wird, ob Red Ink den Anweisungen folgt oder die gewünschten Antworten liefert, hängt vom Sprachmodell ab. Damit Red Ink gut benutzt werden kann, sollte ein fortgeschrittenes Sprachmodell mit guter Fähigkeit, Instruktionen abzuarbeiten, benutzt werden.

## **3. Ich muss teilweise lange auf Antworten warten – warum?**

Das liegt zur Hauptsache am Sprachmodell; hier gibt es nach unserer Erfahrung grosse Unterschiede in der Geschwindigkeit. Je mehr Text verarbeitet werden soll, desto mehr Zeit benötigt das Sprachmodell natürlich. Werden zusätzlich noch die Funktionen zum "Merken" der Formatierungen benutzt, wird dies die Performance zusätzlich senken, weil bei den dabei verwendeten Techniken die Formatierungen (mehr oder weniger weitgehend, je nach verwendeter Methode, siehe Rz. 25 oben) im Text hinterlegt werden, was die zu verarbeitende Textmenge teilweise deutlich erhöhen kann. Auch die Rückcodierung und Umsetzung in Formatierungen braucht eine gewisse Zeit, weil Word (und Outlook) diese Dinge leider nicht wirklich gut unterstützt und daher für Red Ink eigene Lösungen gefunden werden mussten. Manchmal haben Wartezeiten aber ganz einfach auch damit zu tun, dass der Computer des jeweiligen Sprachmodells überlastet ist oder sonst aus einem Grund hängt.

## **4. Es macht mir in meinen Texten manchmal alle Formatierungen kaputt – was kann ich dagegen tun?**

Wir haben in Red Ink verschiedene Verfahren umgesetzt, mit denen das verhindert oder abgemildert werden kann. Diese können zugeschaltet werden, vermögen die Herausforderung aber nicht vollständig zu meistern, da die heutigen Sprachmodelle die Verarbeitung formatierter Texte nur beschränkt unterstützen. Die von uns eingesetzten Methoden funktionieren vor allem bei kürzeren oder teilweise mittellangen Texten, weil sie viel Verarbeitungskapazität beanspruchen, was den Prozess verlangsamen und die Sprachmodelle überfordern kann (zu den Methoden siehe insbesondere Rz. 25 oben). Bei längeren Texten muss daher mit Umgehungslösungen gearbeitet werden. Eine Strategie besteht darin, mit kürzeren Textblöcken zu arbeiten (vgl. etwa den "(iterate)"-Trigger in Freestyle), eine andere, die Kommentierungs- statt Überarbeitungsfunktion zu nutzen ("Bubbles:") – und die Kommentare dann manuell umzusetzen. Wer etwas mit dem Tool ex-

perimentiert hat, wird erfahrungsgemäss rasch herausfinden, was für die eigenen Bedürfnisse am besten passt.

#### **5. Ich habe "Keep character formatting" gewählt und trotzdem geht Fettschreibung etc. manchmal verloren. Warum?**

Sie geht verloren, weil Red Ink in diesem Fall nicht nur versucht, einfache Zeichenformatierungen wie Fettdruck zu erhalten, sondern auch die Absatzformatierungen. Diese können, je nachdem, wie sie definiert sind, Zeichenformatierungen wieder löschen. In einem ersten Schritt schreibt Red Ink ein Wort also wie erwartet fett, doch wird diese Fettschreibung aufgehoben, wenn Red Ink in einem zweiten Schritt dem Absatz die bisherige Formatierung zuweist und sie eine entsprechende Vorgabe zur Zeichenformatierung enthält. Diese Abfolge lässt sich leider nicht ohne Weiteres umkehren. Ebenso häufig geht Zeichenformatierung aber auch verloren, weil die KI sie in ihrer Antwort unterschlägt. Zu beachten ist ferner, dass die Funktion Keep character formatting nur bis zur definierten Zeichenzahl ausgeführt wird, um nicht zu lange Wartezeiten zu verursachen.

#### **6. Warum verschluckt Red Ink gewisse Textteile im bearbeiteten Text?**

Das kann vorkommen, weil Red Ink diese Textteile für Formatierungsbefehle oder interne Platzhalter hält und sie anwendet oder aus Gründen der (internen) Programmkompatibilität entfernt. Wird z.B. die Funktion zum Erhalt der Formatierung benutzt, wird vorgängig in jedem Text jede fette Textstelle mit zwei Sternchen links und rechts eingefasst. Dieser sog. Markdown-Code wird von der KI respektiert und findet sich in ihrer Antwort wieder, wo dann aus dem Text zwischen den Sternchen wieder Fettdruck entsteht. Kommen diese Doppelsternchen jedoch aus anderem Grund im Text vor, weiss dies Red Ink nicht, geht davon aus, es handelt sich um eine Fettmarkierung und setzt sie entsprechend um. Weitere solche Codes, die Red Ink verschlucken kann, sind Angaben in Spitzklammern ("<text>") und in geschwungenen Klammern ("{text}"). Auch hier hält Red Ink diesen Code für interne Formatierungen (HTML) und Platzhalter und entfernt sie im Output. Sie müssen manuell nachgeführt werden. Bei grösseren Texten, wo diese Codes vorkommen, empfiehlt es sich, sie vorher durch einen anderen Text zu ersetzen (z.B. "<" durch "[[" und diese Ersetzung später rückgängig zu machen.

#### **7. Wie kann ich eine Anpassung von Red Ink rückgängig machen?**

In Word und Outlook ist dies über die klassische Undo oder Rückgängig-Funktion möglich. Falls das nicht mit einem Klick geht, sollte versucht werden, die Taste Ctrl-Z mehrere Male zu drücken (in diesem Fall hat die Aufzeichnung der Anpassungen, die Red Ink gemacht hat, nicht funktioniert). Excel wiederum unterstützt die Verwendung der Rückgängig-Funktion bei Anpassungen von Zellen durch Red Ink nicht. Darum hat Red Ink eine eigene Funktion hierzu im Red-Ink-Menü ein-

gebaut (Undo Last Insert). Ist sie ausgeführt, kann allerdings der von KI generierte Inhalt nicht wieder zurückgeholt werden. Diese müsste also separat gesichert werden, bevor die Undo Last Insert-Funktion ausgeübt wird.

## **8. Unterstützt Red Ink auch Quellenzitate?**

Ja, wenn die Antwort der KI Zitate liefert und diese im JSON-Format (das intern für Antworten benutzte Format) als "citations" markiert hat (ggf. noch als "url"-Objekt), dann wird Red Ink diese Zitate am Ende des Textes anhängen. Das kann zum Beispiel mit den Modellen von Perplexity genutzt werden, die Internet-Fundstellen angeben.

## **9. Ich erhalte den Fehler 429. Was bedeutet das?**

Dieser Fehler kann mehrere Gründe haben. Es handelt sich um einen Fehler auf Seiten des KI-Service-Providers, nicht dem Add-in. Er tritt im Rahmen einer KI-Abfrage auf.

Der Fehler tritt erstens auf, wenn zuviel Text pro Sekunde an den Server gesandt wird. Er blockiert dann. Dieser Fehler kann auftreten, wenn mehrere Personen Red Ink über dasselbe Konto zugleich benutzen oder eine Person mehrere Abfragen schnell hintereinander vornimmt. Red Ink ist so programmiert, dass es bei dieser Fehlermeldung zunächst wartet und es dann bis zu zwei Mal nochmals versucht. Die Fehlermeldung erscheint nur, wenn das auch nichts hilft. Die Abfrage ist entweder erneut durchzuführen oder die Wartezeit zwischen zwei Abfragen zu erhöhen.

Der Fehler kann aber auch auftreten, wenn auf Seiten des KI-Service das eigene Konto nicht richtig konfiguriert ist. Bei OpenAI tritt er zum Beispiel auf, wenn zwar ein API-Key vorliegt, für diesen aber keine Kreditkarte hinterlegt und ein Betrag definiert ist. Dann antwortet der OpenAI-Server auf jede Anfrage mit diesem Fehlercode.

## **10. Plötzlich funktionieren die KI-Funktionen von Red Ink nicht mehr. Woran kann das liegen?**

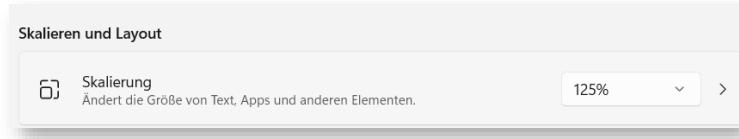
Dies kann natürlich verschiedene Ursachen haben, aber wenn es nicht am KI-Server liegt (temporäre Nicht-Verfügbarkeit), dann ist eine Möglichkeit, dass Red Ink aufgrund eines vorhergehenden Fehlers die Konfigurationsparameter aus seinem Arbeitsspeicher verloren hat. Zwar ist das Add-in darauf programmiert, diese in gewissen Fällen automatisch wieder einzulesen, aber das geht nicht immer. Am einfachsten ist es, das aktuelle Office-Programm vollständig zu schliessen (d.h. bei Word z.B. alle offenen Dokumente) und wieder neu zu starten. Damit wird Red Ink neu initialisiert. Hilft auch dies nicht, so sollte zuerst geprüft werden, ob die neueste Version installiert ist (mittels Edge-Browser auf <https://apps.vischer.com> die roten Buttons drücken). Hilft auch dies nicht, dann Red Ink deinstallieren und nochmals neu installieren.

**11. Warum befolgt Red Ink die Längenangaben bei Kürzungen und Zusammenfassungen manchmal nicht?**

Sprachmodelle haben mitunter Mühe beim Erfassen der Länge eines Textes, vor allem, wenn mit Zeichen gezählt wird. Der Grund liegt darin, dass Sprachmodelle selbst nicht mit Zeichen sondern sog. Tokens arbeiten, was eher Silben oder Wörtern entspricht. Wir geben daher Kürzungen in Wörtern an, womit Sprachmodelle besser zureckkommen. Aber sehr verlässlich ist das leider nicht.

**12. Warum sehe ich die Schrift in den Dialog-Boxen von Red Ink nicht vollständig bzw. sie wird komisch dargestellt?**

Dies kann geschehen, wenn in den Anzeigeeinstellungen ein hoher Wert zur systemseitigen Vergrößerung der Schrift gewählt wird (z.B. 175% oder 200%). Auch Auflösungseinstellungen können sich auswirken. In diesem Fall übersteuert Windows die Schriftgrösse, die Red Ink wählt und es ist u.U. nicht mehr alles wie gewohnt zu sehen. Lassen Sie uns wissen, wo das passiert, und wir versuchen eine Lösung zu finden.



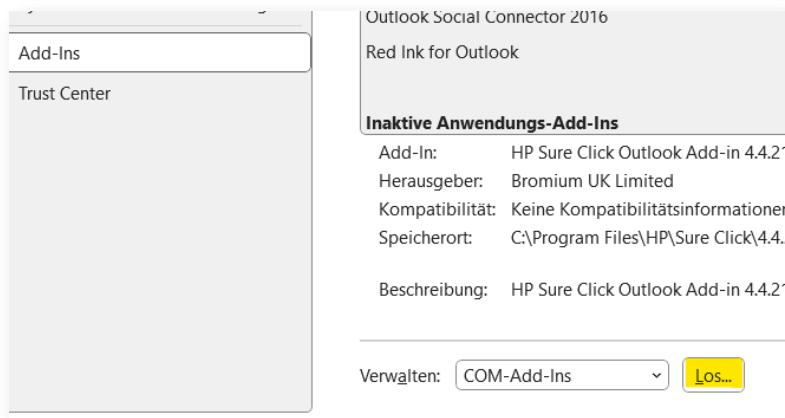
**13. Warum gehen beim Einsatz in Word von Markups wie von Geisterhand neue Fenster auf?**

Das geschieht dann, wenn für die Erstellung von Markups die Word-interne Vergleichsfunktion verwendet wird. Sie funktioniert so, dass drei Dokument vorübergehend erstellt und geöffnet werden (würden wir sie ausblenden, würde das zu einem Fehler führen). Darum sind sie kurzzeitig zu sehen. Das ist aber im Allgemeinen unproblematisch, solange der Prozess nicht unterbrochen wird. Wir haben weiter festgestellt, dass gewisse andere Add-ins dies tun können. Das Dokumenten-Management-System "iManage" schläuft sich zum Beispiel ungefragt in den Prozess ein und fragt, ob die temporären Dateien gespeichert werden sollen, obwohl Word von Red Ink den Auftrag erhält, nicht zu fragen. Dies ist aus unserer Sicht ein Fehler von iManage, den wir bisher aber nicht umgehen konnten und iManage hat ihn bisher nicht korrigiert. Wen das stört, kann versuchen die andere(n) Markup-Methode(n) einzusetzen.

**14. Ich habe Red Ink für Outlook installiert, es wird beim Laden auch kurz angezeigt, aber es erscheint nicht im Programm.**

Das hat in der Regel damit zu tun, dass Outlook Add-ins, die wiederholt mehr als 0.5 bis 1 Sekunde Ladezeit haben, von selbst ausschaltet. Wir versuchen das mit einem Trick zu verhindern, aber es gelingt je nach System leider nicht. Stattdessen muss diese Massnahme durch eine entsprechende Anpassung der Konfiguration von Outlook abge-

stellt werden ("Always enable"). Auch lässt sich Red Ink nachträglich wieder aktivieren. Hierzu ist in den "Optionen" (Datei-Menü) das Untermenü "Add-ins" und dort ganz unten die Option "Verwalten" bzw. "Los ..." für "COM-Add-Ins" zu wählen. Dort kann dann Red Ink wieder aktiviert werden. Outlook muss dazu nicht neu gestartet werden:



## 15. Verlangsamt Red Ink Word, Excel oder Outlook?

Nein, nach unserer Erfahrung im normalen Betrieb nicht. Wir haben aber im Rahmen unserer eigenen Entwicklungsarbeiten festgestellt, dass insbesondere Word im Laufe der Zeit langsamer werden kann, weil die Standard-Formatvorlage "normal.dotm" über die Zeit anwächst, vermutlich weil sie irgendwelche Datenreste aufnimmt und diese nicht ordnungsgemäß entsorgt. Bei uns hat die Datei zwischen 100-200 KB, bei gewissen Personen ist sie aber auf 2-3 MB angewachsen. Das verlangsamt Word enorm. Das Problem wird gelöst, indem die Datei "normal.dotm" wieder mit einer frischen, sauberen Version ersetzt wird (sie ist i.d.R. im Verzeichnis "%AppData%\Microsoft\Templates"). Das Laden des Add-in und der Konfigurationsdateien beim Aufstarten (was bei Word und Excel beim Öffnen eines Textes erfolgt) braucht natürlich einen kurzen Moment. Einen weiteren Moment braucht das Erstellen des Kontext-Menüs vor allem in Word; wem das zu lange geht, kann dies über eine Konfigurationseinstellung ausschalten.

## 16. Wo ist im Add-in für Outlook die Kachel von Red Ink?

Diese erscheint selbst bei installiertem Add-in primär dann, wenn ein Fenster zum Verfassen einer Mail geöffnet wird, also um auf eine Mail zu antworten, eine Mail weiterzuleiten und natürlich eine Mail neu zu schreiben. Zudem muss es sich um ein HTML- oder RTF-Mail handeln, d.h. eine Mail, die grundsätzlich auch Formatierungen enthalten kann. Wer eine "Nur Text"-Mail mit Hilfe von Red Ink bearbeiten oder befragen will, muss also zuerst auf den HTML-Modus umschalten, die Mail abspeichern und erneut öffnen. In den meisten Fällen wird heute aber mit HTML-Mails gearbeitet. Wir haben seit einiger Zeit zwar eine zweite Kachel für Red Ink eingefügt, die auch beim Stöbern in den E-Mails be-

nutzt werden kann, aber dies nur für die "Sumup"-Funktion (wenn mehrere E-Mails selektiert sind). Wird eine E-Mail in der Pane bearbeitet und dann die Red Ink-Kachel mit einem Befehl angeklickt, dann wird die Mail in ein eigenes Fenster übertragen.

**17. Wird das "neue" Outlook auch unterstützt?**

Nein. Das hat damit zu tun, dass Microsoft für das neue Outlook viel weniger weitgehende Befehle für die Programmierung von Add-ins bereitstellt. Viele der Dinge, die Red Ink tut (vor allem mit den Texten), könnten im neuen Outlook nicht realisiert werden. Das bisherige Outlook wird aber den bisherigen Ankündigungen zufolge bis 2029 zur Verfügung stehen. Wir selbst haben nicht vor, auf das neue Outlook zu migrieren; mit einem Wechsel dürften auch diverse andere Add-ins nicht mehr funktionieren.

**18. Warum ist der in Word hinterlegte Autor plötzlich "Red Ink"?**

Das ist auf eine Fehlfunktion zurückzuführen. Kommt es bei der Durchführung von Markups durch Red Ink (z.B. im Chatbot) zu einem Fehler, so dass die Funktion nicht korrekt beendet wird, kann es vereinzelt dazu kommen, dass die Rückstellung der Autorenangabe in Word nicht funktioniert. In diesem Fall ist sie manuell vorzunehmen. In den neueren Versionen der Software wird nicht mehr mit dem Autor "Red Ink" gearbeitet.

**19. Im Transcriptor erhalte ich die Fehlermeldung beim Einsatz von Whisper, dass gewisse "native libraries" fehlen. Was muss ich tun?**

Es handelt sich hierbei um zusätzliche Programmteile, die ins selbe Verzeichnis kopiert werden müssen wie die Sprachmodelle von Whisper (d.h. dort in ein Unterverzeichnis "runtimes"). Diese Programmteile sind im Installationspaket enthalten. Für mehr Informationen siehe Rz. 99 oben.

**20. Im Transcriptor wird kein Transkript angezeigt, obwohl der Transkriptor ohne Fehlermeldung gestartet wird.**

Das kann diverse Gründe haben. Häufigste Ursache ist, dass die gewählte Tonquelle keine Tondaten liefert, die transkribiert werden können. Ist die richtige Quelle gewählt und ist sie richtig konfiguriert (z.B. Lautstärke), kann das u.a. daran liegen, dass die Quelle von einer anderen Anwendung, wie z.B. ein Videokonferenz-Client, exklusiv "gebucht" ist. In diesem Fall ist ein anderes Mikrofon zu wählen, das z.B. den Raumklang aufnimmt, falls mit Lautsprecher konferiert wird. Als praktisch hat sich oft die Tonquelle "Stereo-Mix" oder "Stereomix" erwiesen, die an sich alles, was an Tönen vom Computer verarbeitet wird, kombiniert. Aber auch dies ist nicht zuverlässig. Wir haben in neueren Version des Add-in für jede Version des Mikrofons, das gewählt werden kann, auch die Option "(plus audio output)" vorgesehen; damit wird der Transkriptor angewiesen, dem Mikrofon-Signal noch die

derzeit ausgewählte Standard-Tonquelle beizumischen (diese kann über den "Dev"-Knopf bestimmt werden). Es gibt trotzdem Situationen, in denen keine vernünftige Tonquelle zur Verfügung steht (weshalb dann z.B. in einer Videokonferenz nur die eigene Stimme, nicht aber der Ton der anderen transkribiert wird). Eine weitere Ursache ist, dass zu wenig lange gewartet wird oder ein zu grosses Modell gewählt worden ist. Je grösser das Modell, desto mehr Zeit vergeht bis zum Erscheinen der Transkription – und manchmal wird das Modell schlicht zu gross sein. Typischerweise funktioniert die Transkription auf einem normalen PC nur mit kleinen oder sehr kleinen Modellen. Diese haben allerdings den Nachteil, dass sie fehleranfällig bzw. weniger gut sind. Wir empfehlen für die Live-Transcription die Verwendung von Cloud-basierten Modellen.

**21. Bei dem von mir generierten Podcast fehlt eine Stimme oder es wird überhaupt keine Stimme aufgenommen.**

Das hat vermutlich damit zu tun, dass das Podcast-Script SSML-Kommandos enthält, welche die gewählte(n) Stimme(n) nicht unterstützt. Dasselbe kann bei Anpassungen der Standardwerte für Pitch (0) und Speaking Rate (1) geschehen. In diesem Fall sollten nur die Standardwerte verwendet und den Haken bei "No SSML" gesetzt werden.

**22. Die Kachel mit dem Logo reagiert nur verzögert – woran liegt das?**

Das liegt daran, dass nicht das originale, weisse Windows-Farbschema benutzt wird (sondern z.B. "Grey"). Red Ink funktioniert, aber langsamer. Wir empfehlen die Nutzung des normalen Farbschemas.

**23. Das Add-in für Outlook sagt mir, dass nur HTML- und RTF-Mails unterstützt werden – was soll ich tun?**

Im Reiter "Text formatieren" von Outlook kann eine "Nur Text"-Mail in eine HTML-Mail konvertiert werden.

**24. Warum erscheint im Menü neben der Kachel mit dem Logo statt Direktzugriff-Buttons nur ein blauer Kreis?**

Hat es (zu)viele Kacheln im Menüband, versuchen die Office-Anwendungen die Kacheln zu verdichten, indem sie stattdessen den Kreis anzeigen. Wird mit der Maus draufgeklickt, wird die Kachel vollständig angezeigt. Dieses Verdichten kann programmiertechnisch nicht verhindert werden. Was hilft, ist nicht benötigte Kacheln zu entfernen und die Red Ink-Menüs nach links zu verschieben (über das Anpassen des Menübands).

**25. Warum liefert Red Ink Ergebnisse in der falschen Sprache?**

Dies hat damit zu tun, dass das verwendete Sprachmodell die ihm aufgetragenen Anweisungen nicht korrekt ausführt. Gewisse Sprachmodelle brauchen hier mehr Instruktionen als andere oder können nicht mehrere Instruktionen verarbeiten. Allenfalls kann es helfen, den "Pre-

Correction"-Parameter zu nutzen, um dem Sprachmodell zusätzliche Anweisungen mit auf den Weg zu geben (z.B. "Deine Ausgabe muss immer auf Deutsch erfolgen."). Im Falle von Sum-up innerhalb von Outlook haben wir ferner beobachtet, dass sich die Spracherkennung von Standardangaben von Outlook (wie "Absender" oder "Betreff" irreführen lässt, die immer in der Sprache der Installation erscheinen). Auch beim Chatbot folgt die KI nicht immer genau der Anweisung, immer in der Sprache des letzten Benutzerbefehls zu antworten.

**26. Ich habe die Browser-Erweiterung installiert, aber das System reagiert nicht, wenn ich einen Befehl anwähle.**

Das kann mehrere Ursachen haben. Zunächst muss Outlook mit dem Red Ink Add-in geladen und aktiv sein, da die Browser-Erweiterung davon abhängt (und vom Add-in für Word, falls die "Send to Word"-Funktion benutzt werden soll). Outlook sollte also zuerst gestartet werden. In der Praxis kann es auch vorkommen, dass das Fenster, z.B. zur Sprachwahl, zwar aufgeht, von anderen Fenstern aber verdeckt oder vom Benutzer nicht beachtet wird. Schliesslich ist es auch möglich, dass Sicherheitseinstellungen die Kommunikation zwischen Browser und Red Ink verhindern, oder die Browser-Erweiterung deaktiviert ist (benutzt werden die Ports 12333 und 12334 von 127.0.0.1 mittels http-Protokoll).

**27. Kann aufgezeichnet werden, was die Benutzer mit Red Ink tun?**

Dies sieht Red Ink selbst nicht vor, d.h. es enthält bisher keine Protokollierungsfunktion bzw. keinen Audit Trail. Es ist denkbar, eine solche in das Tool zu integrieren, vor allem für den Einsatz in Unternehmen, aber dies setzt voraus, dass es einen zentralen Server im Netzwerk gibt, an welchen die Inputs und Outputs gesandt werden können. Wir können eine solche Funktion realisieren, sollte sie gewünscht sein. Es ist aber zu beachten, dass hierbei dann, wenn mehr als nur die Kommandos protokolliert werden sollen, sehr viel Text anfällt.

**28. Mit welchen Prompts arbeitet Red Ink und kann ich sie ändern?**

Ja. Alle Prompts, mit denen Red Ink arbeitet, können über die Konfigurationsdatei ausgelesen und in dieser geändert werden. Hierzu ist in Red Ink Settings zu öffnen und dort die "Expert Config" anzuwählen. Es können die Prompts herauskopiert werden. Welche wofür sind, haben wir in der Konfiguration für Fortgeschrittene beschrieben (siehe Rz. 309 ff. oben). Wir behalten uns vor, die Prompts selbst von Zeit zu Zeit zu überarbeiten und zu verbessern, darum werden sie nicht in die Konfigurationsdatei geschrieben, sofern sie dem Standard entsprechen. Umformulierte Prompts werden jedoch aus der Konfigurationsdatei gelesen. Wer daher dort seine eigenen Prompts hinterlegt hat, wird in diesen Fällen nicht von unseren Verbesserungen profitieren, bis sie in der Konfigurationsdatei wieder gelöscht sind.

**29. Kann die Installation der Helper für Word und Excel automatisiert werden?**

Ja, das ist einfach möglich. Die beiden Dateien "redink\_helper.dotm" und "redink\_helper.xlam" müssen bloss aus dem Installationspaket in die betreffenden Verzeichnisse von Word und Excel kopiert werden (siehe Rz. 291 ff. oben). Alles andere passiert beim Aufstarten von Word und Excel automatisch. Dieser Kopiervorgang kann z.B. in eine Batch-Datei ausgelagert werden. Im Installationspaket ist ein Beispiel enthalten. Alternativ kann der Knopf "Install Helper" in Word bzw. Excel benutzt werden; die Dateien werden dann von unserem Server direkt heruntergeladen und in das richtige Verzeichnis kopiert, falls die Sicherheitseinstellungen dies erlauben.

**30. Bei der Installation des Helpers für Excel sagt mir Windows, dass die Datei Malware enthält. Stimmt das?**

Das ist ein Fehlalarm von Windows Defender, der leider immer wieder vorkommen kann und ärgerlich ist, weil Windows die Datei dann löscht. Wir haben das Problem gemeldet, aber bisher ist es noch nicht behoben. Manchmal hilft es zu warten; der Fehlalarm tritt auch auf demselben Computer nicht immer auf. Das Problem lässt sich lokal beheben, indem die betreffende Datei auf die "weisse Liste" gesetzt wird bzw. Windows angewiesen wird, diese angebliche "Bedrohung" zuzulassen. Wir gehen davon aus, dass der Fehlalarm daher röhrt, dass unser Excel-Helper Funktionen zur Übermittlung von Daten enthält, da er ja mit der KI kommunizieren können muss.

**31. Warum funktioniert die Installation von Red Ink auf meinem Bürocomputer nicht?**

Das hat höchstwahrscheinlich mit den Sicherheitseinstellungen zu tun, mit welchen viele Betriebe die Installation von Software durch ihre Mitarbeiter unterbinden, weil diese Schadcode enthalten kann oder sonst unerwünscht ist. Gehen Sie in diesem Fall auf Ihre IT-Abteilung oder Ihren IT-Dienstleister zu. Falls uns als Quelle vertraut wird, kann sie oder er die Installation z.B. basierend auf unseren digitalen Zertifikaten oder unserem Deployment-Server <https://apps.vischer.com> freigeben. Neben der Freigabe für die eigentlichen Add-ins muss darauf geachtet werden, dass möglichst auch die beiden Helper-Programme für Excel und Word installiert und laufengelassen werden dürfen. Sie enthalten Makros bzw. VBA-Code für Excel und Word, was in manchen Unternehmen auch gesperrt wird. Hier ist gleich zu verfahren. Zwingend ist der Einsatz der Helper für die Grundfunktionen von Red Ink allerdings nicht.

**32. Ist Red Ink als Programm sicher?**

Das muss jeder für sich selbst beurteilen. Der Quellcode von Red Ink ist offen zugänglich, für jeden, der ihn prüfen möchte, und so kann gelesen werden, was die Software genau tut und welche Bibliotheken von Dritten zum Einsatz kommen (die alle auch quelloffen sind). Wer

will, kann Red Ink mit den auf Github publizierten Dateien selbst auf Basis des geprüften Quellcodes kompilieren und einsetzen. Dies erfordert allerdings Programmierkenntnisse. Wer uns vertraut wird hingegen möglicherweise einfach darauf abststellen wollen, dass die von uns ausgelieferten Dateien von uns (VISCHER AG) digital signiert sind.

**33. Gehen Daten über die Benutzung von Red Ink an VISCHER?**

Nein, VISCHER sammelt keine Daten über die Benutzung von Red Ink. Auch die eingebaute Funktion zur Lizenzkontrolle besteht aus einem Datum, das der Benutzer selbst konfigurieren kann. Die eingebaute Funktion gibt VISCHER *keine* Rückmeldung. Das Tool hat keinen internen Schalter, mit welchem wir es aus der Ferne deaktivieren können (die Software verfügt aber über einen Schalter, mit dem ein Betrieb die eigene Kopie der Tools so codieren kann, dass das Tool nur in der eigenen Domäne läuft, siehe Rz. 350 oben). Red Ink hat aber eine Funktion, mit welcher es regelmässig automatisch auf unserem Server nach Updates sucht und diese wenn möglich auch installiert (wobei keine Anmeldung oder Registrierung nötig ist, d.h. wir identifizieren die Personen nicht). Ansonsten greift Red Ink nur auf die installierten KI-Endpoints zu und – falls aktiviert – die konfigurierte Suchmaschine. Schliesslich haben die Add-ins für Word und Outlook die Möglichkeit, über einen bei ihnen eingebauten http-Listener Daten zu empfangen und anzuzeigen. Das benutzen wir, wie oben beschrieben, für die Browser-Erweiterung.

**34. Ich traue den KI-Anbietern in der Cloud nicht – kann ich Red Ink auch nur lokal einsetzen?**

Ja, das ist ohne weiteres möglich, sofern auf dem eigenen System oder im eigenen Netzwerk ein geeignetes Sprachmodell betrieben wird. Solche Sprachmodelle sind kostenlos verfügbar, erfordern aber erfahrungsgemäss einen leistungsfähigen Server. Das Tool kann auf diesen Endpoint konfiguriert werden.

**35. Ich habe bei der Verwendung von Red Ink eine Fehlermeldung erhalten, die ich nicht verstehe – was soll ich tun?**

Neben den gängigen Fehlermeldungen, die auf einen Bedienfehler zurückzuführen sind, kann es bei der Software auch zu internen Fehlern kommen. Notieren Sie die Situation, die den Fehler verursacht hat, und teilen Sie uns diese mit dem Fehler mit. Das hilft uns, das Tool zu verbessern. Es ist mit inzwischen gegen 10'000 Codezeilen ein komplexeres Programm geworden.

**36. Wer hat Red Ink entwickelt?**

Red Ink wurde von David Rosenthal in seiner Freizeit entworfen, entwickelt und geschrieben. Natürlich ist KI zum Einsatz gekommen, ebenso kommen diverse Bibliotheken zum Einsatz, z.B. für Verschlüsselungsfunktionen, die Analyse von HTML-Code, das Lesen von PDF-Dokumenten oder die Diff-Methode. In der ursprünglichen Fassung der

Add-ins wurden diese allerdings in VBA entwickelt, wo keine Bibliotheken verwendet werden konnten. In dieser ersten Generation von Red Ink (die ebenfalls noch verfügbar ist) musste alles ausprogrammiert werden, mit Ausnahme der Kryptofunktionen, für welche ein externes Programm benutzt wurde. Mit der Portierung der Add-ins auf VB.net konnten wir Bibliotheken nutzen und brauchen das externe Programm nicht mehr, doch mussten wir die Add-ins im Grunde neu schreiben.

### **37. Warum wurde Red Ink ausgerechnet von VISCHER, einer Anwaltskanzlei, entwickelt?**

Das haben wir zunächst nur für uns selbst getan. Es gab drei Gründe: Erstens haben wir keine Programme gefunden, die das tun, was Red Ink kann. Zweitens wollten wir eine Lösung haben, die wir mit einem API-Zugang konfigurieren können, damit wir sie auch mit Dokumenten nutzen können, die dem Berufsgeheimnis unterliegen. Drittens ging es darum, Kosten zu sparen, weil ein Anbieter einer KI-Übersetzungslösung seine Preise für 2025 massiv heraufgesetzt hat. Dank Red Ink fallen hierfür jetzt sehr viel weniger Kosten an, weil das Tool im Gebrauch viel günstiger ist. Inzwischen haben mich das positive Feedback und immer weitere Ideen angetrieben, das Tool immer noch besser und leistungsfähiger zu machen – und öffentlich anzubieten. Da David Rosenthal in seinen jungen Jahren als Softwareentwickler tätig war und die KI heute beim Programmieren viel Unterstützungsarbeit leisten kann, konnte er das Vorhaben in einigen Monaten nebst seiner normalen Tätigkeit umsetzen.

### **38. Warum heisst das Tool "Red Ink"?**

Das Tool hiess intern "Red Dragon" entsprechend dem Spitznamen, mit welchem eine frühere Mitarbeiterin von David Rosenthal diesen (liebevoll gemeint) bedacht hatte, weil er ihr ihre Entwürfe an Antworten für Klienten zu Beginn jeweils mit massiven (roten) Markups zurückgab. Weil das auch eine Funktion dieses Werkzeugs ist, gaben wir dem Werkzeug intern diesen Namen. Es kann natürlich inzwischen sehr viel mehr als nur Texte verbessern. Nachdem wir jedoch eine Wortmarke für "RED DRAGON" eintragen liessen, meldete sich nach einiger Zeit Microsoft und untersagte die Verwendung mit der Begründung, der Begriff sei insbesondere zu nahe an der Marke "DRAGON" ihres Spracherkennungsprodukts. Wir waren nicht der Auffassung, dass eine Verwechslungsgefahr bestand, wollten aber unsere Ressourcen nicht in einen Rechtsstreit, sondern ein besseres Produkt investieren (auch als Copilot von Microsoft) und änderten noch vor der Lancierung den Namen auf "Red Ink", wie rote Tinte – die Farbe von Fehlerkorrekturen. Das ebenfalls als Marke eingetragene Logo zeigt ein Fabelwesen – eine Mischung aus Seepferd, Drachen und Seeungeheuer. Wir nennen es "Inky", und es kann über den Chatbot mit ihm kommuniziert werden.



**39. Wer darf Red Ink unter welchen Bedingungen benutzen?**

Dies ist den von uns publizierten Lizenzbedingungen zu entnehmen (siehe dazu <https://vischer.com/redink> und <https://apps.vischer.com>). In der öffentlichen Testphase wird es jeder kostenlos benutzen können. Danach ist vorgesehen, dass der persönliche Gebrauch der Generation 2 von Red Ink kostenlos sein wird. Machen Unternehmen es ihren Mitarbeitenden zugänglich, werden wir je nach Konstellation Lizenzen eine moderate jährliche Lizenzgebühr verlangen (in Selbstkontrolle). Wir werden Details auf <https://vischer.com/redink> publizieren. Der Einsatz von Red Ink wird aber selbst unter Einbezug der Kosten für einen API-Zugang und etwaigen Lizenzgebühren nach unserer Erfahrung vielerorts immer noch deutlich preiswerter, als wenn alle Benutzer mit Abos bestimmter kommerzieller Anbieter ausgestattet werden.

**40. Wird Red Ink weiterentwickelt?**

Ja, wer weitere gute Ideen hat, die sich damit umsetzen lassen, soll sich bei mir melden ([david.rosenthal@vischer.com](mailto:david.rosenthal@vischer.com)). Auch, wenn Fehler entdeckt werden, auch wenn ich keinen Support gewährleisten kann. Ich selbst komme immer wieder auf neue Ideen, was sich noch einbauen lässt, das uns die Arbeit erleichtert und uns die Möglichkeiten grosser Sprachmodelle noch besser nutzen lässt.

**41. Wo kann Red Ink bezogen werden?**

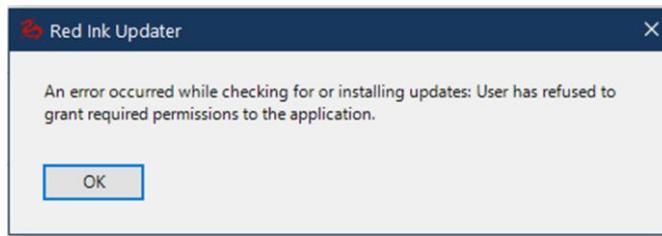
Alle Angaben befinden sich auf <https://vischer.com/redink> und die Dateien für die Installation auf dem Deployment-Server auf <https://apps.vischer.com>. Die Browser-Erweiterung wird über die Edge- und Chrom-Add-on-Store von Microsoft und Google vertrieben.

**42. Gibt es Updates für Red Ink?**

Updates werden von Zeit zu Zeit angeboten, allerdings ohne Versprechen. Red Ink prüft regelmässig selbst auf Updates und installiert diese wenn möglich. Ferner ist jederzeit eine De- und Neuinstallation möglich. Da die Konfiguration separat gespeichert wird, geht dabei nichts verloren.

**43. Ich erhalte beim Aufstarten meiner Programme eine Fehlermeldung des Red Ink Updater.**

Wenn diese Fehlermeldung erscheint, dann liegt dies daran, dass die Installation fehlerhaft erfolgt ist oder der Zugriff auf das Internet gesperrt ist. Diese Fehlermeldung kann auch durch Word selbst ausgelöst werden, wenn Red Ink von einem anderen Benutzer oder als Administrator installiert worden ist. In diesen Fällen hilft es, Red Ink zu deinstallieren und neu zu installieren.



In den neueren Versionen von Red Ink wird diese Meldung ein Mal gezeigt und das Update erst nach einem neuen Zyklus wieder versucht, damit die Meldung nicht zu oft erscheint.

#### 44. **Wie kann Red Ink deinstalliert werden?**

Das erfolgt über die betreffende Windows-Funktion zum Deinstallieren von Software. Red Ink taucht dort für jede Office-Anwendung separat auf und kann binnen weniger Sekunden deinstalliert werden. Die Helper können einfach gelöscht werden. Die Deinstallation entfernt die Konfigurationsdateien und Helper nicht. Diese müssen bei Bedarf manuell gelöscht werden (die Speicherorte sind in Rz. 291 ff. oben beschrieben). Die Helper können auch mit der "Remove Helper" Funktion in Settings von Word oder Excel gelöscht werden (sofern die Programme dies nicht blockieren).

## VI. RELEASE NOTES

352 Die nachfolgende Tabelle dokumentieren die diversen Arbeiten und Anpassungen von Generation 2 von Red Ink (nur ab Beta Test Version):

Datum, Build	Release Notes
10.2.2025  W=0.1.1.0 O=0.1.1.0 E=0.1.10	Release von Beta Test Version
14.2.2025  W=0.1.1.1 O=0.1.1.1 E=0.1.1.1	Korrektur der Setup Wizard Header-Angaben for Google Gemini API; Header-Angaben sind neu Optional; aktives Löschen der Interop-COM-Objekte in Outlook; stabilere Absicherung des httplisteners in Outlook; Undo-Funktion in Excel; Deselect nach Einfügungen in Word; Anpassung des Textes in der Podcast-Maske
19.2.2025  W=0.1.1.2 O=0.1.1.2 E=0.1.1.2	Unterstützung Perplexity Zitate (beim Einsatz von Sonar etc. als Modell) und zusätzlicher Platzhalter bei APICall-Parameter; Fehler entfernt, wodurch Word im Extended Selection Mode blieb; Fehler entfernt, wodurch Word nicht immer die Config-Datei lud (auch bei Outlook wird Laden der Config-Datei besser sichergestellt); Fehler bei der Menüdarstellung von Excel behoben; Fehler in der Darstellung von Settings behoben; Anpassung beim httplistener in Word (i.c. Nachziehen zu Outlook)
22.2.2025  W=0.1.1.3 O=0.1.1.3 E=0.1.1.3	Austausch der Suchen & Ersetzen-Mechanik in der Regex-Replace-Funktionalität, inkl. kleinen Anpassungen auch beim Chatbot (damit gelöschte Texte nicht mehr ersetzt werden, wie Word dies normalerweise tut); Fehlermeldungen von Bubbles werden jetzt in einen letzten Kommentar eingefügt, wenn der Benutzer dies nicht abbricht; Switch Partie schlägt neu die Regex-Replace-Methodik vor; der Regex Cap wurde auf 30k erhöht; kleinere Fehlerbehebungen
23.2.2025  E=0.1.1.4	Fehlerbehebung beim Lesen von Word-Dokumenten in Excel (API)
25.2.2025  W=0.1.1.4 O=0.1.1.4	Verbesserungen bei MarkupRegex; kleine Fehlerbehebungen; Verbesserung des Switch-Party Prompts; Unterstützung von mehr Formaten von Quellen-Angaben in den JSON-Antworten der LLM
26.2.2025  W=0.1.1.5	Fehler mit Bubbles behoben; verbesserte Implementierung von Chatbot-Befehlen
3.3.2025  W=0.1.1.6 E=0.1.1.5	Schriftgrösse im Chatbot kann mit der Maus angepasst werden; unnötige Leerzeilen bei Excel-Zellfunktionen entfernt
10.3.2025  W=0.1.1.7	Verbesserung von Context Search zum Finden einzelner Stellen, inkl. Fehlerbehebung.
2.4.2025  W=0.1.1.8 E=0.1.1.6	Markups werden nicht mehr als "Red Ink" angezeigt; Verbesserungen bei "Bubbles"; wenn der Endpoint überlastet ist (429er-Fehler) wird bis zu drei Mal mit wachsender Wartezeit wiederholt (5, 10, 30 Sekunden).
5.4.2025	Kleinere Korrekturen und Verbesserungen im Chatbot; An-

W=0.1.1.9 E=0.1.1.7 O=0.1.1.5	passung des Bubbles-Prompts für bessere Performance; grössere Prompt-Box; Umstellung des Updaters auf direkten http-Aufruf (da die vorherige Funktion nicht zuverlässig funktionierte).
7.4.2025  W=0.1.1.10 E=0.1.1.8 O=0.1.1.6	Weitere Modelle zur Auswahl bei "Freestyle (2nd)"; Einführung des "Pure:" Prefix in Word; "Ctrl-P" zum Einfügen des letzten Prompts statt Verwendung des Clipboards (in allen drei Add-ins).
7.4.2025  W=0.1.1.11 E=0.1.1.9 O=0.1.1.7	Bug-Fix betr. System-Komponente (System.ValueTuple).
7.4.2025  W=0.1.1.12	Unterstützung der Bildgenerierung (multimodale Gemini-Modelle).
8.4.2025  W=0.1.1.13 E=0.1.1.10	Bug-Fix betr. Compare Selection Halves; Autor bei Word Compare geändert; weitere Modelle und "Pure:" auch bei Excel.
14.4.2025  W=0.1.1.14	Es können alternative Stimmen und ein Prompt zum Bereinigen des Textes verwendet werden bei "Create Audio".
15.4.2025  W=0.1.1.15 E=0.1.1.12 O=0.1.1.8	"Sum-up" funktioniert jetzt auch bei mehrfach-selektierten E-Mails. Standardintervall für die Suche nach Updates auf sieben Tage aktualisiert. Fehler behoben beim automatischen Einfügen von Inhalten in Excel, wenn diese eckige Klammern enthielten.
21.4.2025  W=0.1.1.16 E=0.1.1.14 O=0.1.1.9	Chatbot für Excel; Bubbles für Excel; Transkription mit Google-Speech-to-Text (V1); direktes Einfügen beim "Clipboard"-Befehl; Markdown-Unterstützung in Word ausgebaut; kleinere Fehlerbehebungen im Chatbot für Word, im Transcriotor und an anderen Stellen.
22.4.2025  W=0.1.1.17 E=0.1.1.16	Erhöhte Geschwindigkeit der Zelldatenerfassung in Excel (Chatbot, Freestyle); Zeitlimit für Google Transcription umgangen; kleinere Fehlerbehebungen; Unterstützung der Schriftgrößenanpassung für verschiedene Fenster (nicht alle) in allen Add-Ins
23.4.2025  W=0.1.1.18 E=0.1.1.17 O=0.1.1.10	Time-out-Problem bei Google Transcription behoben; Fehler bei der Zelldatenerfassung behoben; Cuda-Unterstützung aufgrund von Inkompatibilität deaktiviert
24.4.2025  W=0.1.1.19 E=0.1.1.18	Kosmetische Anpassung Transkriptor; Fix im TTS-Formular; Fix für Einfügen von Formeln im Excel
27.4.2025  W=0.1.1.20 E=0.1.1.19 O=0.1.1.11	Einlesen von Bildern, etc. über den Trigger "(file)" (Word); kosmetische Anpassungen (Dialoge)
28.4.2025  W=0.1.1.21 E=0.1.1.21	Transcriptor kann jetzt auch Signale des Standardausgabe-geräts miterfassen; Fix bei TTS-Dialog; Fix bei Erfassung von Zellinhalten mit über 255 Zeichen Text
2.5.2025  E=0.1.1.22	Excel Freestyle unterstützt jetzt auch das Einfügen externer Dokumente über ""

03.05.2025 E=0.1.1.23	Excel Freestyle (und der Chatbot bei Auswahl) erkennt jetzt auch Dropdown-Optionen und Farbcodes
04.05.2025 E=0.1.1.24	Bug fix bei Einlesen von Zelleninhalten (Excel).
16.05.2025 W=0.1.1.26 E=0.1.1.26 O=0.1.1.13	Anbindung von externen Dienstleistungen z.B. für Rechtsinformationen wie etwa von lexisearch.ch oder DeepL (Word), Ausgabe von KI-Resultaten in einer Pane (Word), Möglichkeit unsichtbare Zeichen aus KI-Ergebnissen zu filtern ("Clean"-Parameter), einfache Selektion des gesamten Dokuments ("all", Word); Verbesserung der Konvertierung und Darstellung von Markdown-formatierten Texten; verbesserte Darstellung des Fensters beim "Clipboard"-Befehl; Bug-Fixes; Erweiterung der Prompt Library
19.5.2025 W=0.1.1.27	Bei der Datenübertragung vom Browser wird die Quelle (Word) angegeben; kleinere Fehlerbehebung
20.5.2025 W=0.1.1.28	Behebung des neuen Problems beim Einfügen von Text in Word; Fehlerbehebung für die Initialisierung von Red Ink beim Öffnen geschützter Dokumente
25.5.2025 W=0.1.1.29	Unterstützung für weitere Special Services hinzugefügt, einschliesslich Unterstützung für komplexere JSON-Templates; verbesserte und behobene Probleme beim Einfügen von Markdown-Text in Word; Hyperlinks von Perplexity usw. klickbar
27.5.2025 30.5.2025 (github) W=0.1.1.31	Unterstützung für komplexe Special Services (einschliesslich verschiedener Vorlagen), Unterstützung der Ausgabe in Panes in Excel und verbessert in Word, Unterstützung für Clipboard- und Dateiobjekte im Freestyle auch in Excel und teilweise in Outlook, Verbesserungen für Podcast und Audiobook-Funktionen, neue Transkript-Prompts, besserer Support für das Beimischen der Audio-Output-Quelle im Transcriptor, Bubbles robuster (u.a. Ignorieren von Mehrfach-Leerschlägen)
2.6.2025 W=0.1.1.32 E=0.1.1.27 O=0.1.1.14	Noch mehr Funktionalität bei Special Services (mit verschiedenen weiteren Integrationen); Outputs im Fenster können in die Pane übertragen werden (wo auch die "Merge Selection"-Funktion oder in Excel neu die selektive Anwendung von Zellbefehlen zur Verfügung steht); lokale, transparente Anonymisierungsoption (konfigurierbar); Vektorschreibe und Bag-of-Words-Suche in der Funktion "Context Search"; Standardausgabegerät kann im Transcriptor kann selektiert werden ("Dev"); Freestyle verarbeitet neu auch binäre Objekte in der Zwischenablage; der Bubble-Prompt in Word wurden optimiert; neue Parameter zur Konfiguration von API-Aufrufen (es werden jetzt auch zweifache Aufrufe und die GET-Methode unterstützt); automatische Funktion zur Generierung von "Response"-Templates; diverse Fehlerbehebungen
9.6.2025 W=0.1.1.33 E=0.1.1.28 O=0.1.1.15	Die Pane merkt sich jetzt ihre Breite; Fettdruck, Unterstreichungen und Kursivschrift wird nun erhalten, wenn kein Markup (oder höchstens DiffW) verwendet wird (kann auch ausgeschaltet werden); der Word-Helper "Clip to Text" wurde hinzugefügt; der automatische Updater wurde neu geschrieben; beim Transcriptor und Text-to-Speech-Erzeugung wird Sleep ausgeschaltet; die Text-to-Speech-Funktion unterstützt jetzt auch OpenAI (nur bei OpenAI, nicht Azure), diverse Stabilitätsmassnahmen und Fehlerbehebungen (z.B. bei der Anzeige formatierter Texte, beim Transcriptor bei Service-spezifischen

	MergePrompts; separate OAuth2-Tokens beim Transcriptor und bei Text-to-Speech)
9.6.2025 W=0.1.1.34	Beschränkung für Markdown-Konvertierung um Hänger zu vermeiden
9.6.2025 W=0.1.1.35	Bugfix betr. Umwandlung von Markdown-Formatierungen
10.6.2025 W=0.1.1.36 E=0.1.1.29 O=0.1.1.16	Fehlerbehebung bei der Verarbeitung von Daten aus der Zwischenablage (die zu einem Absturz führen konnte) und der Kontextsuche; "Bubbles" verbessert
13.6.2025 W=0.1.1.37 O=0.1.1.17	Stabilere Integration des Edge-Connectors; verbesserte Initialisierung beim Start von Word
15.6.2025 W=0.1.1.38 O=0.1.1.18	Red Ink kann Word-Kommentare neu automatisch im Dokument umsetzen; Verbesserung der Textmarkierung bei Bubbles und Context Search; Freestyle (Word) unterstützt "Add:"; Sum-up auch bei geöffneter Mail in separatem Fenster; div. Verbesserungen
15.6.2025 W=0.1.1.39	Anpassung bei der Textmarkierung für Bubbles und Context Search; Anpassung bei Apply comment (neu Markupauswahl)
15.6.2025 W=0.1.1.40	Anpassung bei der Textmarkierung für Bubbles und Context Search
17.6.2025 W=0.1.1.41	Anpassung bei der Textmarkierung für Bubbles und Context Search
19.6.2025 W=0.1.1.42	Anpassung bei der Textmarkierung für Bubbles und Context Search; Fehlerkorrekturen
22.6.2025 W=0.1.1.43 O=0.1.1.19 E=0.1.1.30	Unterstützung langerer Texte; Bugfixes der Edge-Integration; Bugfix Fehlermeldungen Freestyle; neuer Transkript-Prompt; Countdown beim Warten auf LLM; Query Assistant bei Special Services; mehr Markdown-Unterstützung in Outlook (z.B. Bullets, Tabellen)
23.6.2025 W=0.1.1.44	Bug Fix re HTML insert
27.6.2025 W=0.1.1.45	Fehlerbehebung HTML Einfügen, Debug-Hilfe hinzugefügt (Ausgabe der LLM-Antwort)
30.6.2025 W=0.1.1.46 O=0.1.1.20 E=0.1.1.31	Verbessertes HTML Einfügen, Funktion "(iterate)" für Freestyle hinzugefügt; Verbesserung der Formatierungserhaltung inkl. Bug Fix
6.7.2025 W=0.1.1.47 O=0.1.1.21	Unterstützung auch für die Bearbeitung von Texten in Fussnoten, Kopf- und Fusszeilen; Bug Fixes; Erweiterung der Parameter für Special Services; Unterstützung der Bearbeitung von Texten in der Spalte einer Tabelle; Translate in Outlook funktioniert jetzt auch bei geschlossenen E-Mails
8.7.2025	Fehlerbehebung bei der Formatierung (Word)

	<b>W=0.1.1.48</b>	
13.7.2025  W=0.1.1.49 O=0.1.1.22 E=0.1.1.31	Modelle können jetzt auch für das Senden von Multipart-Anfragen konfiguriert werden (z.B. OpenAI Bildbearbeitung); es werden jetzt Gemini-Search-Grounding-Zitate unterstützt; die Diff-Markup-Funktion wurde deutlich verbessert und schneller (Word); diverse Verbesserungen und Bug Fixes (u.a. bei Iterationen); der letzte Freestyle-Befehl kann mit einem Klick wiederholt werden (Word)	
14.7.2025 15.7.2025  W=0.1.1.51 E=0.1.1.32	Chatbot sieht immer auch die Auswahl (Word, Excel); Bug Fixes	
16.7.2025  W=0.1.1.52 O=0.1.1.23 E=0.1.1.33	Outlook unterstützt nun auch alternative Modelle; Bug Fix (Panee)	
20.7.2025  W=0.1.1.53 O=0.1.1.24 E=0.1.1.34	Neuer Markdown-Parser für Word (mehr Unterstützung, Bug Fixes); verbesserte Übersetzung/Zusammenfassung in Outlook bei mehreren geöffneten Fenstern; neue Suchkonfiguration für OpenAI; Bug Fixes	
23.7.2025 24.7.2025  W=0.1.1.54 W=0.1.1.55 O=0.1.1.25 E=0.1.1.35	KI-Output in ein neues Dokument leiten ("Newdoc:"); Bug Fixes; Erweiterung des Supports bei Special-Services (für FragenOK.ch, Entscheidsuche.ch)	
27.7.2025  W=0.1.1.56 O=0.1.1.26 E=0.1.1.36	Clipboard-to-Text auch in Outlook; Freestyle in Excel kann neu auf mehrere Arbeitsblätter zugreifen; Verbesserung der Copy-to-Clipboard-Funktionen; Bug Fixes	
28.7.2025  W=0.1.1.57	Bug Fix; Verbesserung beim Diff-Markup (Word)	
1.8.2025  W=0.1.1.58 O=0.1.1.27 E=0.1.1.37	Ausgabe formatierter Texte erneut komplett überarbeitet und auf neues System umgestellt (Word, Outlook); Excel-Chatbot unterstützt ebenfalls "(addws)"; Slides-Funktion in Word; Token-Log	
6.8.2025  W=0.1.1.59	Slides-Funktion erweitert (Grafiken, Icons); Create Audio unterstützt jetzt auch die Vertonung von Powerpoint-Folien	
10.8.2025  W=0.1.1.60	Slides-Funktion unterstützt neu Erstellen neuer Präsentationen; Create Audio erstellt neu eine Textdatei mit dem bereinigten Text; MP3-Datei wird zum Schluss neu codiert, damit die Längenangabe durchgängig stimmt	
11.8.2025  W=0.1.1.61 O=0.1.1.28 E=0.1.1.38	Slides-Funktion robuster; aktualisierter Installationsassistent; einige verbesserte Prompts	

17.8.2025  W=0.1.1.62 O=0.1.1.29 E=0.1.1.39	Prompt-Fenster neu mit Buttons für Prefixe; Bug Fixes
24.8.2025  W=0.1.1.63 O=0.1.1.30 E=0.1.1.40	Neuer Trigger "(adddoc)"; Anpassung der Prompt Library Prompts und der Prompt Library Anzeige; Einführung der Funktionen "MyStyle"; Bug Fixes (u.a. Excel Chatbot Formelumsetzung); Strg-Z in Word fasst jetzt alle Schritte zusammen (Rückgängig/Wiederherstellen mit einem Klick)
27.8.2025  O=0.1.1.32	Bug Fix (Reply)
31.8.2025  W=0.1.1.64 O=0.1.1.33 E=0.1.1.41	"Batch:" erlaubt in Excel das Abarbeiten von Textdokumenten in einem Verzeichnis; "{doc}" unterstützt in Word neu auch Powerpoint-Dateien und in Excel/Word auch OCR mittels KI; diverse Bug Fixes (Outlook Clipboard to Text, Fehler im Chatbot bei fehlendem aktiven Dokument, Fehler bei Formatierungen)
7.9.2025- 15.9.2025  W=0.1.1.71 E=0.1.1.43 O=0.1.1.50	"Document Check" erlaubt Prüfung von Word-Dokumenten nach Prüfskripten, "Red Ink Local Chat" bietet im Webbrowser einen Chatbot für alle konfigurierten KI-Modelle (läuft innerhalb des Outlook-Add-ins), Freestyle Prompts werden automatisch protokolliert, div. Bugfixes (Bullet-Ebene bei Slides, Define MyStyle-Zuordnung, "(addws)" in Excel, Timeout- und Power-handling beim Local Chatbot)
16.9.2025  W=0.1.1.72 E=0.1.1.44 O=0.1.1.52	Notice-Parameter bei DocCheck; Bug Fix (Expert Config, Umwandlung in Markdown, Resume bei Outlook)
25.9.2025- 12.10.2025  W=0.1.1.77 E=0.1.1.47 O=0.1.1.58	FindClause-Funktion (Word); WebAgent-Funktion (Word); CSV Analyzer (Excel); DocCheck erweitert (Prüfung auch von PDF und Powerpoint, Möglichkeit für zusammenfassenden Kommentar); Ausbau des Local Chatbot (zwei Threads, Anzeige für Fenced Code, Pure-Knopf, Bugfixes, verbesserte Bearbeitung von Bildern); Edge-Freestyle mit Knopf für Newdoc in Ergebnisanzeige; mehrere "{doc}"-Platzhalter im selben Prompt (Word, Excel); zusätzliche Konfiguration des ResponseKey (z.a. zur Filterung von Thinking-Outputs); Editor für Konfigurationsdateien (im Expert Mode); Explain und Process von Transkripten mit aktuellem Datum (Word); Erhalt auch von Text mit Highlighting (Word); verbesserte Interaktion mit Dokument bei Chatbot für Word; Anzeige weiterer Gründe für 429er-Fehler; Präzisere Formaterhaltung, z.B. bei Links mit überlagernden Texten (Word); div. Bugfixes (z.B. Anklicken von Links in der Pane)
12.10.2025  W=0.1.1.80 E=0.1.1.51 O=0.1.1.62	Verbesserung des automatischen Update-Handlers; Bug Fix (Start von Outlook)
14.10.2025  W=0.1.1.81 E=0.1.1.52 O=0.1.1.63	Verbesserung des automatischen und manuellen Update-Handlers und des Settings-Fensters; Verbesserung von Clipboard to Text (Outlook); Bug Fixes (Edge Translate)
15.10.2025	Benutzerparameter für Prompt-Bibliothek-Prompts hinzugefügt (Word); mehr Informationen aus pptx-Dateien auslesen

E=0.1.1.43 W=0.1.1.82	(Word); Fehlerbehebungen
16.10.2025  E=0.1.1.44 W=0.1.1.83	Verbesserte Handhabung von formatiertem Text in Freestyle (Excel)

## VII. ROADMAP

353 Weitere angedachte Entwicklungen:

- OpenAI-Modelle für die Spracherkennung/Transkription
- Pane auch in Outlook
- Möglichkeit, schnelle, günstige Modelle zu wählen für einfache Anfragen (Übersetzungen und Korrekturen)
- Automatische modellbasierte Anonymisierung
- Intelligente automatische Anwendung von benutzerdefinierten Word-Formatvorlagen
- Automatische Regex-Konvertierungen von aus Webseiten markierten Inhalten, wenn diese in Word übernommen werden, inkl. der Möglichkeit, hierfür Python-Skripte in Red Ink zu installieren
- KeepFormat auch für Freestyle in Outlook
- Unterstützung für MacOS und weitere Plattformen
- App für Mobile

## ANHANG 1: IDEEN ZUM KENNENLERNEN VON RED INK

(Diese Ideen sind für Juristinnen und Juristen verfasst; es können natürlich auch andere Texte als Verträge verwendet werden.)

- Öffnen Sie ein leeres Word-Dokument, schreiben Sie einige Wörter eines Textes und dann gehen Sie auf die rechte Maustaste oder die Kachel und wählen Sie die Übersetzungsfunktion aus und schauen Sie was passiert.
- Markieren Sie erneut Ihren Text und gehen Sie auf Freestyle. Es öffnet sich ein Textfeld zum Prompten. Dort geben Sie ein: "Mache mir daraus ein zehnzeiliges Gedicht" und klicken Sie auf OK.
- Gehen Sie in ein Dokument mit einem Vertrag und wählen Sie eine Klausel aus. Gehen Sie erneut auf Freestyle, klicken Sie dann aber ohne einen Prompt einzugeben auf OK. Es geht ein neues Fenster auf, wo Sie einen vordefinierten Prompt auswählen können. Wählen Sie den ersten aus, den "Challenger", und schauen Sie, was geschieht.
- Wenn Sie ein Vertragsdokument haben, dann öffnen Sie das Dokument und markieren Sie das gesamte Dokument. Dann Freestyle wählen und den Prompt eingeben: "Clipboard: Was sagt die Kündigungsklausel?" und die Eingabetaste drücken. Der Vorsatz "Clipboard:" bedeutet, dass die Antwort in einem Fenster und in der Zwischenablage landet statt im Dokument.
- Gehen Sie nun an den Anfang des Dokuments und rufen Sie "Context Search" auf und geben Sie einen Suchbegriff zu einem Thema ein, zu dem es eine Klausel gibt, die aber den Suchbegriff so nicht enthält, z.B. "Schadenersatz", wenn in der Klausel nur von "Haftung" die Rede ist. Lassen sich nach dem Kontext "Schadenersatz" suchen – und er sollte Ihnen die erste Textstelle zeigen, die damit zu tun hat. Probieren Sie es nochmals, aber fügen Sie dem Suchbegriff ein "(m)" (für alle Treffer = Multiple) und ein "(all)" für Suche im ganzen Dokument an. Die Stellen werden Ihnen gelb markiert angezeigt.
- Nun markieren Sie den ganzen Text oder gar nichts und rufen Freestyle auf. Dort geben Sie ein "Bubbles: Welche der Klauseln enthalten unklare Regelungen und warum?". Nun sollte Ihnen das Tool nach einer gewissen Verarbeitungszeit alle Klauseln mit einem Kommentar versehen haben, der die Frage beantwortet.

- Passen Sie ein, zwei Klauseln im "Änderungen verfolgen"-Modus inhaltlich an, markieren dann den ganzen Text und geben in Freestyle den Prompt "Clipboard: Erläutere mir die Änderungen, die am Text vorgenommen wurden. (rev)" Es wird Ihnen diese zusammenfassen.
- Im selben Dokument den ganzen Text nochmals markieren und Summarize wählen. Im Fenster, das aufgeht, die Zahl bestätigen oder z.B. 200 eingeben und OK drücken. Am Ende des Vertrags erscheint eine Zusammenfassung.
- Dann kopieren Sie aus Outlook den Inhalt einer Mail oder sonst einem Dokument eine Passage eines Textes in ein neues Dokument, der auch Namen und Firmenbezeichnungen enthält. Selektieren Sie den Text und wählen Sie auf der Red Ink-Kachel Anonymize.

ACHTUNG: Jetzt wird je nach Einstellung die Compare-Funktion von Word aktiviert um ein Markup zu erstellen. Dabei gehen diverse Fenster auf und wieder zu. Es sollte danach ein anonymisierter Markup Ihres Textes erscheinen.

- Jetzt öffnen Sie in Outlook eine längere E-Mail-Kette. Klicken Sie auf "Forward" oder "Reply" und schauen Sie, dass die Mail zum Schreiben in einem eigenen Fenster geöffnet ist. Dort sollte im Menüband das Red Ink Logo erscheinen und daneben eine weitere Kachel mit drei Buttons (falls nur ein Kreis erscheint, dann machen Sie das Fenster breiter).
- Wählen Sie aber noch nichts an. Klicken Sie einfach auf den Knopf "Sum-up" und warten Sie. In Kürze erscheint eine Zusammenfassung der E-Mail-Chain.
- Dann löschen Sie diese Zusammenfassung und verfassen Sie irgendeinen Satz mit zwei, drei Sätzen mit Schreibfehlern oder Wörtern zuviel drin. Selektieren Sie den Text und wählen Sie "Correct" an. Ihr Text erscheint in korrigierter Form. Optional lässt sich noch ein Markup dazuschalten, aber der braucht etwas Zeit (hierzu Settings benutzen, dann "Save Configuration"). Dauert der Markup mit "Diff" zu lange, kann er mit "Esc" einfach abgebrochen werden. Für grössere Texte ist die Word-Markup-Methode besser geeignet. Ansonsten empfehlen wir "DiffW".
- Zum Schluss gehen Sie zurück in Word und öffnen dort ein Dokument und den Chatbot. Nun bitten Sie den Chatbot, in Ihrem Dokument etwas zu ändern. Dazu muss "Grant write access" angewählt sein und das Kästchen rechts daneben ebenfalls, um dem Chatbot das ganze Dokument zugänglich zu machen. Beispiel: "Ergänze mir bitte den zweiten Absatz mit einem weiteren Gedankengang, der zum Thema passt."



## ANHANG 2: PROGRAMMIEREN VON RESPONSE-TEMPLATES

*Hinweis: Eine automatische Programmierung von Templates basierend auf einem Muster-JSON-String und einer Beschreibung des Outputs ist mit dem Freestyle-Befehl "generateresponsetemplate" möglich.*

### Überblick

Templates ermöglichen es, JSON-Daten in Textdokumente zu überführen. Dabei werden Platzhalter und spezielle Anweisungen verwendet, um Werte aus verschiedenen Pfaden im JSON-Dokument anzuzeigen, Listen zu durchlaufen und Inhalte zu formatieren.

### Grundlegende Platzhalter

Ein Platzhalter ist ein Textmuster in geschweiften Klammern, das durch einen JSON-Wert ersetzt wird. Syntax: {Pfad}

- Pfade folgen dem JSONPath-Konzept. Beispiele:
  - Einfacher Feldzugriff: {benutzer.name} greift auf das Feld 'name' des Objekts 'benutzer' zu.
  - Zugriff auf verschachtelte Felder: {bestellung.adresse.ort} greift auf 'ort' in 'adresse' der 'bestellung' zu.
  - Pfade können mit oder ohne führendes '\$.' angegeben werden. Beide sind äquivalent:
    - {\$ .benutzer.alter} oder {benutzer.alter}

Wird der Pfad nicht gefunden, bleibt die Ausgabe leer.

### JSONPath-Grundlagen

JSONPath ist eine Möglichkeit, gezielt Werte aus einem JSON-Dokument auszuwählen. Einige wichtige Regeln:

- Objekte werden durch ihren Namen adressiert: 'objektName.feldName'.
- Arrays werden durch eckige Klammern und Index oder Filter angesprochen: 'artikel[0]' für das erste Element.
- Platzhalter in Templates unterstützen einfache Pfade und Array-Durchläufe (siehe Kapitel 4).

Beispiele:

- {produkte[0].preis} zeigt den Preis des ersten Produkts.
- {kategorie.name} zeigt den Namen der Kategorie.

### Schleifen für Listen

Mit Schleifen kann man wiederholt dasselbe Template-Segment für alle Elemente einer Liste ausgeben.

Syntax:

```
{% for listenPfad %}
```

Inhalt mit Platzhaltern, z. B. {feld1}, {feld2} usw.

```
{% endfor %}
```

Erklärung:

- 'listenPfad' ist ein Pfad, der auf ein Array oder ein Objekt verweist.
- Innerhalb einer Schleife wird für jedes Element der angegebene Abschnitt ersetzt. Dabei beziehen sich Platzhalter innerhalb auf das aktuelle Element.
- Beispiel: {% for artikel %}Artikel: {name} - Preis: {preis}{% endfor %} durchläuft das Array unter 'artikel' und gibt für jeden Eintrag Name und Preis aus.
- Wird unter dem Pfad ein einzelnes Objekt gefunden, wird ebenfalls einmal ausgegeben.

### **Modifikatoren für Platzhalter**

Platzhalter können durch Präfixe angepasst werden:

- html: Wandelt HTML-Inhalte in Markdown um. Beispiel: {html:beschreibung}.
- nocr: Entfernt alle Zeilenumbrüche und ersetzt sie durch Leerzeichen. Beispiel: {nocr:kommentar}.
- htmlnocr: Kombiniert beide Effekte. Beispiel: {htmlnocr:beschreibung}.

Die Reihenfolge spielt keine Rolle, Groß-/Kleinschreibung wird ignoriert.

### **Separatoren festlegen**

Für Pfade, die mehrere Werte zurückliefern (z. B. mehrere gleichnamige Felder), kann ein eigener Trenner angegeben werden. Syntax: {Pfad|Trenner}

Beispiel:

- {tags|; } verbindet mehrere Schlagwörter mit einem Semikolon und Leerzeichen.
- {produkte[?(@.verfügbar==true)].name|, } verbindet alle Namen verfügbarer Produkte mit Komma und Leerzeichen.

Standardmäßig werden Werte durch einen Zeilenumbruch getrennt.

### **Mapping-Definitionen**

Mit einem Mapping kann man bestimmte Ausgabewerte ersetzen.  
Syntax: {Pfad|schlüssel1=Text1;schlüssel2=Text2;...}

Erklärung:

- Jeder Schlüssel = Originalwert aus JSON, der durch Text1, Text2 etc. ersetzt wird.
- Beispiel: {status|0=Inaktiv;1=Aktiv;2=Gesperrt} ersetzt 0 durch 'Inaktiv', 1 durch 'Aktiv', 2 durch 'Gesperrt'.
- Werden keine Mapping-Regeln angewendet oder der JSON-Wert passt zu keinem Schlüssel, wird der Originalwert ausgegeben.

## **Zeilenumbrüche und Sonderzeichen**

Um innerhalb des Templates feste Zeilenumbrüche zu setzen, können folgende Platzhalter genutzt werden:

- \N fügt einen Zeilenumbruch an dieser Stelle ein.
- \n oder \r oder \R funktioniert ebenso.
- <cr> (unabhängig von Groß-/Kleinschreibung) wird durch einen Zeilenumbruch ersetzt.

Beispiel:

Text vor Zeilenumbruch \N Text nach Zeilenumbruch.

## **Beispiele**

### **Beispiel 1:** Einfacher Platzhalter

Wenn das JSON-Objekt einen Schlüssel 'benutzer' mit Feld 'name' enthält, kommt das ins Template:

{benutzer.name}

und erhält nur den Namen.

### **Beispiel 2:** Liste von Einträgen mit Schleife und Separator

{% for artikel %}

Artikel: {name} - Preis: {preis}

{% endfor %}

Falls es mehrere Artikel gibt, wird jeder in einer neuen Zeile ausgegeben. Soll ein Komma als Trennzeichen zwischen Artikeln verwendet werden, wird z.B. genutzt:

{% for artikel %}{name} - {preis}{% if not for\_last %}, {% endif %}{% endfor %}

(Hierbei existiert implizit ein Platzhalter 'for\_last', der bei letzter Iteration wahr ist.)

### **Beispiel 3:** Mapping mit benutzerdefinierten Texten

Angenommen, das JSON-Feld 'status' kann Werte 0, 1 oder 2 haben, kommt das ins Template:

{status|0=Inaktiv;1=Aktiv;2=Gesperrt}

und erhält je nach Status den entsprechenden Text.

## **ANHANG 3: WEB AGENT SCRIPTING**

*(nur auf Englisch verfügbar)*

The Web Agent executes declarative JSON scripts to retrieve and analyze web pages (HTTP requests + HTML parsing), extract data, save files, send emails, embed LLM evaluations, and render reports as Markdown – without browser automation (pure HTTP + HtmlAgilityPack). It supports variables, secret management, templating, control structures (if / foreach), retry and error strategies, as well as simple Mustache-like placeholders.

The scripts can be created manually or by using a built in function of Red Ink that can be provided with natural language instructions.

### **Introduction: Script structure and rules**

- Top-level JSON object:
  - meta: optional
    - default\_timeout\_ms: integer. Default step timeout if the step doesn't override it.
    - user\_agent: string. Default HTTP User-Agent applied globally unless changed later.
  - env: optional
    - base\_url: string. Used as a base for resolving relative links when no absolute URL and there is no last response URL.
    - headers: object. Default headers applied to every request (until changed).
    - secrets: object. Key/value secrets. Values are also exposed into variables under the same key. Note: values are read as-is; if you use secret://name it resolves only if name already exists among secrets.
    - variables: object. Arbitrary variables available to templating and conditions.
  - steps: array of step objects executed sequentially (with optional jumps via on\_error.goto or guard.else\_goto).
- Each step object supports:
  - id: string. A unique identifier you can jump to (via on\_error.goto or guard.else\_goto).
  - command: string. One of the commands listed in the Command Reference below.
  - timeout\_ms: integer. Per-step timeout (otherwise meta.default\_timeout\_ms applies).
  - retry: object. Optional retry policy for the step:

- max: integer. Number of retries (0 = no retry).
- delay\_ms: integer. First delay before retry (defaults vary by command).
- backoff: number. Multiplier for each subsequent retry delay (e.g., 2.0 doubles each time).
- on\_error: object. Optional error handling after retries exhausted:
  - action: "continue" | "goto" | "abort" | "retry"
  - goto: string. Target step id (required when action = "goto").
- guard: object. Optional precondition:
  - if: string condition expression (see Conditions below). If false, the step is skipped.
  - else\_goto: string. Optional alternate step id when guard fails.
- wait\_for: object. Optional waits:
  - type: "time" | "url" | "selector"
    - time: { timeout\_ms: integer } pre-delay before executing the command.
    - url: { value: string } post-check: warns if current URL does not contain value.
    - selector: { selector: SelectorObject } post-check: warns if selector not found.
- params: object. Parameters for the command (shape depends on the command).
- assign: object. Store a step's result/part of result into a variable:
  - var: string. Name of variable to set.
  - path: string. Optional JSONPath-like token path (SelectToken semantics) to select a sub-value from the command result before assignment.

### **Selectors (used by extract\_\* and wait\_for.selector)**

- SelectorObject:
  - strategy: "xpath" | "css" | "text" | "regex"
  - value: string. Meaning depends on strategy:
    - xpath: an XPath
    - css: a simple CSS selector (internally converted to XPath; supports tag, #id, .class, [attr] and :nth-child(n))

- text: match normalized innerText. Use "exact:..." for exact-match; otherwise contains.
- regex: regex applied to normalized innerText
- within: SelectorObject (optional). Limits the search scope to results of this nested selector.
- relative: object (optional). Narrow matches:
  - position: "first" | "last" | "nth"
  - nth: integer (1-based) when position = "nth"

### **Templating and variables**

- All string parameters (URLs, headers, bodies, etc.) support template expansion with `{...}`:
- `{varName}` reads from env.variables or previously set variables (including secrets promoted to variables).
- `{variables.X}` also reads X from the variables bag.
- `{env.NAME}` reads OS environment variables (and special env.DESKTOP for Desktop path).
- `{base_url}` reads env.base\_url.
- Unresolved placeholders are left intact (e.g., "`{missing}`") and logged; some commands (like open\_url) will error if critical fields remain unresolved.
- The template and render\_report commands support Mustache-like sections: `{#name}...{/name}` for truthy/iteration, `{^name}...{/name}` for inverted, triple `{var}` for raw.

### **Conditions (guard.if and if.condition)**

- Supported forms:
  - OR: expr1 || expr2
  - exists `{path}` → true if resolved value is non-empty
  - `{path} == "value"` → case-insensitive equality
  - `{path} contains "substring"` → substring match (case-insensitive)
  - `{path} =~ "regex"` → regex match (singleline, ignorecase)
  - `{path} == []` → true if value is null/empty or empty enumerable
- The `{path}` inside conditions resolves the same way as templating.

### **Built-in and auto variables (set by the engine)**

- After HTTP commands: last\_http\_status (int), last\_http\_elapsed\_ms (ms)

- After open\_url/http\_request: internal current page: URL and HTML are tracked; selectors operate on it.
- After LLM: lastLlm (object), lastLlm\_page\_url (string), lastLlm\_raw (string), lastLlm\_latency\_ms (ms)
- Other: last\_step\_id (string), auto\_links (list of URLs) when auto-linking is enabled

## **Command Reference (all commands and their parameters)**

- Notes:
  - Unless stated otherwise, step-level timeout\_ms, retry, guard, wait\_for, assign are available.
  - “Required” parameters are marked with (required). All values support templating unless noted.

### **1. set\_user\_agent**

- Purpose: Override the HTTP User-Agent globally.
- params:
  - user\_agent (required): string
- Returns: { user\_agent }
- Side effects: Updates default headers of the HttpClient.

### **2. set\_headers**

- Purpose: Add/replace default headers for subsequent HTTP requests.
- params:
  - mode: "replace" | (default append)
  - headers (required): object of headerName → headerValue
- Returns: { headers: currentDefaults }

### **3. set\_cookies**

- Purpose: Add cookies to the internal CookieContainer.
- params:
  - cookies (required): array of:
    - name (required)
    - value (required)
    - domain (required)
    - path: string (default "/")
    - secure: bool
    - httpOnly: bool
  - Returns: { count }
  - Notes: Cookie is added without making a request.

### **4. open\_url**

- Purpose: Load a page (and parse HTML for selectors).
- params:

- url (required): string. Must be absolute or resolvable via last URL or base\_url. Must not contain unresolved {{...}}.
- method: "GET" (default) | "POST" | ...
- headers: object (per-request headers)
- body: any JSON/string (request body)
- body\_type: "json" | "form" | other (raw)
- timeout\_ms: integer (optional; step-level also exists)
- return\_body: bool (default false) include "body" in result
- retry: { max, delay\_ms, backoff } (optional, overrides step retry)
- Returns: { status, url, elapsed\_ms, body? }
- Side effects: Updates current document/URL; auto-extracts links (see auto-links below).
- Errors: Transient HTTP errors (e.g., 408/429/5xx) use retry policy.

## 5. **http\_request**

- Purpose: Make an HTTP call; also updates current document and URL.
- params:
  - url (required): absolute URL
  - method: "GET" (default) | others
  - headers: object
  - query: object (key-value querystring appended)
  - body: any JSON/string
  - body\_type: "json" | "form" | other (raw)
- Returns: { status, headers (string), body (string), url }
- Side effects: Updates current document/URL.

## 6. **download\_url**

- Purpose: Download a URL to a file.
- params:
  - url (required)
  - target\_dir (required)
  - filename: string (default "download.bin")
  - method: "GET" (default) | others
  - headers: object
  - body: any
  - body\_type: "json" | "form" | other
- Returns: { path, status }

## 7. **save\_file**

- Purpose: Save content to disk.
- params:
  - path (required)
  - content (required)
  - encoding: "binary" (content is base64) | default UTF-8 text
- Returns: { path }

## **8. `read_file`**

- Purpose: Read a file.
- params:
  - path (required)
  - encoding: "binary" (returns base64) | default UTF-8 text
- Returns: string (text or base64)

## **9. `wait`**

- Purpose: Sleep for N milliseconds.
- params:
  - ms: integer (default 0)
- Returns: { slept }

## **10. `find`**

- Purpose: Case-insensitive substring search inside a variable.
- params:
  - in (required): string var name (haystack)
  - text (required): string (needle)
  - assign: { var: string } (optional; sets the var to true/false)
- Returns: { found: bool, index: int } (index = first match or -1)

## **11. `extract_text`**

- Purpose: Extract normalized text from selector matches.
- params:
  - selector (required): SelectorObject
  - all: bool (default false). If false returns first match; if true returns array of strings.
  - normalize\_whitespace: bool (default true)
  - regex: string (optional post-filter applied to text)
  - group: int (regex group index, default 0)
- Returns: string or array<string>

## **12. `extract_html`**

- Purpose: Extract HTML fragment for first node.

- params:
  - selector (required): SelectorObject
  - outer: bool (default false). true = outerHtml; false = innerHtml
- Returns: string (empty string if no match)

### **13. extract\_attribute**

- Purpose: Extract attribute value(s) from a previously stored node list.
- params:
  - nodes\_var (required): variable that holds a list of serialized nodes
  - attribute (required): attribute name, e.g. "href"
  - var (required): variable to store the resulting list of strings
- Returns: null; side effect: sets target var to array<string>
- Note: Nodes must be stored as serializable objects with "attributes" previously; if you don't have such a list, consider extracting URLs via page parsing or auto\_links.

### **14. set\_var**

- Purpose: Set a variable to a value (string expanded via templating or raw JSON).
- params:
  - name (required): string
  - value: any JSON; if string, templating is applied
- Returns: { name, value }

### **15. template**

- Purpose: Render a string using a lightweight Mustache (sections, inverted, triple braces).
- params:
  - template (required): string
  - context: object (used for {{}} resolution inside the template; supports nested JSON)
- Returns: rendered string
- Notes: After Mustache render, a second pass of {{...}} expansion runs against global variables.

### **16. render\_report**

- Purpose: Render final Markdown (and optionally write it to disk).
- params:
  - engine: string (reserved; currently not branching behavior)
  - template (required): string (Markdown template)

- context: object (see template)
- output\_path: string (optional). If provided and free of unresolved {{...}}, writes to this path.
- Returns: { output: "(memory)" | path }
- Side effects: Sets the interpreter's final Markdown output.

## **17. delete\_file**

- Purpose: Delete a file if it exists.
- params:
  - path (required)
- Returns: true/false (success)

## **18. send\_email\_report**

- Purpose: Send a multipart/alternative email (text+HTML), converting Markdown to HTML if needed and adding a small footer.
- params (all support templating):
  - to (required): string of addresses separated by "," or ";"
  - subject: string (default "Report")
  - smtp\_host (required): string
  - smtp\_port: int (default 25)
  - smtp\_ssl: "true"/"false" (bool-like)
  - smtp\_auth: "true"/"false" (bool-like)
  - smtp\_user: string
  - smtp\_pass: string
  - from\_email: string (default "noreply@noreply.com")
  - from\_name: string (default "Red Ink WebAgent")
  - body\_markdown: string. If looks like HTML (<html>...), used as-is; otherwise converted from Markdown.
  - ip\_override or ip: string (optional; for footer)
  - helo\_domain: string (optional; best-effort HELO domain override)
  - net: string (optional; network/domain label for footer)
- Returns: true/false (sent/not sent)
- Side effects: Adds a footer indicating agent, IP, and domain.

## **19. log**

- Purpose: Append a line to the interpreter log.
- params:
  - level: string (free form; e.g., "info", "warn")
  - message: string

- Returns: null

## **20. if**

- Purpose: Conditional execution of nested steps.
- params:
  - condition: string (see Conditions)
  - steps: array of step objects (executed if condition true)
  - else\_steps: array of step objects (optional; executed if condition false)
- Returns: null

## **21. foreach**

- Purpose: Loop over a list or single value and run nested steps.
- params:
  - list (required): name of variable containing IEnumerable/array (or a single object; it will iterate once)
  - item\_var (required): name of variable to assign current item to
  - steps (required): array of step objects to execute for each item
  - continue\_on\_error: bool (default true unless stop\_on\_error = true)
  - stop\_on\_error: bool (default false; if true, overrides continue\_on\_error)
  - max\_items: int (optional cap)
  - break\_if\_var\_true: string (variable name). If that variable becomes truthy, the loop breaks after the current item.
- Returns: { count: iterated, executed: successfulIterations }

## **22. array\_push**

- Purpose: Push an item into an array variable (creating it if needed).
- params:
  - array (required): array variable name
  - item\_var: variable name to copy the item from (preferred)
  - item: inline JSON value (used if item\_var not provided)
- Returns: { pushed: bool, count: int, array: string }

## **23. enable\_dynamic**

- Purpose: Enable dynamic expansion of page content after load (best-effort fetch of script srcs and inline-discovered AJAX URLs and append their responses).

- params: none
- Returns: { "status": "ok", "dynamic": true }
- Notes: Fetch count is capped; intended for pages that assemble content through auxiliary endpoints. Not all dynamic sites can be reconstructed via static HTTP.

## 24. llm\_analyze (aliases: llm, llmanalyze)

- Purpose: Call the configured LLM to analyze or transform content. Includes robust JSON extraction/sanitization and validation.
- params (all optional unless stated):
  - system / systemPrompt: string. System prompt.
  - user / prompt / input / arguments: string. User prompt.
  - temperature: string/number. If omitted, uses configured default.
  - timeoutMs: integer. Overrides step timeout for this call.
  - inner\_attempts: int (>=1). If non-JSON/plaintext is returned, attempts quick retries inside the step (default 1).
  - inner\_delay\_ms: int. Delay between inner attempts (default 800 ms).
  - status\_var: string. If that variable equals "404" and allow\_llm\_on\_404 is not set, the call is skipped.
  - Validation and acceptance:
    - retry\_on\_invalid: bool. Throw on invalid output so outer step retry can kick in.
    - require\_key: "k1,k2,...". Comma-separated required top-level JSON keys.
    - require\_array\_key: "arrKey1,arrKey2,...". Comma-separated required array keys.
    - require\_min\_items: int. If require\_array\_key set, those arrays must have at least this many items.
    - require\_key\_all: bool (default true). If true, all listed required keys must exist.
    - reject\_if\_empty: bool. Fail if (sanitized) output is empty.
    - reject\_if\_plaintext: bool (default true). Fail if output is not valid JSON. Set allow\_non\_json to override.
    - allow\_non\_json: bool. If true, accept non-JSON (turns off reject\_if\_plaintext).
  - Logging/UX:

- log\_raw: bool. If debug enabled, writes trimmed raw model output to the debug log.
- max\_preview: int (default 250). Length shown in on-screen preview.
- Returns: object:
  - If valid JSON object: the object, plus injected fields: step\_id, page\_url; and possibly \_invalid/\_reason if validation failed but allowed.
  - If non-object JSON: wrapped into { value: <stringified>, step\_id, page\_url, \_invalid? }.
  - If non-JSON and allowed: wrapped invalid object with metadata.
  - Side effects: Sets variables lastLlm, lastLlm\_page\_url, lastLlm\_raw, lastLlm\_latency\_ms.
  - JSON extraction/sanitization behavior:
    - Prefers JSON inside fenced code blocks; otherwise extracts the first balanced {} or [] from the text; strips fences; trims.

## 25. Unsupported alias

- navigate: not available in HTTP mode. Use open\_url.

## Step/global flags (set via env.variables or set\_var)

- Debugging:
  - debug: bool. Enable step-level snapshots and request dumps.
  - debug\_allAttempts: bool. Log each attempt, not just final.
  - debug\_substeps: bool. Log nested steps inside if/foreach.
  - debug\_var\_changes: bool. Log variable changes.
  - debug\_include\_script: bool. Dump the (masked) script JSON and a step overview to RI\_Debug\_Webagent.txt on Desktop (or app base).
  - debug\_summary: bool. Append final summary to debug log.
  - debug\_clear\_llm\_state: bool. Clear lastLlm/lastLlm\_page\_url before non-LLM steps.
  - llm\_rethrow\_all: bool. Rethrow LLM exceptions (by-pass wrapping).
- LLM flow control:
  - allow\_llm\_on\_404: bool. Permit LLM step even if status\_var is "404".
  - allow\_llm\_invalid: bool. Accept invalid LLM output in the step even when retry\_on\_invalid is set.
  - continue\_on\_llm\_timeout: bool. Inside foreach, tolerate timeouts as soft failures and continue.

- Auto link extraction after page load:
  - auto\_link\_enable: bool (default true). If false, disables auto link collection.
  - auto\_link\_patterns: string or array<string> regexes. Default pattern matches common detail/download/decision links.
  - auto\_link\_min: int. Min href length (default 15).
  - Output variable: auto\_links (array<string> absolute URLs).

### **Additional important aspects**

- Execution order: Steps run in order unless redirected by on\_error.goto or guard.else\_goto. A skipped step (guard false) is considered successful and can branch via else\_goto.
- Assign semantics: assign.var stores the command result; assign.path (SelectToken) lets you store a sub-value (e.g., "\$.items[0].url"). If the path does not resolve, the variable is set to null.
- URL resolution: Relative URLs are resolved against the last response URL; if not available, against env.base\_url; sanitizePotentialMarkdownUrl allows pasting Markdown links by extracting the target inside parentheses; absolute URLs must start with http/https.
- Timeouts and retries:
  - step.timeout\_ms applies to the command (some commands also accept params.timeout overrides).
  - retry applies per step; transient HTTP statuses (408, 425, 429, 500, 502, 503, 504) are considered retryable in open\_url.
- Dynamic expansion: enable\_dynamic appends fetched script and AJAX-like responses to the current HTML and reparses; capped to a small number of fetches; intended as a best-effort static reconstruction (not a full browser).
- HTML normalization: For extract\_text and extract\_\* selectors, innerText is normalized (collapsing whitespace) when normalize\_whitespace = true.
- Email sending: The agent builds proper multipart/alternative (plain + HTML). If body\_markdown looks like full HTML, it won't be converted. A footer indicating the agent, local IP/domain is appended.
- Security: Secrets provided in env.secrets are masked in debug logs. Avoid logging sensitive data manually in log steps. Never hardcode credentials—prefer secrets.
- Unsupported features: There is no browser/DOM execution, no JS execution, and no CSS/visual layout. navigate is not supported.
- Error handling policy: If on\_error is absent and the command fails after retries:

- If failHard is false (current build), the interpreter logs the error; on some commands it may continue; but many errors will still bubble and stop execution unless action=continue/goto is specified. Prefer explicit on\_error for robust flows.
- Return value: The final result of the whole run is either the last render\_report's Markdown or a Markdown-wrapped log when no report was produced.

## User-specific parameters (entered at runtime)

### Overview

- Parameter DEFINITIONS embed inline in the script: {parameterN=...definition...}
- Parameter REFERENCES elsewhere: {parameterN}
- First definition wins per N; duplicates with same N are ignored.
- Prompt order = ascending N (1,2,3,...).
- Replacement order = from end to start (reverse indices) to keep positions stable.
- On Cancel → processing returns False; callers should abort execution.

### Regex matches (whitespace tolerant)

- Definition regex: { parameter(\d+)=(.\*) } (singleline)
- Reference regex: { parameter(\d+) }

### Definition syntax (semicolon-separated segments)

- {parameterN=Description ; Type ; DefaultValue ; RangeOrOptions ; ExtraOptions }
- Minimal form: {parameter1=Choose item} (Type defaults to string, default empty)

### Segments

- [0] Description (mandatory; shown in UI)
- [1] Type (optional): string | integer | long | double | boolean (case-insensitive; default:string)
- [2] DefaultValue (optional; interpreted per type)
- [3] Either a numeric range MIN-MAX (integers only) OR a comma list of options
- [4] If [3] is a range and [4] present → treated as an additional comma-separated options list

### Options & Mapping

- Options: LabelA<codeA>, LabelB<codeB>, ... (if <...> missing, label=code)
- UI shows labels; after submit, labels map back to their code for substitution.
- If DefaultValue matches a code, corresponding label is pre-selected.

## Type handling

- boolean: Boolean.TryParse; output lowercased true/false.
- integer/long/double: parsed numerically.
  - Range pattern:  $^{\wedge} \d+ \s* - \s* \d+ \$$  → clamp to inclusive [min,max].
  - For integer/long: value rounded (Math.Round) then cast to integral.
- string: raw string (after optional display→code mapping).

## Special/empty selection

- If the chosen value (case-insensitive) starts with "(keine auswahl)" or "(no selection)", or is exactly "----", the substituted value becomes an empty string.

## JSON Escaping

- Only backslash (\ → \\) and double quote (" → \") are escaped before insertion.
- - No further normalization; ensure resulting JSON stays valid where substituted (e.g., surround with quotes if a JSON string is expected).

## Replacement behaviour

- Definitions are replaced in place (the entire {parameterN=...} becomes the final value).
- References {parameterN} are replaced with the same resolved value if the definition existed.
- References to undefined N are left unchanged.

## No-definition case

- If no {parameterN=...} is present: returns True immediately; no prompt; references remain literal.

## Examples

- {parameter1=API environment ; string ; prod ; prod<https://api.prod.example>, staging<https://api.staging.example>, dev<http://localhost:8080>}
- {parameter2=Max retries ; integer ; 3 ; 0-10}
- {parameter3=Confidence threshold ; double ; 0.65 ; 0-1 ; 0.25,0.5,0.65,0.75,0.9}
- {parameter4=Enable verbose logging ; boolean ; true}
- {parameter5=Output type ; string ; json ; json<application/json>,xml<application/xml>}

## Integration notes

- Prefer defining parameters in fields that will become the final literal value (to keep JSON valid after replacement).
- Example (recommended pattern): "base\_url": "{parameter1=Base URL ; string ; https://api.example.com}"

- Example (not recommended): "url": "{parameter1=https://api.example.com ; string ; https://api.example.com}/v1/items"  
(because the definition collapses to a raw URL; better define once and reference its value elsewhere).

### Edge cases & safeguards

- Whitespace around semicolons ignored; extra segments beyond defined semantics are ignored.
- If range is provided but input is not numeric, original string is left (then possibly empty if sentinel chosen).
- Multiple identical options allowed; first matching code resolves default.
- Duplicated definitions with same N: only the first is used; later ones become inert text and are overwritten by reverse-order replacement.

### Quick templates

- Integer with range: {parameter1=Retry count ; integer ; 3 ; 0-10}
- Double with options: {parameter2=Threshold ; double ; 0.75 ; 0.25,0.5,0.75,0.9}
- Enum string: {parameter3=Mode ; string ; safe ; safe<safe>,fast<fast>,audit<audit>}
- Boolean: {parameter4=Enable cache ; boolean ; false}

### Minimal example

```
{
 "meta": { "default_timeout_ms": 15000, "user_agent": "My-Agent/1.0" },
 "env": {
 "base_url": "https://example.com",
 "headers": { "Accept": "text/html" },
 "variables": { "q": "contract" }
 },
 "steps": [
 { "id": "load", "command": "open_url",
 "params": { "url": "{{base_url}}/search?q={{q}}" },
 "retry": { "max": 2, "delay_ms": 1000, "backoff": 2.0 }
 },
 { "id": "title", "command": "extract_text",
 "params": { "selector": { "strategy": "css", "value": "h1" } },
 "assign": { "var": "page_title" }
 },
 { "id": "report", "command": "render_report",
 "params": { "template": "# Title\n\n{{page_title}}" }
 }
]
}
```

## Practical example

```
{
 "meta": {
 "user_agent": "RedInk-WebAgent/1.0",
 "default_timeout_ms": 100000
 },
 "env": {
 "variables": {
 "threshold_date": "{parameter1 = Publishing date; string; >= 07.10.2025}",
 "summary_list": "",
 "summary_array": [],
 "debug": false
 }
 },
 "steps": [
 { "id": "tune_llm_tolerance", "command": "set_var", "params": {"name": "continue_on_llm_timeout", "value": true} },
 { "id": "tune_llm_invalid", "command": "set_var", "params": {"name": "allow_llm_invalid", "value": true} },
 {
 "id": "open_index",
 "command": "open_url",
 "params": {
 "url": "https://search.bger.ch/ext/eurospider/live/de/php/aza/http/index_aza.php?lang=de&mode=index&search=false",
 "return_body": true
 },
 "assign": { "var": "index_body_raw", "path": "body" }
 },
 {
 "id": "find_date_links_llm",
 "command": "llm_analyze",
 "params": {
 "system": "You are a deterministic HTML link extractor. You receive raw HTML that contains anchor (<a>) elements whose visible text contains a date in the format dd.mm.yyyy optionally followed by descriptive text (e.g. '22.02.2024 22 Entscheide'). Extract ONLY the href values of those links whose date is {{threshold_date}} (format dd.mm.yyyy). Normalize relative URLs to absolute using the page URL if necessary (omit fragments). Return ONLY valid JSON of the form {\\"links\\": [\"url1\", \"url2\"]}. If none, return {\\"links\\\": []}. No extra text, no fences."
 },
 "user": "HTML:\n{{index_body_raw}}",
 "retry": { "max": 2, "delay_ms": 3000, "backoff": 2.5 },
 "retry_on_invalid": true,
 "require_key": "links",
 "temperature": 0
 },
 "assign": { "var": "date_links", "path": "links" }
],
 {
 "id": "ensure_date_links_array",
 "command": "arrayify"
 }
}
```

```

"command": "if",
"params": {
 "condition": "exists {{date_links}}",
 "else_steps": [
 {
 "id": "init_empty_date_links",
 "command": "set_var",
 "params": { "name": "date_links", "value": [] }
 }
]
},
{
 "id": "loop_date_pages",
 "command": "foreach",
 "params": {
 "list": "date_links",
 "item_var": "date_page_url",
 "steps": [
 {
 "id": "open_date_page",
 "command": "open_url",
 "params": {
 "url": "{{date_page_url}}",
 "return_body": true
 },
 "assign": { "var": "date_page_html", "path": "body" }
 },
 {
 "id": "extract_decision_links",
 "command": "lxml_analyze",
 "params": {
 "system": "You are a deterministic HTML decision link extractor. You receive raw HTML of a daily decisions page of the Swiss Federal Supreme Court. Each genuine decision row is a table (or table-like) row where: (1) the first column contains a date in the format dd.mm.yyyy; (2) the second column contains an <a> anchor whose visible text is the official case / docket identifier. Some rows may be headers, pagination, navigation, empty, or not real decisions—ignore those. REQUIREMENTS: 1) For every genuine decision row, extract an object {{\"date\": \"dd.mm.yyyy\", \"id\": \"CASE_ID\", \"url\": \"ABSOLUTE_URL\"}}. 2) Preserve order of appearance (top to bottom). 3) De-duplicate strictly by (date,id,url) first; if duplicates differ only by relative vs absolute URL, keep the absolute form once. 4) Accept only dates matching regex ^\\d{2}\\.\\d{2}\\.\\d{4}$. Skip rows with malformed or future-inconsistent dates. 5) The case id is the trimmed visible text of the anchor in the second column (do not synthesize or alter spacing except collapse internal runs to single spaces). 6) Resolve relative hrefs against base https://search.bger.ch/ext/eurospider/live/de/php/aza/http/ (strip fragments and query parameters only if they are empty; otherwise keep them). 7) Exclude anchors that are purely pagination, sorting, JavaScript, mailto:, or '#' placeholders. 8) No guessing: if a row's structure is ambiguous, skip it rather than inventing data. 9) Output ONLY valid JSON: {\"decisions\": [{...}, ...]}. If none: {\"decisions\": []}. 10) No extra text, no
 }
 }
]
 }
]
}

```

Markdown, no code fences, no comments. 11) Never output duplicate keys or trailing commas. 12) Do not add fields other than date,id,url.",

```

 "user": "HTML:\n{{date_page_html}}",
 "retry": { "max": 2, "delay_ms": 3000, "backoff": 2.5
 },
 "retry_on_invalid": true,
 "require_key": "decisions",
 "temperature": 0
},
"assign": { "var": "decision_links", "path": "decisions" }
},
{
 "id": "normalize_decision_links_if_missing",
 "command": "if",
 "params": {
 "condition": "exists {{decision_links}}",
 "else_steps": [
 {
 "id": "force_empty_decisions",
 "command": "set_var",
 "params": { "name": "decision_links", "value": [] }
 }
]
 }
},
{
 "id": "loop_decisions",
 "command": "foreach",
 "params": {
 "list": "decision_links",
 "item_var": "decision",
 "max_items": 100,
 "steps": [
 {
 "id": "open_decision",
 "command": "open_url",
 "params": {
 "url": "{{decision.url}}",
 "return_body": true
 },
 "assign": { "var": "decision_html", "path": "body" }
 },
 {
 "id": "summarize_decision",
 "command": "llm_analyze",
 "params": {
 "system": "You are a deterministic legal decision summarizer. Input: (a) metadata (id,date,url) and (b) raw HTML of a single Swiss Federal Supreme Court decision page. Produce 2-3 concise German sentences: (1) legal domain / core issue, (2) essential holding / outcome. No speculation. If meaningful content cannot be reliably extracted, use summary=\"(Keine verwertbare Entscheidgrundlage extrahierbar)\". Output ONLY valid JSON:
{{\"id\":\"...\",\"date\":\"dd.mm.yyyy\",\"url\":\"...\",\"summary\":\"...\"}}. No extra text, no code fences."
 }
]
 }
}

```

```

 "user": "Metadata:\nID: {{decision.id}}\nDate:
{{decision.date}}\nURL: {{deci-
sion.url}}\n\nHTML:\n{{decision_html}}",
 "retry": { "max": 2, "delay_ms": 3000, "backoff":
2.5 },
 "retry_on_invalid": true,
 "require_key":
"id,date,url,summary",
 "temperature": 0
},
 "assign": { "var": "decision_summary_obj" }
},
{
 "id": "append_summary_markdown_if_nonempty",
 "command": "if",
 "params": {
 "condition": "exists {{summary_list}}",
 "steps": [
 {
 "id": "append_with_nl",
 "command": "template",
 "params": {
 "template": "{{summary_list}}\n-
{{decision_summary_obj.id}} ({{deci-
sion_summary_obj.date}})
[Link]({{decision_summary_obj.url}}): {{deci-
sion_summary_obj.summary}}"
 },
 "assign": { "var": "summary_list" }
 }
],
 "else_steps": [
 {
 "id": "append_first",
 "command": "template",
 "params": {
 "template": "-
{{decision_summary_obj.id}} ({{deci-
sion_summary_obj.date}})
[Link]({{decision_summary_obj.url}}): {{deci-
sion_summary_obj.summary}}"
 },
 "assign": { "var": "summary_list" }
 }
]
 },
 {
 "id": "push_structured_summary",
 "command": "array_push",
 "params": {
 "array": "summary_array",
 "item_var": "decision_summary_obj"
 }
 }
]
}

```

```
 }
]
}
},
{
 "id": "render_final_report",
 "command": "render_report",
 "params": {
 "template": "{{#summaries}}# Zusammenfassung Bundesgerichtsentscheide (Publikation: {{threshold_date}})\n{{summaries}}\n{{/summaries}}{{^summaries}}Keine Entscheidungen oder extrahierbaren Inhalte gefunden (Publikation: {{threshold_date}}).{{/summaries}}",
 "context": {
 "summaries": "{{summary_list}}",
 "threshold_date": "{{threshold_date}}",
 "summary_array": "{{summary_array}}"
 },
 "output_path": "{{env.DESKTOP}}\\BGer_Summaries.md"
 }
}
```